Statistics for Transcription Factor Binding Sites

Utz J. Pape

August 2008

Dissertation zur Erlangung des Grades
eines Doktors der Naturwissenschaften (Dr. rer. nat.)
am Fachbereich Mathematik und Informatik
der Freien Universität Berlin

Gutachter:
Prof. Dr. Martin Vingron
Prof. Dr. Sven Rahmann
Prof. Dr. Dr. h.c. Peter Deuflhard

1. Gutachter: Prof. Dr. Martin Vingron
2. Gutachter: Prof. Dr. Sven Rahmann
3. Gutachter: Prof. Dr. Dr. h.c. Peter Deuflhard

Tag der Promotion: 10.10.2008

Abstract

Transcription factors (TFs) play a key role in gene regulation. They interact with specific binding sites or motifs on the DNA sequence and regulate expression of genes downstream of these binding sites. *In silico* prediction of potential binding of a TF to a binding site is an important task in computational biology. From a statistical point of view, the DNA sequence is a long text consisting of four different letters ('A','C','G', and 'T'). The binding of a TF to the sequence corresponds to the occurence of a word in the sequence, e.g. 'AACCTC'. Hence, word count statistics can be applied to problems such as number of binding sites and distances between binding sites. The major problem in word count statistics are dependencies between sequence positions. These dependencies arise due to possible overlaps of words. So far, exact formulae to compute the count distribution of clustered occurrences only exist based on generating functions. We newly derive a recursive formula and use it to obtain a normal approximation.

In fact, a TF does not bind to one single word but allows mismatches and substitutions. This is captured in a statistical model called Position Frequency Matrix (PFM). A PFM assigns a score to each position of the word and letter. If the summed score reaches a certain threshold, the TF is assumed to bind to that sequence region. In fact, one can transform this representation to a set of words which are bound by the TF. Unfortunately, enumeration of the set of words takes exponential costs. In addition, the set of words grows enourmously for longer binding sites (around 500,000 for a binding site of length 15). Hence, word count statistics and its approximations become inefficient and very inaccurate.

Therefore, the need for new statistics and efficient algorithms arises. Instead of enumerating all words, we use a statistical representation - the PFM - and model dependencies explicitly. In fact, probabilities for overlaps are dependencies of the summed scores between two positions. Hence, we reduce the problem to computing the two dimensional convolution of the score distributions for each possible overlap and derive an exact formula for the variance of PFM counts. Furthermore, we found an accurate approximation for the distribution of the number of occurrences using a compound Poisson distribution. Our approximation outperforms all alternative approaches. In addition, we give Poisson statistics for the number of occurrences without overlaps such that other standard word count statistics (like distances between occurences) can be applied. Third, we develop statistics to compute the significance of co-occurrences and co-operativity among sets of TFs. Fourth, we use the variance to define a *natural* measure of similarity between DNA motifs. We explicitly state formulae for PFMs. Compared to standard approaches, it shows higher correlation with empirical data. It also allows to cluster sets of TFs and gives results comparable with more sophisticated clustering algorithms. Finally, we use this similarity measure to compute the representation quality of PFMs for a set of experimentally verified binding sites. Besides a threshold optimization method which significantly improves the quality of PFMs in Transfac and Jaspar, we can indeed select DNA motifs, which violate PFM assumptions and, therefore, cannot be reasonbly represented as PFMs.

Preface

Acknowledgements

First of all my thanks go to Prof. Martin Vingron, my supervisor, who gave me this opportunity and has supported me enormously since my Bachelor thesis. He also made my stay at *University of Southern California* (USC) at the *Molecular Computational Biology* section possible. There, I want to thank Prof. Fengzhu Sun for supporting me and helping scientifically with intense discussion. His whole group and other members of the institute welcomed me warmly. Furthermore, Prof. Michael Waterman inspired me to delve into word count statistics and his joint lecture with Prof. Fengzhu Sun and Prof. Simon Tavare gave me a good starting point in this field. However, most time I spent in Berlin in the *Computational Molecular Biology* department of Prof. Martin Vingron in the *Max Planck Institute for Molecular Genetics*. Especially, the gene regulation group as well as all other members of the department created a nice environment to work (and live). Pierre Nicodème who visited our group improved the thesis by discussions and many suggestions. Furthermore, the PhD students institutionalized in the *Student Association* (STA) were a big support from a scientific, sportive and friendship point of view. Also the members of the *International Max Planck Research School for Computational Biology and Scientific Computing* helped in discussing scientific things and spending free time with me. Additionally, I enjoyed teaching tutorials, seminars and lectures at the *Free University of Berlin* which were my main financial funding source. Lastly, I would like to especially thank Prof. Sven Rahmann for scientific discussions (often via internet), ideas, hints and contributions to our publications.

Published and related work

The main contribution of this thesis - derivation of the count statistics for transcription factors - has been publishd in *Journal of Computational Biology* (Pape *et al.*, 2008c) and was presented at the *International Workshop on Applied Probability 2008* by invitation. A preliminary idea has been previously published in *Genome Informatics* (Pape *et al.*, 2006). The application to co-occurrences and cis-regulatory modules (CRMs) has been presentend and published at the *International Workshop on Applied Probability 2008* (Pape and Vingron, 2008) while an extension was accepted at the *German Conference for Bioinformatics 2008* (Pape *et al.*, 2008a). The first application of the derived count statistics - similarity and clustering of DNA motifs - has appeared in *Bioinformatics* (Pape *et al.*, 2008d). The second application - quality of representation of DNA motifs - has been submitted to *Bioinformatics* (Pape *et al.*, 2008b). Chapter 3 about Word Counting is part of a script for the lecture 'Probability and Statistics for Sequence Analysis' from Prof. Martin Vingron and me at *Free University of Berlin* in winter semester 2007/2008 where I presented the part about count statistics.

Utz J. Pape Berlin, August 2008

Contents

Chapter 1

Introduction

1.1 Molecular Biology

Proteins Research over the last decades has revealed that proteins are the main ingredient of what we call *life*. Proteins are not only catalysts of most biochemical reactions in living systems but also serve as important structural constituents of living systems like blood, cytoskeleton and hair. Proteins are polymers: molecules made of repeated similar subunits called monomers. In the case of proteins, the monomers are amino acids. There are 20 amino acids used in living systems. One can break down the description of a protein to a sequence of amino acids. This, however, ignores important characteristics of the protein like modifications of the polymer and 3-dimensional structure. The sequence of amino acids is passed from generation to generation in living systems via the genetic code.

Desoxyribonucleic Acid (DNA) The genetic code is contained in the DNA (desoxyribonucleic acid). The DNA molecule is a double-stranded polymer (Watson and Crick, 1953). Its monomers are four different nucleotides (Alberts *et al.*, 2002) each characterized by an included *base*: adenine ('A'), cytosine ('C'), guanine ('G') or thymine ('T'). Both strands of the DNA molecule are directed from 5' to 3' (labeled by the number of a carbon atom). The strands are paired by complementary bases ('A' is complement of 'T' and 'C' is complement of 'G') such that the 5' end from one strand corresponds to the 3' end of the other strand. Hence, both strands contain the same information but are read in reverse direction and with complementary bases (Berg *et al.*, 2002). For example, the sequence 5'-ACCGAT-3' has the complementary strand 3'-TGGCTA-5' or - read in the common direction - the reverse complementary strand 5'-ATCGGT-3'.

Genetic Code Embedded in the DNA are the genes. Each gene corresponds to a stretch of DNA containing information for building a protein. This information is encoded by triplets - codons - of nucleotides each representing an amino acid (Osawa *et al.*, 1992).[1] A protein is produced by steps called transcription (from DNA to an intermediator) and, subsequently, translation (form the intermediator to a protein) - referred to as the *central dogma of molecular biology*. The rate or amount of transcription is the *expression level* of a gene. Hence, for a gene with a high expression level, the corresponding protein is produced in high amounts.[2]

[1] Codons are not necessarily unique. For example, alanine is also encoded by 'GCC' and others.

[2] Here, we ignore post-transcriptional effects like miRNAs (Lim *et al.*, 2005) and others (Scheper *et al.*, 2007).

Gene Regulation All cells[3] in a living system carry a copy of the same DNA molecule. However, specialized cells look different and do different things. The main reason for that is the different composition of proteins obtained by modified expression levels of the corresponding genes. Primarily, the gene expression level is regulated by combinatorial absence and presence of specific proteins called transcription factors (TFs; Myers and Kronberg, 2000). This specificity is seen in DNA motifs, 5-25 base pairs long, in the vicinity of the transcription start site (TSS). The set of DNA motifs close to the gene's TSS is called *cis* regulatory module (CRM; Berman *et al.*, 2002; Clyde *et al.*, 2003; Harbison *et al.*, 2004).

A DNA motif is a set of binding sites, which can be bound by a TF (Pabo and Sauer, 1984). They are usually illustrated by sequence logos (Crooks *et al.*, 2004). TFs bind to their specific binding sites as long as the binding sites are accessible (Spiegelman and Heinrich, 2004) and the TF is sufficiently abundant in the cell. Binding sites can be rendered inaccessible by TFs bound nearby and spatially covering the binding site or due to structural properties of DNA like chromatin structure (Klose and Bird, 2006). Hence, gene expression is regulated combinatorially by presence/absence of TF-specific binding sites (TFBSs) close to the TSS and the presence/absence of TFs which are again regulated in the same manner. Therefore, description of the regulatory region of a gene by its binding sites is crucial to understanding gene regulation.

Detection of TFBSs Since high throughput methods for genome sequencing are available (Maxam and Gilbert, 1977; Sanger *et al.*, 1977), computational biology tries to decipher gene regulation *in silico* (MacIsaac and Fraenkel, 2006; Tompa *et al.*, 2005). Therefore, approaches are developed to pinpoint binding sites on sequences. Usually this is done by searching for experimentally verified binding sites in sequences near to the TSS with length between 500 to 10,000 bases. Considering binding sites as words or sets of words in an alphabet with four letters 'A','C','G' and 'T' and the sequence as one long text in the same alphabet, detection of binding sites becomes the problem of searching words in a text.

Statistics for TFBSs Detection of binding sites results in a certain number of occurrences of given words in a text of given length. The problem is the interpretation of such a result: a long random text based on a few letters will almost always contain a certain number of occurrences just by chance. This raises the question whether the outcome is exceptional or statistically significant. To answer this question, one assumes a null model for the text, which randomizes the occurrences of the given words. Then, the significance is the probability to find at least as many occurrences in the random text as one has observed. Dependencies complicate the calculation of significance: for example, the occurrence of the word 'AAAA' raises the probability to observe a second occurrence of 'AAAA' shifted by one position. Additionally, the complementary strand introduces further dependencies, e.g. an occurrence of 'ACGT' implies an occurrence on the complementary strand and an occurrence 'ATA' increases the probability to observe another occurrence on the reverse complementary strand shifted by one position. Finally, a TF does not bind to only one binding site but to hundreds or millions of binding sites depending on the length.

[3] that have a nucleus

1.2 Thesis Overview

Besides contributions to word counting, this thesis studies statistics for TFBSs. Based on a specific model - the position frequency matrix (PFM) model (Stormo, 2000) - for DNA motifs, we derive efficient and exact formulae to compute the variance of motif counts in random sequences. Furthermore, we develop an efficient and accurate approximation for the distribution of the number of occurrences. Based on these results, we approximate significance values for co-occurrences of a set of different DNA motifs.

The availability of the count distribution leads to several applications in the context of DNA motifs: We define a *natural* and *general* (independent of the chosen DNA motif model) similarity measure between DNA motifs and introduce a new clustering of PFMs based on a related measure. Finally, we show how to compute the representation quality of a DNA motif model with respect to a given set of experimentally verified binding sites. For those *general* measures, we explicitly state formulae to deal with the PFM model.

The thesis is divided into two parts. In the theory part, we give a more specific introduction into DNA motifs, derive the count distribution and its second moment and deal with sets of TFs. In the second part, we present results indicating the accuracy and efficiency of our approximations and develop the similarity measure, the clustering and the quality of representation. A more precise overview of the thesis is given in the remainder of this section. In Appendix A, our software package to compute all above statistics is described in detail. Furthermore, a website http://mosta.molgen.mpg.de offers a user interface for the calculation.

1.2.1 Theory

DNA Motifs Chapter 2 gives an overview of experimental (Elnitski *et al.*, 2006; Maston *et al.*, 2006) and computational approaches (e.q. Bulyk, 2003; Pavesi *et al.*, 2004; Das and Dai, 2007) to detect TFBS. Furthermore, we present several models for DNA motifs with special focus on the PFM model (Stormo, 2000) and describe null models for sequences (Robin *et al.*, 2005). For the remainder of the thesis, we stick to the Bernoulli (i.i.d.) model. We also discuss how to derive a set of words from the PFM model such that one can apply classical word counting approaches (Robin *et al.*, 2005) to compute the significance of TFBS occurrences. However, computation of this set of words takes exponential time in the length of the DNA motif (see Section 2.4.4) and, therefore, is infeasible for longer DNA motifs.

Word Count Statistics Chapter 3 gives a detailed review of word counting approaches (Reinert *et al.*, 2005; Robin *et al.*, 2005). Starting with exact formulae for occurrences of single words (Rahmann, 2000; Robin *et al.*, 2005; Zhang *et al.*, 2007), we proceed with (compound) Poisson approximations (Reinert and Schbath, 1999; Roquain and Schbath, 2007) and normal approximations (Waterman, 2000) to cope with sets of words. Since some exact formulae are not available in the literature, we newly derive them. This enables us to give the first normal approximation for non-overlapping occurrences. Furthermore, we compute the count distribution and its approximations for a number of examples to clarify the drawbacks of certain approximations. In the subsequent chapter, we present

generating functions for word counting (Rahmann, 2000; Stefanov *et al.*, 2007) since we develop generating functions for PFMs under certain assumptions in Chapter 5.

TF Count Statistics In Chapter 5, we present the first exact formulae for the variance of the number of occurrences of PFMs based on overlap probabilities. Furthermore, we develop an approximation for the count distribution by considering the complementary strand and self-overlaps of PFMs using a compound Poisson distribution. Based on this, we describe two characteristic values indicating palindromicity and self-overlaps of PFMs. After deriving generating functions under a simplified sequence model, we present an efficient branch-and-bound dynamic programming algorithm to compute overlap probabilities of PFMs.

cis-regulatory Modules A related problem is the computation of co-occurrences of sets of PFMs organized in CRMs. Again, based on the overlap probabilities, we derive an accurate approximation for the significance of co-occurrences in Chapter 6. PFMs which co-occur exceptionally often are called co-operative. Hence, we show how to obtain the significance for co-operativity of pairs and sets of PFMs. Since one has to divide the sequence into windows to compute co-operativity, we also compute the Chen-Stein approximation error for overlapping windows.

1.2.2 Applications

Count Statistics In Chapter 7, we apply the approximated count distribution to artificial and real PFMs and compare the results to a simulation study. The approximation is very accurate and, furthermore, outperforms all competitive approaches (Reinert and Schbath, 1999; Waterman, 2000; Roquain and Schbath, 2007) in at least one case. Furthermore, interpretation of the characteristic values is presented, as well as a running time comparison with the state-of-the-art exact count distribution (Zhang *et al.*, 2007).

Co-Occurrences and Co-Operativity For pairs of PFMs, we compute probabilities for co-occurrences and co-operativity for artificial PFMs in Chapter 8. A simulation shows that our approximation works well. Comparison to an approach ignoring dependencies yields much better results for the new approach. Incorporating empirical occurrence frequencies does not change the picture. We also show and discuss the Chen-Stein bounds and their relation to similarity between PFMs.

Similarity of DNA Motifs Based on the exact formula of the variance, we develop a similarity measure for DNA motifs in Chapter 9. Applied to PFMs, we show its performance in comparison to a simulation and to other approaches (Schones *et al.*, 2005; Roepcke *et al.*, 2005; Gupta *et al.*, 2007). The simulation is also performed for Transfac (Matys *et al.*, 2003) PFMs and yields similar results.

Clustering of PFMs As shown in Chapter 10, a related similarity measure can be used to retrieve a clustering of PFMs. We develop a new clustering algorithm ensuring consistent clusters and apply it to a set of Jaspar (Sandelin *et al.*, 2004a) PFMs labeled by the corresponding TF class. In comparison to optimized clustering procedures (Mahony *et al.*, 2007), we retrieve similar or better results. Furthermore, the clustering automatically generates representative PFMs for each cluster which can be used as substitute for redundant and, therefore, similar PFMs.

Quality of Representation In Chapter 11, the final application uses the similarity measure to compute the quality of representation between DNA motifs and the experimentally verified binding sequences. For PFMs, we show how to use the quality to optimize parameters for PFMs and that this optimization yields a better overall quality of Transfac (Matys *et al.*, 2003) and Jaspar (Sandelin *et al.*, 2004a) PFMs. Finally, comparison of DNA motifs with and without position dependencies reveal that the quality measure can indeed differentiate between them.

Part I

Count Statistics

Chapter 2

DNA Motifs

A DNA motif is a set of prefered binding sequences of a TF. In computational biology, DNA motifs are mainly used for two different but related tasks:

- **Detection of Binding Sites:** Based on a given DNA motif, one annotates a given sequence with its binding sites (see Stormo (2000) for a review). Usually, this involves a score calculation for each position. If the score exceeds a certain threshold, the position is called a hit.
- **Discovery of New DNA Motifs:** Since experimental discovery of DNA motifs (see Section 2.1) is still difficult, a vast amount of computational methods have emerged to discover new DNA motifs *in silico* (for review, see Bulyk, 2003; Pavesi *et al.*, 2004; Wasserman and Sandelin, 2004; Li and Tompa, 2006; MacIsaac and Fraenkel, 2006; Sandve and Drabløs, 2006; Das and Dai, 2007). Given a set of sequences, which are assumed to be regulated by the same TF, one tries to pinpoint the corresponding binding sites without knowing the actual DNA motif. Often, one applies these methods to regulatory regions retrieved by experiments (see below) or to sets of co-expressed genes (Brazma *et al.*, 1998b; van Helden *et al.*, 1998; Bussemaker *et al.*, 2001).

Both tasks depend on a predefined model for the DNA motif. After reviewing experimental methods to discover DNA motifs, we present models for DNA motifs differing in their complexity. Finally, we will put more emphasis on the PFM model as this model is going to be used throughout the thesis.

2.1 Experimental Detection of Transcription Factor Binding Sites

The natural way to detect TFBSs is by wet-lab experiments. Many different approaches and enhancements have been proposed (for review, see Elnitski *et al.*, 2006; Maston *et al.*, 2006). Here, we shortly discuss the main approaches and mention their advantages and drawbacks. Generally, the approaches can be divided into two classes (Elnitski *et al.*, 2006) depending on whether the TF is known. The approaches also differ in their capability to precisely localize the binding site and to perform high-throughput experiments. In fact, there is a trade-off between localization and high-throughput capability. Precise localization is very time-consuming while high-throughput methods are supposed to run fast. As a consequence, most high-throughput methods need to be followed by a computational analysis to precisely locate the binding site.

2.1.1 Unknown Transcription Factor

DNaseI Hypersensitivity

DNaseI is a nuclease which cleaves DNA. DNaseI hypersensitive regions are parts of the DNA sequence which are easily cleaved by DNaseI. Such regions have an open chromatin structure. Due to the open chromatin structure, the DNA is sensitive to cleavage done by nucleases like DNaseI (Gazit and Cedar, 1980). The genome contains generalized nuclease sensitivity regions comprising 10-100 kilobases (e.g. Lawson *et al.*, 1982) and local hypersensitivity regions of lengths around 100-400bp (Gross and Garrard, 1988). If such a hypersensitive region is a non-coding sequence, the reason for an open chromatin structure might be due to binding of TFs. Thus, it can be used as a marker for a region containg binding sites (Cereghini *et al.*, 1984).

The high-throughput approaches differ in their resolution of hypersensitive regions. The indirect end-labeling technique yields a resolution of around 500bp (Wu, 1980) while (quantitative) PCR methods can resolve nearly every nucleotide (Yoo *et al.*, 1996; McArthur *et al.*, 2001). Other new methods are quantitative chromatin profiling (Dorschner *et al.*, 2004), massively parallel signature sequencing (Crawford *et al.*, 2006), and determining nucleosome positions by tiled microarrays (Yuan *et al.*, 2005).

These methods can be used for whole genome analysis. However, the opening of the chromatin structure depends on the cell type and developmental stage as well as environmental factors. Thus, many experiments have to be performed to detect all regulatory regions. Furthermore, the insensitivity with respect to a certain TF limits the use for the focussed detection of TF binding sites.

Promoter Analyses

Promoter analysis is a functional assay measuring the change of expression of a reporter in dependence on the upstream sequence. Common reporter genes are the luciferase reporter gene (de Wet *et al.*, 1987) and the green fluorescent protein GFP (Tsien, 1998). The chosen gene is incorporated into a plasmid which is used to transfect a cell. Depending on the upstream sequence of the gene in the plasmid, TFs can bind to that upstream sequence and enhance or inhibit expression of the gene. Due to the expressed reporter protein such changes in the expression can be measured. Mutations at positions of the binding sites alter the expression of the reporter gene. Thus, nucleotides which are crucial for the affinity of the TFs can be located. This assay can also be done in a high-throughput manner based on lipofection (Strauss, 1996), coinjection (Müller *et al.*, 1997) or nucleofection (Siemen *et al.*, 2005).

Although high-throughput methods exist to screen different upstream sequences of the plasmid, the number of sufficient mutations to precisely localize the binding site is high. Furthermore, different TFs in the transfected cell might have combinatorial effects on the (mutated) upstream sequence. Then, the location of the binding site can be the combination of several binding sites.

Mobility Shift Assays

In a mobility shift assay, the radionucleotide-labeled DNA bound to TFs is put into a polyacrylamide gel. After switching on the voltage, the molecules move to the anode with speed depending on the size of the molecule. Due to smaller size, unbound DNA moves faster than protein bound DNA.

Gel Shift Assay The electrophoretic mobility shift assay (EMSA, see Fried and Crothers, 1981; Garner and Revzin, 1981) detects binding of a TF to DNA (around 25bp length). In case the protein does not bind to the DNA, the gel fractionates two bands: The DNA band near the anode and the band for the slower protein near the cathode. In contrast, if the DNA is bound by the protein, another band occurs between the protein band and the cathode since the DNA bound by the protein is bigger than each single molecules. This way, one can localize the binding site by using different DNA sequences (Kadonaga and Tjian, 1986). The major disadvantage is the unwanted detection of non-specific DNA-protein interactions due to DNA repair molecules and others (Klug, 1997).

DNaseI Protection / Footprinting A TF bound to DNA protects it from cleavage by covering it. Removing the TFs after cleavage, the radionucleotide-labeled DNA in the polyacrylamide gel yields many bands: one for each length of DNA sequences. Since bound TFs prevent DNA pieces of intermediate lengths from cleavage by protection, areas without bands occur. These indicate the location of binding sites. As these areas look like footprints, DNaseI protection is also called DNaseI footprinting. The length of the DNA piece is usually around 500bp, thus, multiple binding sites can be detected simultaneously (Galas and Schmitz, 1978). Instead of enzyme cleavage chemical cleavage can also be used to avoid some enzyme-specific limitations (Drouin *et al.*, 2001). In all cases, again, the major drawback is the detection of unintended interactions (Klug, 1997).

2.1.2 Known Transcription Factor

SELEX and CASTing

Two high-throughput variants of mobility shift assays exist. Both, systematic evolution of ligands by exponential enrichment (SELEX, see Oliphant *et al.*, 1989; Tuerk and Gold, 1990; Fitzwater and Polisky, 1996) and cyclic amplification and selection of targets (CASTing, see Wright *et al.*, 1991), are *in vitro* approaches so that specific TFs can be used. They generate a pool of short random oligonucleotides which are subsequently screened using high-throughput mobility assays.

ChIP Assays

In contrast to *in vitro* methods, the main problem with *in vivo* methods is the selection of a specific TF. Here, chromatin-immunoprecipitation (ChIP) can be used (Boyd *et al.*, 1998; Orlando, 2000). TFs bound to DNA are crosslinked to the DNA by formaldehyde. Subsequently, the DNA is cleaved into 100-500bp long sequences. Then, a TF specific

antibody combined with a retrievable anchor molecule tags the TF. Based on the anchor, the TF and its bound DNA can be precipitated. The retrieved DNA is further analyzed in a second step. The main drawback of the ChIP method is the limited availability of TF specific antibodies.

ChIP-Amplification The retrieved DNA can be sequenced to retrieve the exact sequence of the binding region. Due to the length of DNA fragments, precise localization of the binding site is not possible. However, one can combine this technique with DNaseI protection and cleave DNA before releasing it from the TF (Kang *et al.*, 2002). Thereby, one detects the precise location of a specific TF.

ChIP-chip Alternative to amplifying the retrieved DNA, microarray analysis can be performed (Ren *et al.*, 2000; Iyer *et al.*, 2001; Hanlon and Lieb, 2004). Due to hybridization of the retrieved DNA to the probes on the microarray, enrichment of the sequences bound by the TF can be detected. Ren *et al.* (2000) chose a cDNA microarray while newer applications are based on 50mer oligonucleotide tiling arrays (Kim *et al.*, 2005). One problem is the dependence on statistical assessment of spot intensities of microarrays (for a summary, see Buck and Lieb, 2004). The main problem remains: the resolution of the location of binding sites depends on the length of the oligonucleotides. Miniaturization of microarrays might solve this problem in future.

Summary

Many different experimental methods to localize binding sites exist. Most *in vivo* methods - except the ChIP approach - cannot be applied to a specific TF. Although availability of antibodies to precipitate the DNA-TF complex is limited, the ChIP approach is widely and sucessfully applied. The ChIP-chip method, as well as most other high-throughput methods, suffer from restricted resolution of the binding site location. Therefore, *in silico* methods are applied in a post-processing step to precisely pinpoint binding sites. These computational methods for discovery of DNA motifs are based on a model for DNA motifs. Different models are discussed in the next section.

2.2 Models for DNA Motifs

A set of experimentally retrieved binding sites is often the starting point for further computational analyses. The experimentally retrieved binding sites are assumed to be instances of one TF. Furthermore, one assumes that all binding sites are recognized by the same protein-DNA interaction site of the TF. Hence, the binding sites are aligned in a first step. Classic multiple sequence alignment algorithms (Thompson *et al.*, 1994; Notredame *et al.*, 2000; Morgenstern, 2004) are sufficient since neither the length of the binding sites nor the number of the binding sites is very high (see Notredame (2002) for an overview). Subsequently, the aligned set of binding sites is plugged into a model for a DNA motif. The selection of an appropriate model is crucial. Thus, we give an overview of the most important DNA motif models, which split into two types (GuhaThakurta, 2006): pattern-based and profile-based models.

2.2.1 Pattern-Based Models

The Model Pattern-based models generally assume that positions are independent. Based on the multiple sequence alignment, one assigns a consensus letter to each position using the IUPAC nucleic acid alphabet (IUPAC-IUB Commission on Biochemical Nomenclature, 1970). The alphabet contains a letter for each nucleotide, as well as for each combination of nucleotides. A good survey of rules to compute the consensus is Day and McMorris (1992).

Example 2.1. *Consider the following multiple sequence alignment where '-' denotes a gap in the alignment:*

$$\begin{array}{lccccc} \textit{Site 1:} & G & C & C & A & A \\ \textit{Site 2:} & G & T & C & A & T \\ \textit{Site 3:} & G & C & C & A & A \\ \textit{Site 4:} & - & T & C & A & T \\ \textit{Site 5:} & G & C & C & A & A \end{array}$$

One possible consensus is 'GCCAA' retrieved by the 'majority vote' rule (Yamauchi, 1991). One obviously loses much information on the second and the fifth position. Another possibility is the consensus 'GYCAW' where 'Y' denotes a pyrimidine and 'W' corresponds either to an 'A' or a 'T' (see Choo et al. *(1991) for the exact rules).* □

Detection of Binding Sites Given a pattern-based DNA motif, one can annotate a sequence with binding sites. Since the experimentally verified sequences are usually assumed to be a subset of the real binding sites, one allows a certain number of mismatches between the consensus and the sequence (Queen *et al.*, 1982). After defining a threshold for the number of mismatches, one can enumerate the words which have a distance (in terms of mismatches) less or equal to the threshold (for a more efficient algorithm, see Section 2.4.4). We call these words the compatible words. In Chapter 11, we derive a new quality measure for DNA motifs assessing how well the experimentally verified sequences are represented. The result of the annotation can be statistically assessed either by specialized significance estimation (Waterman *et al.*, 1984) or by applying standard word counting approaches (see Chapter 3). However, none of these approaches considers the complementary strand. In contrast, our new approach (see Chapter 5) can be applied to set of words incorporating the complementary strand.

Example 2.2. *The consensus 'GCCAA' with a threshold equal to one yields the set of compatible words*

$$\begin{array}{ll} \{ & 'GCCAA', 'ACCAA', 'CCCAA', 'TCCAA', 'GACAA', 'GGCAA', \\ & 'GTCAA', 'GCAAA', 'GCGAA', 'GCTAA', 'GCCCA', 'GCCGA', \\ & 'GCCTA', 'GCCAC', 'GCCAG', 'GCCAT' \quad \}. \end{array}$$

The size of the set of compatible words is equal to 16. *In Example 2.1, only the three 'GCCAA' (sites 1, 3, 5) from the set of experimentally verified sequences are contained in the*

set of compatible words. In contrast, applying a threshold of zero to the consensus 'GYCAW', the set of compatible words contains 10 *words. All experimentally verified sequences are present except 'ATCAT', 'CTCAT', 'TTCAT' which are obtained by extending '-TCAT' to all possibilities.* □

Discovery of New DNA Motifs Discovery of new pattern-based DNA motifs is done by searching for over-represented words (Korn *et al.*, 1977). Manual approaches determined the consensus by eye inspection of the set of given sequences (Pribnow, 1975; Rosenberg and Court, 1979). Since the amount of data grew, automatic method appeared which searched for common words and their neighbors in a set of sequences (Galas *et al.*, 1985; Mengeritsky and Smith, 1987) given a set of alignments. Nowadays, search algorithm also work without given alignments (see Brazma *et al.* (1998a) for a review). More recently, also whole cell expression data and functional annotations are integrated (Jensen and Knudsen, 2000), as well as sequence conservation (Corá *et al.*, 2005).

Summary The main advantage of pattern-based DNA motifs is the simple nature of the model. Therefore, construction of the DNA motif is very fast, as well, as calculation of the set of compatible words. Furthermore, the model is easy to understand and one only needs one parameter (threshold) for a given consensus. Furthermore, standard count statistics can be used to compute the significance of an observed number of binding sites. The major drawbacks are the arbitrary choice of a consensus method and the restricted qualitative nature of the model which reduces the space of DNA motifs. The model assumes position independence and all positions are equally important. Thus, in many cases, the DNA motif does not represent the experimentally verified sequences well. This stimulated the development of advanced models.

2.2.2 Profile-Based Models

More complex models for the representation of DNA motifs can be summarized under the concept of profile-based models. In general, a profile is a map from certain objects (letters or words) to empirical frequencies. In our case, the objects are words (the binding sites) of a given length. The different models within this class differ in their assumptions and, therefore, in their parameterization. We start with the most simple model and proceed to the most complex model.

Position Independence Model

The Model The most simple model is called Position Frequency Matrix (PFM) model (see Stormo (2000); Wasserman and Sandelin (2004) for a review). In this section, we only give a short overview since the model is described in more detail in Section 2.4. Since many DNA motifs do not show dependencies between their positions (Schneider, 1997; Stormo, 1998), the model assumes independence between them. Considering an alphabet $\mathfrak{A} = \{0, 1, 2, \ldots, |\mathfrak{A}| - 1\}$, each position κ is described by a multinomial probability vector $\pi_\kappa = (\pi_{\kappa,a})_{a \in \mathfrak{A}}$. A motif of length ℓ is represented by ℓ multinomial random variables $P_1, \ldots, P_\ell$

with distribution $P_\kappa \sim \mathcal{M}(|\mathfrak{A}|, \pi_\kappa)$ where $\mathcal{M}(\cdot,\cdot)$ denotes the multinomial distribution. Stringing together the probability vectors, one obtains the PFM:

$$\Pi = \begin{pmatrix} \pi_{1,1} & \dots & \pi_{\ell,1} \\ \dots & & \dots \\ \pi_{1,|\mathfrak{A}|} & \dots & \pi_{\ell,|\mathfrak{A}|} \end{pmatrix} \tag{2.1}$$

Using statistical mechanics theory, one can show that the logarithms of the nucleotide frequencies $\pi_{\kappa,a}$ are proportional to their contribution to the binding energy (Berg and von Hippel, 1987) if the surrounding sequence is unbiased in terms of base composition. In this case, the binding energy with a potential binding site $a = a_1 \dots a_\ell$ is proportional to $-\sum_{\kappa=1}^{\ell} \log_2(\pi_{\kappa,a_\kappa})$. Relaxing the assumption of unbiased nucleotide composition to a position independent composition μ_a for $a \in \mathfrak{A}$, one can derive the log-likelihood ratio $-\sum_{\kappa=1}^{\ell} \log_2(\pi_{\kappa,a_\kappa}/\mu(a_\kappa))$. Hence, one can use this proportionality to detect binding sites (Staden, 1984).

The PFM can be estimated from a set of experimentally verified binding sites using a maximum likelihood approach (Stormo, 1990). Based on a multiple sequence alignment of the verified binding sites, one obtains the position count matrix (PCM), which contains for each position κ and for each letter $a \in \mathfrak{A}$ the number of binding sites with letter a at position κ. Dividing by the number of binding sites yields the maximum likelihood estimates $\hat{\pi}_{\kappa,a}$. To reduce overfitting, the PFM is usually regularized (see Section 2.4.1 for details). The regularized PFM is often called the Position Weight Matrix (PWM). This also avoids the singularity of the logarithm at 0. Although in practice, one estimates the PFM from a set of experimentally verified binding sites, we usually assume the PFM to be given.

Example 2.3. *Consider the multiple sequence alignment from Example 2.1. The following PCM is obtained:*

Position :	*1*	*2*	*3*	*4*	*5*
A	0	0	0	5	3
C	0	3	5	0	0
G	4	0	0	0	0
T	0	2	0	0	2

Dividing by the number of counts per columns, one obtains the estimated PFM:

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0.6 \\ 0 & 0.6 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0.4 & 0 & 0 & 0.4 \end{pmatrix}$$

This PFM can be regularized yielding the PWM. Here, we use a simple regularization by adding 0.01 to each element before re-normalizing. One obtains:

$$\begin{pmatrix} 0.0096 & 0.0096 & 0.0096 & 0.9712 & 0.5865 \\ 0.0096 & 0.5865 & 0.9712 & 0.0096 & 0.0096 \\ 0.9712 & 0.0096 & 0.0096 & 0.0096 & 0.0096 \\ 0.0096 & 0.3942 & 0.0096 & 0.0096 & 0.3942 \end{pmatrix}$$

□

Detection of Binding Sites The detection of binding sites is based on a score for each window of length ℓ on the sequence. The score is the sum of the binding energy contributions. We define the Position Scoring Matrix (PSM) $\Psi = (\psi_{\kappa,a})$ containing log-likelihoods $\psi_{\kappa,a} = \ln(\pi_{\kappa,a}/\mu_{a_\kappa})$). Then, the score for a potential binding site $a = a_1 \dots a_\ell$ is proportional to $\sum_{\kappa=1}^{\ell} \psi_{\kappa,a_\kappa}$. Heumann *et al.* (1994) show that this definition of the score maximizes the probability that the scored sequence binds to any of the sequences the PFM is based on. If the score exceeds a certain threshold, the position is called a hit (see Section 2.4.3) for details). Figure 2.1 illustrates the PFM model, its estimation as well as the PSM.

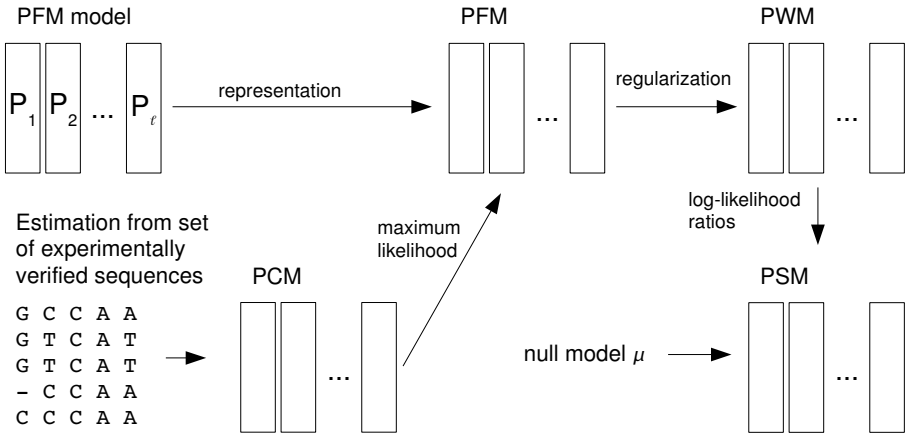

Figure 2.1: The figure illustrates the PFM model, its estimation and the derivation of the scoring matrix. The multinomial distributions are represented by probability vectors contained in the PFM. The PFM can be estimated from a set of experimentally verified binding sites by the empirical frequencies of position-specific nucleotide counts. Overfitting is avoided by regularization retrieving the PWM (note that some regularization methods might also use the null model μ as parameter). To detect binding sites, log-likelihood ratios are computed which yields the PSM.

Example 2.4. *Continuing Example 2.3, the PSM for a background model with a GC content of 50% ($\mu = (0.25, 0.25, 0.25, 0.25)$) is equal to*

$$\begin{pmatrix} -3.25 & -3.25 & -3.25 & 1.35 & 0.85 \\ -3.25 & 0.85 & 1.35 & -3.25 & -3.25 \\ 1.35 & -3.25 & -3.25 & -3.25 & -3.25 \\ -3.25 & 0.45 & -3.25 & -3.25 & 0.45 \end{pmatrix}$$

Due to computational issues (especially to compute the scoring distribution, see Section 2.4.2), one usually transforms the entries of the PSM to integers. Throughout this thesis,

we use a bin size of 0.05. The result is

$$\begin{pmatrix} -65 & -65 & -65 & 27 & 17 \\ -65 & 17 & 27 & -65 & -65 \\ 27 & -65 & -65 & -65 & -65 \\ -65 & 9 & -65 & -65 & 9 \end{pmatrix}$$

□

Discovery of New DNA Motifs Discovery of new PFMs on a set of given sequences is a difficult problem. After the development of greedy algorithms (Stormo and Hartzell, 1989; Hertz *et al.*, 1990), expectation-maximization (EM) algorithms have emerged (Lawrence and Reilly, 1990; Bailey and Elkan, 1994, 1995; Grundy *et al.*, 1996; Sinha *et al.*, 2004; Prakash *et al.*, 2004), as well, as Gibbs Sampling methods (Lawrence *et al.*, 1993; Neuwald *et al.*, 1995; Liu *et al.*, 2001; Thompson *et al.*, 2003). Recently, also simulated annealing algorithms are proposed (Grad *et al.*, 2004). Still, the problem is far from being solved (Hu *et al.*, 2005). Nowadays, a huge number of discovery algorithms are used (for review, see Li and Tompa, 2006; MacIsaac and Fraenkel, 2006; Das and Dai, 2007). Furthermore, some approaches also integrate data of different kind (Sandve and Drabløs, 2006), e.g. sequence conservation (Prakash *et al.*, 2004) or functional annotation (McGuire *et al.*, 2000).

Model Extensions

The model presented in the last section assumes independence between all positions. However, several publications indicate that positions of binding sites are dependent on each other (Benos *et al.*, 2001; Bulyk *et al.*, 2002). Hence, more complex models relaxing the independence assumption are proposed in the literature. Here, we shortly describe an extension of the PFM model using mixture models. Then, we review the incorporation of adjacent and non-adjacent position dependencies.

Mixture Models The PFM contains one distribution at each position. Thus, one assumes that all binding sites of a TF stem from the same consensus sequence. Since TFs interact with other molecules leading to topological changes, the binding site of the TF might alter (Bilu and Barkai, 2005). To model this issue, the single distributions can be substituted by mixture distributions where each component represents one type of binding (Barash *et al.*, 2003; Hannenhalli and Wang, 2005). The high number of parameters restricts their applicability. However, many positions are usually best described by one component since the topological change only affects a few positions. Thus, one can also use the context-specific independence mixture model (Georgi and Schliep, 2006). This model allows different number of components for each position. Thus, the total number of parameters is reduced again.

Adjacent Position Dependencies In the standard PFM model, each column of the PSM corresponds to one mono-nucleotide. Since the scores is simply added up, each mono-nucleotide at each each position contributes independently to the overall score. To relax the independence assumption, one can consider multi-nucleotides at each position (Brendel and Trifonov, 1984; Stormo *et al.*, 1986; Zhang and Marr, 1993; Gunewardena and Zhang, 2008). Although this representation might fit better to the real binding sites, the problem arises that one needs many experimentally verified sequences to circumvent over-fitting. Hence, the multi-nucleotide models have only been rarely used (Stormo, 2000). Still, some more complex models exist which are presented next.

General Position Dependencies The multi-nucleotide PFM model only incorporates adjacent position dependencies but also non-adjacent dependencies are observed (Agarwal and Bafna, 1998). Hence, the most flexible model would be modeling the joint distribution over all words. Since this would require too many parameters, some assumptions are required. In the tree model, each position of the binding site is modeled as a random variable. In contrast to the PFM model, the random variabels can be dependent on each other. The tree model comprises all models where the graph with the random variables as nodes and the dependencies as edges is a tree or a forest. A further extension is the use of Bayesian networks (Pearl, 1988) as presented by Barash *et al.* (2003). Another possibility is variable length permuted Markov models (Zhao *et al.*, 2005).

Conclusion

The PFM model is a very simple model assuming position independence. Due to its simplicity, computational issues like algorithmic complexity as well as statistical issues like over-fitting in the discovery of new motifs play a minor role. In contrast, the general drawback of the more complex models is the higher number of parameters. Therefore, parameter estimation as well as computations involving the score distribution introduce severe complications. Since lots of binding sites can be sufficiently represented by the standard PFM model, we only use the standard PFM model in this thesis. To deal with the issue of further dependencies, we introduce a new quality measure for the representation quality of the binding sites by the PFM (see Chapter 11). If this quality value is low, one might try more complicated models while otherwise the PFM model is appropriate.

2.3 Sequence Models

In computational biology, one is interested in findings with a biological meaning. For example, the assessment of the regulatory potential of a sequence regarding a given TF is such a task. Under the assumption that the number of binding sites is correlated with the regulatory potential, one can count the number of binding sites. This leads to the main statistical question whether the observed counts are exceptionally high. Given a probabilistic sequence model, one can compute the probability that such a high number of counts or more appear in a random sequence generated by the sequence model. If the probability, the p-value, is very low (e.g. $< 5\%$ or $< 1\%$) one calls the number of counts to be significant under the sequence model. In this way, the sequence model serves as null model

to reject the null hypothesis that the number of counts is not exceptional. Beforehand, one usually estimates the parameters for the sequence model from the given sequence. Here, we describe the main sequence models following Robin *et al.* (2005, Chapter 2).

2.3.1 Permutation Model

The permutation model is the most intuitive sequence model. The model contains all sequences which have the same number of each nucleotide $a \in \mathfrak{A}$ as the observed sequence. In general, we denote the number of counts on a sequence $\mathbf{X}$ by $N_{\mathbf{X}}(a)$ for the lettter/word $a \in \mathfrak{A}^+$. Then, we obtain as the set of sequences $\mathcal{X} := \{\mathbf{X} : \forall_{a \in \mathfrak{A}} N_{\mathbf{X}}(a) = N_{obs}(a)\}$. Obviously, the model is completely described by the observed number of counts of each nucleotide $N_{obs}(a)$. Furthermore, each sequence $\mathbf{X} \in \mathcal{X}$ has equal probability.

Generating a sequence under this model is fairly easy. After initializing a counter for each nucleotide with its observed number of counts, one starts with an empty sequence. At each step one nucleotide is randomly selected proportional to the counters whose values are greater than zero. After adding the nucleotide to the sequence, the corresponding nucleotide counter is decremented. This procedure is repeated until all counters are zero. A more illustrative description is one urn containing the whole sequence cut into single nucleotides. Then, one draws step by step one nucleotide randomly and concatenates to the sequence until the urn is empty.

Example 2.5. *We assume that the TF only binds to the motif 'ACT' and we ignore the complementary strand of the DNA. The observed sequence is 'AACTACACT' and obviously contains* 2 *occurrences of the motif. We compute the p-value for this number of counts. For the set of sequences, one obtains* $\mathcal{X} := \{\mathbf{X} : N_{\mathbf{X}}('A') = 4, N_{\mathbf{X}}('C') = 3, N_{\mathbf{X}}('G') = 0, N_{\mathbf{X}}('T') = 2\}$. *The size* $|\mathcal{X}|$ *of this set of sequences is computed by the multinomial coefficient*

$$|\mathcal{X}| = \frac{\left(\sum_{a \in \mathfrak{A}} N_{obs}(a)\right)!}{\prod_{a \in \mathfrak{A}} N_{obs}(a)!} = 1260.$$

Hence, each sequence occurs with a probability of $\frac{1}{1260} \approx 0.08\%$.

Next, we are interested in the probability to generate a sequence with at least two occurrences of 'ACT'. This is more complicated since we have to enumerate all the possibilities of 'ACT' occurrences. First of all, it is clear that the maximum number of occurrences is 2 *since the sequence only contains two 'T's. Hence, in this simple example, it would be sufficient to enumerate these cases. For better understanding, we show how to compute the whole count distribution. We can enumerate the positions of 'ACT' for each number of occurrences (see Table 2.1) to compute the distribution of the number of counts:*

- $N_{\mathbf{X}}('ACT') = 2$*: There are ten possible positions of the two occurrences. The dots have to be filled with the remaining nucleotides* $\{A, A, C\}$. *Thus, each of the ten possible positions can occur in* $\frac{3!}{2!}$ *variants. Hence, there are* 30 *different sequences with two occurences.*

	$N_{\mathbf{X}}$('ACT') = 1									$N_{\mathbf{X}}$('ACT') = 2								
(i)	A	C	T	.	.	.	.	.	.	A	C	T	A	C	T	.	.	.
(ii)	.	A	C	T	.	.	.	.	.	A	C	T	.	A	C	T	.	.
(iii)	.	.	A	C	T	.	.	.	.	A	C	T	.	.	A	C	T	.
(iv)	.	.	.	A	C	T	.	.	.	A	C	T	.	.	.	A	C	T
(v)	.	.	.	.	A	C	T	.	.	.	A	C	T	A	C	T	.	.
(vi)	.	.	.	.	.	A	C	T	.	.	A	C	T	.	A	C	T	.
(vii)	.	.	.	.	.	.	A	C	T	.	A	C	T	.	.	A	C	T
(viii)										.	.	A	C	T	A	C	T	.
(ix)										.	.	A	C	T	.	A	C	T
(x)										.	.	.	A	C	T	A	C	T

Table 2.1: This tables enumerates the possible positions for ACT for one (left) and two (right) occurrences.

- $N_{\mathbf{X}}(\text{'}ACT\text{'}) = 1$*: Following the same reasoning, each of the seven possibilities has* $\frac{6!}{3!\cdot 2!} = 60$ *variants. However, this might introduce a second occurrence. In fact, this happens twice for each of the possibilities of two occurences. E.g. the first possibility 'ACTACT...' is counted in 'ACT.....' and in the fourth possiblity '...ACT...'. Hence, we obtain* $60 \cdot 7 - (2 \cdot 30) = 360$ *sequences with exactly one occurrence.*
- $N_{\mathbf{X}}(\text{'}ACT\text{'}) = 0$*: The number of sequences without any occurrence of 'ACT' is equal to remaining sequences. Since the total number of sequences is* 1260*, we obtain* $1260 - 360 - 30 = 870$ *sequences without any occurrence.*

Therefore, the probability to observe equal to or more than 2 *occurrences is* $30/1260 \approx 2.4\%$*.* □

The example shows that naive calculation of the count distribution rapidly becomes rather complicated. Considering words with overlaps would further exacerbate the calculation while longer sequences would make the calculation infeasible. These problems are enhanced for permutation models based on di- or multi-nucleotides (Prum *et al.*, 1995; Robin *et al.*, 2005).

Generating sequences from di- or multi-nucleotide models is polynomial in sequence length (Kandel *et al.*, 1996). This problem can be transformed to sample with equillibrium probability Eulerian paths (Euler, 1736; Edmonds and Johnson, 1973; Altschul and Erickson, 1985; Wilson, 1986) where the nodes correspond to the di- or multi-nucleotides and edges are drawn if the suffix of a node matches the prefix of the next node (Robin *et al.*, 2005). Permutation models are rarely used. Instead, the restrictions of the sequence space are relaxed at the expense of the equillibrium probability distribution over the observed sequences. This leads to the Bernoulli Model.

2.3.2 Bernoulli/Multinomial Model

The Bernoulli or multinomial model considers the sequence $\mathbf{X}$ with length n to be a sequence of independently and identically distributed (i.i.d.) random variables $X_1, \ldots, X_n$. Each position X_i follows the nucleotide distribution $\mu(a)$ for $a \in \mathfrak{A}$. Given $\mu(a) > 0$ for all $a \in \mathfrak{A}$,

the sequence space has size $|\mathfrak{A}|^n$. It is much bigger than the space of the permutation model. Furthermore, not all sequences have equal probability since $\mathbb{P}_\mu(\mathbf{X}) = \prod_{i=1}^n \mu(X_i)$ for a realization of $\mathbf{X}$.

The nucleotide distribution μ is usually retrieved by a maximum likelihood approach (Aldrich, 1997). Given an observed sequence with counts $N_{obs}(a)$, the likelihood is equal to

$$\mathbb{P}(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = \prod_{a \in \mathfrak{A}} \mu(a)^{N_{obs}(a)}.$$

Considering $\mu(a) > 0$ and $\sum_{a \in \mathfrak{A}} \mu(a) = 1$, one can maximize the likelihood by computing the derivative and setting the result to zero (Reinert *et al.*, 2005). One obtains the maximum likelihood estimator $\hat{\mu}(a) = N_{obs}(a)/n$.

A sequence can be generated by sampling a new nucleotide from μ for each position. In other words, consider an urn which contains multiple copies of the nucleotides. The number of copies for each nucleotide is proportional to μ. Then, the sequence is created by drawing for each position one nucleotide from the urn with replacement. The likelihood to retrieve a sequence with the same nucleotide composition as the observed sequence is maximal due to the definition of the nucleotide distribution by the maximum likelihood estimator.

2.3.3 Markov Model

The Markov model is an extension of the Bernoulli model by relaxing the independence assumption of the positions. In a Markov model of first order, each position only depends on its predecessor. Hence, the Bernoulli model can also be considered a Markov model of zero order. The Markov model of first order has a di-nucleotide distribution instead of the mono-nucleotide distribution $\mu(a)$ for $a \in \mathfrak{A}$. Again, the nucleotide distribution is estimated from the observed sequence using a maximum likelihood approach yielding a similar result as above but for di-nucleotides. Also Markov models of higher order are possible (Brendel *et al.*, 1986). To estimate the minimum order of a Markov model for an observed sequence, one can perform a χ^2 test by comparing the frequencies of the nucleotide distributions (Rice, 1995; Reinert *et al.*, 2005). To consider heterogeneity, non-stationary Markov models can be applied (Borodovsky *et al.*, 1986; Tavaré and Song, 1989).

2.3.4 Conclusion

The most intuitive model is the permutation model. Due to the necessity of enumerating all words to compute a p-value, it is hardly used in practise. The Bernoulli model, or Markov model of zero order, considers the sequence to be a sequence of i.i.d. random variables which is asymptotically equal to the permutation model for infinite sequence lengths. The independence assumption simplifies most computations (see Chapter 3). Furthermore, incorporation of the complementary strand is straight forward if one assumes equal nucleotide probabilities for complementary bases. The Markov models of higher order capture more properties of the given sequence. Since positions are not independent any more, most calculation become more difficult. In addition, it is not obvious how to incorporate the

complementary strand. Therefore, we focus on the Bernoulli model throughout the remaining parts of this thesis.

2.4 The Position Frequency Matrix (PFM) Model

In Section 2.2.2, we have introduced the PFM model as a profile-based position independence model. On the one hand, the model is sufficiently flexible to model a big class of DNA motifs. On the other hand, the position independence assumption together with the Bernoulli sequence model allow efficient computation of many statistical properties. Thus, we choose this model for motif representation.

Here, we derive some basic facts for PFMs. After introducing regularization methods, we show how to compute the score distribution under the background model as well as under the motif model. This leads to threshold selection method which are subsequently discussed. Given a threshold, we present an algorithm to enumerate the compatible words. We finish with the description of a PFM visualization approach.

2.4.1 Regularization

The PFM contains the probabilities for each nucleotide and position. Due to the limited sample size of binding sites, the PFM might contain zero entries. However, nothing should be impossible, merely very improbable, which prevents overfitting. Therefore, one employs a regularization of the PFM. This also circumvents the singularity at zero for the logarithm.

The most straight-forward method is the addition of pseudocounts. One adds a small value to each PFM entry before re-normalizing the matrix. A more sophisticated method developed by Rahmann *et al.* (2003) emphasizes the information in high affinity positions. This method is based on certain requirements of a regularization method: The overall distribution of the PFM should be conserved and the binding signal should not be destroyed. Since the signal/relative entropy usually differs between the positions, this leads to a position-dependent regularization method which in general is able to support biological evidence (Mirny and Gelfand, 2002). For each position, one computes a weight which reflects the distance of the position-specific distribution to the overall nucleotide distribution of the motif. With consideration of the sampling error, one shifts the position-specific distribution towards the overall distribution based on the weights without destroying a significant signal.

Example 2.6. *Continuing Example 2.3, we first show the regularized PFM after adding pseuo-counts, again, for comparison:*

$$\begin{pmatrix} 0.0096 & 0.0096 & 0.0096 & 0.9712 & 0.5865 \\ 0.0096 & 0.5865 & 0.9712 & 0.0096 & 0.0096 \\ 0.9712 & 0.0096 & 0.0096 & 0.0096 & 0.0096 \\ 0.0096 & 0.3942 & 0.0096 & 0.0096 & 0.3942 \end{pmatrix}$$

Using the more sophisticated approach, one obtains

$$\begin{pmatrix} 0.0124 & 0.0228 & 0.0143 & 0.9713 & 0.5818 \\ 0.0124 & 0.5818 & 0.9713 & 0.0143 & 0.0228 \\ 0.9691 & 0.0114 & 0.0072 & 0.0072 & 0.0114 \\ 0.0062 & 0.3840 & 0.0072 & 0.0072 & 0.3840 \end{pmatrix}$$

The first qualitative difference occurs for position one: The weight for 'T' is less than for 'A' and 'C' although the entries in the PCM are equal. This is due to the conservation of the model distribution. Therefore, the distribution used for regularization is the model nucleotide distribution. Since the model contains hardly 'T's but more 'A' and 'C's, the 'T' gets less weight. The next difference is in position two where the entry for 'G' is different from the entry of 'T' of the first position although both have equal weights in the model nucleotide distribution. As the position specific nucleotide distribution at position two has not such a high relative entropy than the distribution of position one, position two is more regularized. Thus, the position specific distribution is shifted more towards the model nucleotide distribution. □

Besides these two regularization methods, other methods are developed especially in the context of the motif discovery program MEME (Bailey and Elkan, 1994, 1995). However, if not mentioned otherwise, we use the regularization method developed by Rahmann *et al.* (2003). Since the regularized PFM (PWM) is used for all computations, from now on, we use the term PFM and PWM interchangeably.

2.4.2 Score Distribution

A PSM yields a hit on a sequence if the score s reaches the threshold t. The significance of the hit is the probability of this event under a sequence model. Many articles deal with the efficient calculation of this probability (Staden, 1989; Claverie and Audic, 1996; Stormo, 2000; Wu *et al.*, 2000; Rahmann *et al.*, 2003; Beckstette *et al.*, 2006; Touzet and Varré, 2007). Here, we derive the score distribution and give a dynamic programming algorithm following Beckstette *et al.* (2006).

The score for a word $w = w_1, \ldots, w_\ell$ is computed by summing the position-specific scores corresponding to the nucleotides of the word:

$$s(w) = \sum_{\kappa=1}^{\ell} \Psi_{\kappa,w_\kappa}. \tag{2.2}$$

Given a sequence model instead of an actual word, the score is a random variable S. Similarily to Eq. (2.2), the score S is the sum of the position-specific scores $S^{(\kappa)}$ which are also random variables. Hence, the distribution of S denoted by $\mathcal{L}(S)$ is given by the convolution of $S^{(\kappa)}$:

$$\mathcal{L}(S) = \mathcal{L}(S^{(1)}) * \mathcal{L}(S^{(2)}) * \ldots * \mathcal{L}(S^{(\ell)}).$$

One can use a dynamic programming approach to compute $\mathcal{L}(S)$. The idea is simple: Before starting the summation, we have a Dirac score distribution with all its probability weight at 0. In each step, we add the scores of the next position. Thus, the score s yields the probability of the position-specific score $\Psi_{\kappa,a}$ and the probability of the score $s - \Psi_{\kappa,a}$ of the previous step. Using the Bernoulli sequence model with probabilities $\mu(a)$, this is

$$\begin{aligned} Q_0(s) &:= \begin{cases} 1 & \text{if} \quad s = 0, \\ \text{undefined} & \text{else}, \end{cases} \\ Q_\kappa(s) &:= \sum_{a \in \mathfrak{A}} Q_{\kappa-1}(s - \Psi_{\kappa,a}) \cdot \mu(a). \end{aligned}$$

After the last step, $Q_\ell(s)$ contains the probability to observe score s. Hence, we can write $\mathbb{P}_\mu(S = s) = Q_\ell(s)$. Replacing μ_a in the equation for Q_κ by a different nucleotide distribution yields the score distribution of S under another model. Furthermore, we can also use position dependent nucleotide distribution. In this way, it is straight-forward to compute the score distribution under the motif model. The distribution of the nucleotides of the motif are given in the PFM Π. Thus, we can compute the distribution by

$$\begin{aligned} Q'_0(s) &:= Q_0(s), \\ Q'_\kappa(s) &:= \sum_{a \in \mathfrak{A}} Q'_{\kappa-1}(s - \Psi_{\kappa,a}) \cdot \pi_{\kappa,a}. \end{aligned}$$

Hence, we have $\mathbb{P}_\Pi(S = s) = Q'_\ell(s)$.

Type-I and II error probabilities Based on these score distributions, we can compute type-I and type-II error probabilities. The type-I error occurs if the score reaches the threshold but without an actual binding site at this position (false positive). Using the sequence model μ as background model for a sequence without binding sites we can compute the type-I error probability by

$$\alpha := \mathbb{P}_\mu(S \geq t) = \sum_{s \geq t} Q_\ell(s).$$

Thus, α is the p-value or significance of a hit.

Likewise, we can compute the type-II error probability. Retrieving a score lower than the threshold on a position which is an actual binding site is a type-II error (false negative). Hence, we have to use the sequence model Π instead of μ and get for the type-II error probability

$$\beta := \mathbb{P}_\Pi(S < t) = \sum_{s < t} Q'_\ell(s).$$

Example 2.7. *Here, we reconsider the example DNA motif (see Ex. 2.1). The upper panel of Fig. 2.2 contains the distributions for the score of the PSM under the Bernoulli sequence model μ as background model and the motif model Π. The score distribution under the background model has in general lower scores and obtains very low probabilities for scores higher than zero. In contrast, the score distribution under the motif model has mainly scores higher than zero with increasing probabilities. The lower panel of Fig. 2.2 visualizes α and β. Both errors are almost equal to a score of 13. At this score, α is equal to 0.03 and β is 0.032.*

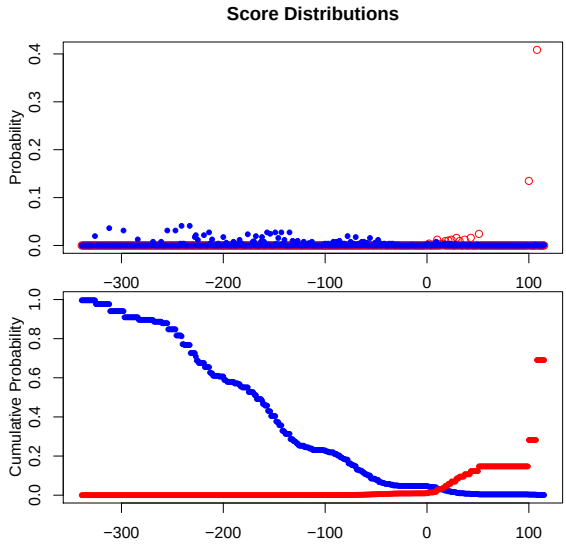

Figure 2.2: Score distribution for the example DNA motif (see Ex. 2.1) using the background model (blue) and the motif model (red). The lower panel contains the cumulated score distributions where the background distribution is reversely accumulated. Hence, they correspond to the α (blue) and β (red) probabilities.

□

Rahmann *et al.* (2003) introduces the concept of power for PFMs based on the two score distributions. If both distribution can be well separated by a threshold, the PFM is said to have good/high power. Otherwise, the motif is weak since it is hard to differentiate between motif and background model. Surprisingly, only one fifth of Transfac PFMs (Matys *et al.*, 2003) are shown to have a reasonable power (Rahmann *et al.*, 2003).

2.4.3 Threshold Selection

Previously, we have seen that the significance of a hit depends on the threshold t. Thus, selection of an appropriate threshold is crucial. Here, we introduce five different threshold selection methods and give examples for some of them.

Type-I Error Probability Instead of deducing the type-I error probability from the threshold, one can define the type-I error probability and compute the corresponding threshold

(Rahmann *et al.*, 2003; Touzet and Varré, 2007). However, the expected number of false positives on a sequence of length n is $n \cdot \alpha$. Hence, one should consider multiple testing. Therefore, we control the probability α_n to find at least one false positive on a sequence of length n. We obtain $\alpha_n \approx 1 - (1 - \alpha)^n \approx 1 - \exp(-n\alpha)$ where the first approximation is due to the fact that overlaps are ignored. Instead of using the actual sequence length, we always set $n = 500$ heurstically to get a threshold independent of the actual sequence length. We obtain a threshold which has a clear statistical background and also restricts the number of false positives independently of the motif.

Due to the discrete nature of the score, one usually cannot obtain a threshold which exactly corresponds to the pre-defined type-I error. Therefore, one could 'control' the type-I error such that the pre-defined type-I error is never exceeded. Unfortunately, for very low type-I errors which cannot be retrieved by a motif, this leads to a threshold which cannot be reached. Hence, we 'bound' the type-I error probability such that the next higher threshold $t + 1$ is less or equal to the pre-defined type-I error probability.

Type-II Error Probability In general, one can also pre-define the type-II error probability β instead of the type-I error. For motifs with weak power, this might lead to a huge amount of false positives and insignificant hits. Therefore, this threshold selection method only plays a minor role in practice. Especially, since it can be nicely combined with the type-I error as shown in the next method.

Balanced Error A reasonable threshold is a threshold which restricts the number of false positices as well as the number of false negatives. Therefore, we can combine the type-I and the type-II error by setting $t = t_{bal}$ such that $\alpha_{500} = \beta$ and call it 'balanced threshold' (Rahmann *et al.*, 2003). Again, the discrete nature of the score prevents both probabilities to be equal. Hence, we set the threshold such that $\beta < \alpha_{500}$ and for the next higher threshold $t + 1$ the inequality $\beta > \alpha_{500}$ holds.

Type-I Extended Error The balanced threshold can lead to very high false positive numbers if the power of the PFM is weak. Therefore, we newly introduce a threshold selection method (Pape *et al.*, 2006). The threshold is set to the balanced threshold if a pre-defined type-I error probability is not exceeded. Otherwise, the type-I error is used for threshold selection (as described earlier). Hence, one achieves a good balance between type-I and type-II error by ensuring a small number of false positives.

Number of Compatible Words Another new threshold selection method pre-defines the number of compatible words. This can be useful for analyses comparing two PFMs. One can achieve that by using the type-I error selelection method (using α instead of α_{500}) based on an equi-probable sequence model. Since all words have the same probability, the type-I error probability corresponds to the ratio of the number of compatible words and the number of all possible words.

Example 2.8. *Again, considering the example DNA motif (see Ex. 2.1), we can compute the different thresholds. Figure 2.3 contains the trajectories for α, β and α_{500} for all thresholds. The first two trajectories are already discussed in Example 2.7. The new trajectory*

is the probability α_{500} *for at least one false positive on a sequence of length* 500*. For low thresholds* ($t < 0$)*, the probability is almost* 1*. For higher thresholds, the probability drops and finally reaches around* 40 %*.*

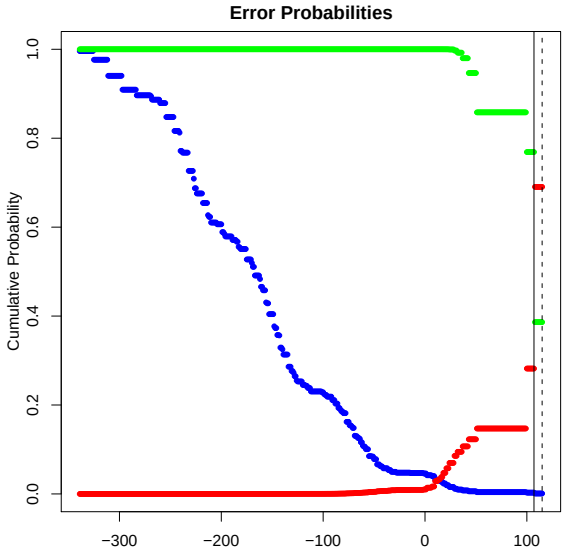

Figure 2.3: Error probabilties α and β for the example DNA motif (see Ex. 2.1) using the background model (blue) and the motif model (red). The green points correspond to the probability α_{500} of at least one false positive on a sequence of length 500. The solid line indicates the 'balanced' threshold while the dashed line corresponds to a 'type-I' error probability 'bound' at 10%.

Figure 2.3 also contains the type-I threshold for a level of 10 %*. The threshold is at* $t = 115$ *where we have* $\alpha = 0.00098$*,* $\alpha_{500} = 0.39$ *and* $\beta = 0.69$*. On a first look, it seems that the pre-defined type-I error probability of* 0.1 *is not really considered since* $\alpha_{500} = 0.39$*. However, the next higher threshold* $t + 1 = 116$ *achieves error probabilities* $\alpha = 0$*,* $\alpha_{500} = 0$ *and* $\beta = 1$*. For this threshold, no word would have a score reaching the threshold. Hence, the threshold* $t = 115$ *is the highest, reasonable threshold. This is the reason for setting the type-I error threshold such that the next higher threshold is below the pre-defined error probability. In practice, such extreme examples (large difference between pre-defined and resulting type-I error probabilities) do not occur since the motifs are longer and the PCM contains more different values.*

The balanced threshold is also depicted in Fig. 2.3. The threshold being at $t = 107$ *is slightly lower than the type-I threshold at* $t = 115$*. For the balanced threshold, we obtain* $\alpha = 0.0029$*,* $\alpha_{500} = 0.77$ *and* $\beta = 0.28$*. Considering the next higher threshold* $t+1 = 108$*, the error probabilities are* $\alpha = 0.00098$*,* $\alpha_{500} = 0.39$ *and* $\beta = 0.69$*. Hence, the balanced threshold fulfils* $\alpha_{500} > \beta$ *and the next higher threshold leads to* $\alpha_{500} < \beta$*. Applying the type-I extended threshold, one would disregard the balanced threshold since the corresponding type-I error probability exceeds* 0.1 *and, instead, use the type-I threshold of* $t = 115$*. Furthermore, the relative number of compatible words is equal to* α *because we use a GC content of* 50 %*. Since* $0.00098 \cdot 4^5 = 1$*, only one word ('GCCAA') is in the set of compatible words. This can be verified by applying the threshold* $t = 115$ *to the PSM in Ex. 2.4. However, the set of compatible words is usually much larger since real motifs are longer.* □

2.4.4 Enumerating Compatible Words

Enumeration of compatible words is important to apply word approaches (e.g. for word counting) to PFMs. The PFM model is fully described by the set of compatible words to annotate sequences given a threshold. Therefore, it is crucial to enumerate the compatible words efficiently. First, we describe a rank algorithm (Zhang *et al.*, 2007) which only depends on the length of the PFM and the number of compatible words. Hence, we subsequently discuss the number of compatible words depending on the chosen threshold method.

Rank Algorithm

The rank algorithm proposed by Zhang *et al.* (2007) uses a rank-lexikographic order of all words of the length of the PFM. Since the rank is based on the score of the letter, the ordering implies that all successors of a given word w with the same prefix have a score less or equal to $s(w)$. For words whose score does not reach the threshold, one can skip all successors with the same prefix. While formalizing the algorithm, we give examples and, finally, analyze the running time.

Based on the PSM, we assign to each position κ and each letter a the rank r of its score and obtain a re-ordered letter-matrix Ψ^* with general term $\psi^*_{\kappa,r} = a$ for $r \in \{0, \ldots, |\mathfrak{A}| - 1\}$. The ranks are assigned such that the word with highest resp. lowest score is $\psi^*_{1,0} \ldots \psi^*_{\ell,0}$ and $\psi^*_{1,|\mathfrak{A}|-1} \ldots \psi^*_{\ell,|\mathfrak{A}|-1}$. Thus, the successor of a given word $\psi^*_{1,r_1} \ldots \psi^*_{\ell,r_\ell}$ is the word $\psi^*_{1,r_1} \ldots \psi^*_{\kappa-1,r_{\kappa-1}} \psi^*_{\kappa,r_\kappa+1} \psi^*_{\kappa+1,0} \ldots \psi^*_{\ell,0}$ where κ the highest position with letter rank less than $|\mathfrak{A}| - 1$.

Example 2.9. *Given the following PSSM for a motif of length $\ell = 2$*

$$\Psi = \begin{pmatrix} 0 & 4 \\ 1 & 4 \\ 3 & -1 \\ 2 & 0 \end{pmatrix}$$

we obtain the re-ordered letter-matrix

$$\Psi^* = \begin{pmatrix} \text{'G'} & \text{'A'} \\ \text{'T'} & \text{'C'} \\ \text{'C'} & \text{'T'} \\ \text{'A'} & \text{'G'} \end{pmatrix}.$$

This implies the following ordering (in brackets we write the score of the word) 'GA' (7), 'GC' (7), 'GT' (3), 'GG' (2), 'TA' (6), 'TC' (6), 'TT' (2), 'TG' (1), 'CA' (5), 'CC' (5), 'CT' (1), 'CG' (0), 'AA' (4), 'AC' (4), 'AT' (0) and 'AG' (−1). Obviously, the ordering does not yield a monotously decreasing function of the score. However, given a threshold of 5, the first word below the threshold is 'GT'. Since all words with the same prefix 'G' cannot yield a score higher than 'GT', one skips the word 'GG' and directly proceeds with 'TA'. Although the savings are not astonishing in this example, it is easy to see that for larger PFMs many words can be skipped. □

Figure 2.4: Sequence logo created by Crooks *et al.* (2004) for the PFM defined in Ex. 2.1 with consensus 'GCCAA'.

To formalize the skipping procedure, we start with a given word $\psi^*_{1,r_1} \dots \psi^*_{\ell,r_\ell}$, which does not reach the threshold. Let τ denote the largest position with a non-zero rank. Then, all successive words with the same prefix of length $\tau - 1$ cannot yield a score higher than the given word. Hence, we can skip all successive words $\psi^*_{1,r_1} \dots \psi^*_{\tau-1,r_{\tau-1}} \psi^*_{\tau,r_{\tau+}} \psi^*_{\tau+1,\cdot} \dots \psi^*_{\ell,\cdot}$. In fact, these are $|\mathfrak{A}|^{\ell-\tau}$ words. The next word with a different prefix, which we have to consider, is $\psi^*_{1,r_1} \dots \psi^*_{\kappa-1,r_{\kappa-1}} \psi^*_{\kappa,r_\kappa+1} \psi^*_{\kappa+1,0} \dots \psi^*_{\ell,0}$ with κ being the largest position before τ with letter rank less than $|\mathfrak{A}| - 1$.

Example 2.10. *In above example with threshold* 5*, we can skip* 6 *words (colored in red where blue indicates words, which do not reach the threshold but are enumerated): 'GA' (7), 'GC' (7), 'GT' (3), 'GG' (2), 'TA' (6), 'TC' (6), 'TT' (2), 'TG' (1), 'CA' (5), 'CC' (5), 'CT' (1), 'CG' (0), 'AA' (4), 'AC' (4), 'AT' (0) and 'AG' (−1). Of course, for longer PFMs with a high threshold, the percentage of skipped words is significantly higher.* □

For each compatible word, one considers at most ℓ words, which do not reach the threshold. Denoting the set of compatible words by $\mathcal{A}$, one obtains an asymptotic complexity of $O(\ell|\mathcal{A}|)$. Zhang *et al.* (2007) mention that this is a polynomial time algorithm. However, depending on the threshold selection method, the size of $\mathcal{A}$ can be expressed in terms of the PFM length. As we see in the next section, this leads to an exponential time algorithm.

Number of Compatible Words

The number of compatible words depends on the threshold. Since we already introduced different threshold selection methods, we discuss the size of the set of compatible words $\mathcal{A}$ with respect to the threshold selection and the length of the PFM. If one defines the number of compatible words, obviously, the size of $\mathcal{A}$ is an input parameter. However, the threshold is chosen based on the type-I or type-II probabilities (including the balanced and the extended type-I errors). To obtain the asymptotic size of $\mathcal{A}$, one can assume an equi-probable sequence model. In this case, the number of compatible words is given by $\alpha|\mathfrak{A}|^\ell$. Since we assume α (or α_n) as a constant input parameter, we obtain asymptotically $|\mathcal{A}| = O(|\mathfrak{A}|^\ell)$. Hence, the rank algorithm has an asymptotic complexity of $O(\ell|\mathfrak{A}|^\ell)$. Therefore, enumerating the set of compatible words is not possible with a polynomial in ℓ algorithm if the threshold is chosen based on the error probabilities. It might also be possible to circumvent the enumeration of compatible words by using IUPAC codes. However, it is not obvious how to do this and whether complexity would be improved.

2.4.5 Sequence Logo

A convenient illustration of a DNA motif based on its PFM is the so-called sequence logo (Schneider and Stephens, 1990), see Fig. 2.4 for an example. The sequence logo shows the

preference of the TF to a certain nucleotide for each position. If the preference is very high, the position-specific distribution is a Dirac distribution with all probability weight at the prefered nucleotide. From a point of information theory (for an introduction, see Shannon and Weaver, 1949; MacKay, 2003), the information content is maximal (Shannon, 1948). The more unspecific the preference at a position is, the lower the information content. Therefore, one can use the information content to represent the strength of affinity of the TF for each position κ defined by

$$\log_2 |\mathfrak{A}| + \sum_{a \in \mathfrak{A}} \pi_{\kappa,a} \log_2 \pi_{\kappa,a}. \tag{2.3}$$

This formula assumes an equi-probable background distribution. Otherwise, one has to use the relative entropy (or Kullback-Leibler distance) between the position-specific and the background distribution defined by

$$\sum_{a \in \mathfrak{A}} \pi_{\kappa,a} \log_2 \frac{\pi_{\kappa,a}}{\mu(a)}.$$

For a given background distribution, the relative entropy reaches its maximum at $\log_2 |\mathfrak{A}|$ if $\pi_{\kappa,\cdot}$ is a Dirac distribution. The minimum 0 occurs if the position-specific distribution is equal to the background distribution. Note that the information content in Eq. (2.3) can be derived by setting $\mu(a) = |\mathfrak{A}|^{-1}$ for all a. The contribution of each nucleotide to the relative entropy is retrieved by multiplication with the position specific nucleotide frequency $\pi_{\kappa,a}$.

Based on these thoughts, one can draw a sequence logo where each position contains the contribution of each letter encoded by the height of the nucleotide such that the summed heights correspond to the relative entropy. In Fig. 2.4, the PFM from Ex. 2.1 with consensus 'GCCAA' is shown created by the program weblogo (Crooks *et al.*, 2004). Obviously, the first, third, and fourth positions have a very strong preference towards the consensus letters. In contrast, the second and the fifth positions have a weaker affinity but still a preference to 'C', 'T' respectively 'A', 'T' exists. Hence, the sequence logo is a suitable tool to visualize a DNA motif. However, dependencies between positions are not reflected in the sequence logos. This would require a more sophisticated approach like structural logos (Gorodkin *et al.*, 1997).

Chapter 3

Word Count Statistics

3.1 Introduction

Rapid sequencing of DNA (Maxam and Gilbert, 1977; Sanger *et al.*, 1977) generated a vast amount of sequences to be analyzed. First studies focused on protein coding sequences and analyzed codon usage (Almagor, 1983). Later, interest rose in non-coding sequences and in detection of exceptional words in sequences hinting for biological function (Pevzner *et al.*, 1989). Pattern occurrences in random strings is a classical problem (Feller, 1968). First exact results for the expected value of the number of occurrences were revealed based on simple probabilistic models (Dayhoff, 1984; Santibanez-Koref, 1987). This chapter reviews different methods for computing the distribution of the number of words (for other reviews, see Reinert *et al.*, 2005; Robin *et al.*, 2005). We also investigate the distribution of word clusters (clumps). We present a new exact formula to compute the variance and the exact distribution without using generating functions or automata. Computational issues are mainly ignored except for few remarks about algorithmic complexity (for an overview, consult Gusfield, 1997; Waterman, 2000; Lonardi, 2001).

Our review starts by considering single words (roughly following the exposition in Robin *et al.*, 2005). We present two exact approaches: First, the classical approach based on waiting time (Gentleman and Mullin, 1989; Gentleman, 1994; Robin *et al.*, 2005) and, second, a very recent approach (Zhang *et al.*, 2007) - we call it *conditional approach* - using optimally spaced seeds motivated by homology search (Ma *et al.*, 2002). Although this approach is specifically designed for PFMs, it takes as input a set of words (for PFMs, the set of compatible words). Hence, we classify it as a word counting approach. Since the exact approaches for words are infeasible to compute for large sets of words, we also introduce approximations. Initially, we consider an independence model and obtain a binomial and a Poisson approximation. For the Poisson approximation, we derive the Chen-Stein bounds explicitly. We also introduce a normal approximation. Then, we consider clumps instead of occurrences. After presenting the exact distribution, the binomial approximation and the Poisson approximation, we introduce the compound Poisson distribution. Although the compound Poisson distribution is used to compute the distribution of the number of occurrences, clumps are modelled explicitly. Therefore, we include this this approximation in Section 3.3 about clumps. We finish by deriving a asymptotic normal distribution. In the same order, we discuss the statistics for multiple words and clumps of multiple words. The whole chapter serves as a basis to treat bigger sets of words as encoded by PFMs.

Preliminaries We briefly repeat the notation from the last chapter and introduce some new definitions. The random sequence $\mathbf{X}$ consists of nucleotides $X_1, \ldots, X_n \in \mathfrak{A}^n$ assumed to be i.i.d. in the alphabet $\mathfrak{A}$. The alphabet $\mathfrak{A}$ is a set $\{0, 1, \ldots, |\mathfrak{A}| - 1\}$ where for DNA $|\mathfrak{A}| = 4$. For better readability, we sometimes refer to $0 \in \mathfrak{A}$ as 'A', $1 \in \mathfrak{A}$ as 'C', $2 \in \mathfrak{A}$ as 'G' and $3 \in \mathfrak{A}$ as 'T'. Each position X_i has the nucleotide distribution μ which is a map (of the σ-algebra of) $\mathfrak{A} \to [0, 1]$. We also write $\mu(w)$ for the probability of a word $w = w_1, \ldots, w_\ell$:

$$\mu(w) = \prod_{\kappa=1}^{\ell} \mu(w_\kappa).$$

Note that we use greek letters for indices within a word.

An occurrence of w is the event of w starting at any position i in the sequence $\mathbf{X}$. The binary random variables $Y_i(w)$ indicate this event:

$$Y_i(w) := \begin{cases} 1 & \text{if } X_i, X_{i+1}, \ldots, X_{i+\ell-1} = w_1, w_2, \ldots, w_\ell, \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

Hence, the number $N_n(w)$ of w occurring in a random sequence of length n is given by $N_n(w) = \sum_{i=1}^{n-\ell+1} Y_i(w)$. This is the key random variable. This chapter reviews different approaches to compute the distribution $\mathcal{L}(N_n(w))$ and its properties such as its first two moments. We always assume the parameters of the sequence model (basically μ) to be given, hence, not to be estimated. Otherwise, the asymptotical distributions change (Lundstrom, 1990; Prum *et al.*, 1995; Waterman, 2000; Robin *et al.*, 2005).

Chen-Stein Error Bounds For (compound) Poisson distributions, one can bound the approximation error using the Chen-Stein method (Chen, 1975). For a good introduction with many examples, see Arratia *et al.* (1989, 1990); Barbour *et al.* (1992). Furthermore, Barbour and Chryssaphinou (2001) give a guide to using compound Poisson distributions as approximations. Quantification of the approximation error is performed in terms of the total variation distance. Let U and V be any two random processes taking values in the same space E, then the total variation distance between their distributions is (Barbour *et al.*, 1992)

$$d_{\mathrm{TV}}\left(\mathcal{L}(U), \mathcal{L}(V)\right) = \sup_{D \subset E} \left|\mathbb{P}(U \in D) - \mathbb{P}(V \in D)\right|. \tag{3.2}$$

The subsets D are assumed to be measurable. For $E = \mathbb{N}$ all subsets $D \subseteq E$ are measurable and the total variation distance can be written as

$$d_{\mathrm{TV}}\left(\mathcal{L}(U), \mathcal{L}(V)\right) = \frac{1}{2} \sum_{i \geq 0} \left|\mathbb{P}(U = i) - \mathbb{P}(V = i)\right|$$

Typically, we measure the total variation distance between a sum of Bernoulli variables $U = \sum_{i \in I} U_i$ from a finite integer index set $I \subset \mathbb{N}$ and a (compound) Poisson approximation $\mathcal{P}(\vartheta)$ with $\vartheta = \sum_{i \in I} \mathbb{E}[U_i]$. The main idea is to define for each Bernoulli variable U_i a neighborhood set $B_i \subseteq I$ of variables which have strong dependencies on U_i. We require $i \in B_i$. Then, we obtain the Chen-Stein bound derived from Theorem 1 in Arratia *et al.* (1990)

$$d_{\mathrm{TV}}\left(\mathcal{L}(U), \mathcal{P}(\vartheta)\right) \leq b_1 + b_2 + b_3, \tag{3.3}$$

where

$$\begin{aligned} b_1 &= \sum_{i \in I} \sum_{j \in B_i} \mathbb{E}[U_i]\mathbb{E}[U_j], \\ b_2 &= \sum_{i \in I} \sum_{j \in B_i \setminus \{i\}} \mathbb{E}[U_i U_j], \\ b_3 &= \sum_{i \in I} \mathbb{E}|\mathbb{E}[U_i - \mathbb{E}[U_i]|\mathfrak{U}(U_j : j \notin B_i)]|. \end{aligned}$$

Here, $\mathfrak{U}(\cdot)$ denotes the σ-algebra generated by $U_j : j \notin B_i$. One tries to define a neighborhood such that neither b_1 nor b_2 are too large but also the antagonist b_3 is sufficiently small. If only local dependencies exist, one can catch all dependencies in the neighborhoods, thus, obtaining $b_3 = 0$. Furthermore, one can use the improved Chen-Stein bound (Barbour *et al.*, 1992)

$$d_{\mathrm{TV}}\left(\mathcal{L}(U), \mathcal{P}(\vartheta)\right) \leq \frac{1 - e^{-\vartheta}}{\vartheta} \left(b_1 + b_2\right). \tag{3.4}$$

3.2 Single Word Occurrences

3.2.1 Exact Count Distribution

First results on the exact distribution of the number of occurrences are retrieved by Gentleman and Mullin (1989). They assume an equi-probable i.i.d. sequence model. In this publication, the formulae are explicitly stated for any two-letter pattern of length 2 to 8. Later, algorithms are used to generate them (Gentleman, 1994). Most further research uses Markov chains as sequence models. Kleffe and Langbecker (1990) employ an automaton to compute the exact count distribution. Later, Nicodeme *et al.* (1999) extend this approach to create generating functions by automatons. Other approaches are based on language decomposition (Régnier and Szpankowski, 1998; Régnier, 2000; Régnier and Denise, 2004) or probabilistic arithmetic automata (Marschall and Rahmann, 2008). Eventually, the very recent conditional approach (Zhang *et al.*, 2007) is motivated by optimally spaced seeds for

homology search (Ma *et al.*, 2002). Even though this approach is specifically designed for PFMs, it is a word counting approach and we think it is relevant to introduce it here. Furthermore, it is the only approach, which is straight-forward enough to extend to multiple words.

Expected Value

The expected value is easily computed because the position dependencies have no influence. Since the probability of an occurrence at one position is $\mu(w)$, the expected number of occurrences on a sequence of length n is

$$\mathbb{E}[N_n(w)] = (n - \ell + 1) \cdot \mu(w).$$

Variance

Due to position dependencies, the variance for the number of occurrences is more complicated to compute. First, we decompose the variance of the number of occurrences into the (co-)variances of the occurrence indicator random variables:

$$\mathbb{V}[N_n(w)] = \mathbb{V}[\sum_{i=1}^{n-\ell+1} Y_i(w)] = \sum_{i=1}^{n-\ell+1} \mathbb{V}[Y_i(w)] + 2 \cdot \sum_{i=1}^{n-\ell+1} \sum_{j=i+1}^{n-\ell+1} \mathbb{C}\mathrm{ov}[Y_i(w), Y_j(w)]. \quad (3.5)$$

Applying the definition of the variance $\mathbb{V}[Y_i(w)] = \mathbb{E}[Y_i(w)^2] - \mathbb{E}[Y_i(w)]^2$ yields for the terms in the first sum $\mu(w) - \mu(w)^2$. The covariances can be computed using $\mathbb{C}\mathrm{ov}[Y_i(w), Y_j(w)] = \mathbb{E}[Y_i(w) \cdot Y_j(w)] - \mathbb{E}[Y_i(w)]\mathbb{E}[Y_j(w)]$. The second part, again, yields $\mu(w)^2$ while the first part is more difficult to compute. If the occurrences at position i and j do not overlap, thus, $Y_i(w)$ and $Y_j(w)$ are independent, it also yields $\mu(w)^2$ diminishing the whole term. In case of an overlap, we have to compute the joint probability of one occurrence at position i and another occurrence at position j. For this purpose, we define the overlap bit for $0 \leq d < \ell$ which indicates whether an overlap for a certain distance is possible (Guibas and Odlyzko, 1980; Li, 1980; Guibas and Odlyzko, 1981a):

$$\epsilon_d(w) := \begin{cases} 1 & \text{if } \forall_{0<\kappa\leq\ell-d} \quad w_{\kappa+d} = w_\kappa, \\ 0 & \text{otherwise.} \end{cases} \quad (3.6)$$

For convenience, we define the overlap bit $\epsilon_d(w) = 1$ for $d \geq \ell$. The overlap bit can be efficiently computed in linear time using the Z-algorithm (Gusfield, 1997, Chapter 1).

Example 3.1. *The overlap bit for the word $w =$ 'ACA' is given by*

d	A	C	A			$\epsilon_d(w)$
0	A	C	A			1
1		A	C	A		0
2			A	C	A	1

□

We can compute the probability $\gamma_d(w)$ for joint occurrences with distance $0 \leq d < \ell$ by

$$\gamma_d(w) := \mathbb{P}_\mu(Y_i(w) = 1, Y_{i+d}(w) = 1) = \mu(w)\epsilon_d(w) \cdot \prod_{\kappa=\ell-d+1}^{\ell} \mu(w_\kappa). \tag{3.7}$$

In Eq. (3.5), we replace the sums over the position i and j by a sum over the distance d between i and j and apply above equation:

$$\mathbb{V}[N_n(w)] = (n-\ell+1)[\mu(w) - \mu(w)^2] + 2\sum_{d=1}^{\ell-1}(n-\ell-d+1)\left[\gamma_d(w) - \mu(w)^2\right]. \tag{3.8}$$

This formula demonstrates that the variance depends on the probability of an occurrence ($\mu(w)$) and the possibilities of self-overlaps ($\epsilon_d(w)$). The higher the probability of an occurrence, the more occurrences one would expect. For a word with no self-overlap, $\epsilon_d(w)$ is 0 for $d > 0$. Hence, only $-\mu(w)$ is left in the right term of the last equation. Therefore, the variance of the number of occurrences becomes smaller.

Example 3.2. *We consider the words v = 'GCCAA' and w = 'CGCGC' in an i.i.d. sequence with equi-probable nucleotide distribution of length $n = 10000$. Obviously, the word w is highly self-overlapping in contrast to v. Hence, if the word w occurs at one position, it is very likely to see a subsequent occurrence either two or four positions farther. We call such a congregation of overlapping hits a clump with the additional requirement that no two clumps can overlap. Intuitively, one might conclude that w occurs more often in the sequence than v. However, due to the equi-probable nucleotide distribution, the expected value of the number of counts is the same for both words, namely 9.8. Thus, the distance between two clumps of w has to be larger than for v. Therefore, hits of v occur more equidistant than hits of w leading to a smaller variance of the number of counts for v. Indeed, the variance for v is 9.7 while the variance for w is 11.* □

Count Distribution

The exact distribution of the number of occurrences is usually computed using a duality between the number of occurrences $N_n(w)$ and the position of the mth occurrence $T_m(w)$ called occurrence time. If the mth occurrence is before or at position $n-\ell+1$, the number of occurrences is equal to or greater than m. Thus, we obtain the duality $N_n(w) \geq m \Leftrightarrow T_m(w) \leq n-\ell+1$. Hence, under the random sequence model μ, we obtain

$$\begin{aligned} \mathbb{P}_\mu(N_n(w) = m) &= \mathbb{P}_\mu(T_m(w) \leq n-\ell+1) - \mathbb{P}_\mu(T_{m+1}(w) \leq n-\ell+1) \\ &= \sum_{i=1}^{n-\ell+1} \mathbb{P}_\mu(T_m(w) = i) - \sum_{i=1}^{n-\ell+1} \mathbb{P}_\mu(T_{m+1}(w) = i). \end{aligned} \tag{3.9}$$

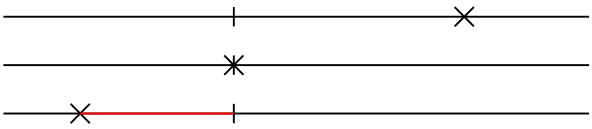

Figure 3.1: The event of an occurrence of a word at position i (vertical bar) on a sequence (horizontal line) can be decomposed into three disjoint events regarding the mth occurrence (marked by a cross): (a) the occurrence at i occurs before the mth occurrence, (b) the mth occurrence is at position i, (c) the mth occurrence is at position $i - d$ where the distance d is indicated in red.

Now, we are left with computing the distribution for the occurrence time. We can decompose the event $\{Y_i = 1\}$ which has probability $\mu(w)$ into three parts (for an illustration, see Fig. 3.1):

- The occurrence at position i is not yet the mth occurrence. Hence, position i is either the first, the second, ..., or the $m-1$th occurrence.
- The occurrence at position i is exactly the mth occurrence.
- The mth occurrence has already been occurred. Hence, there is a distance d between the mth occurrence and the position i.

We can write this decomposition by

$$\{Y_i(w) = 1\} \equiv \bigcup_{k=1}^{m-1} \{T_k = i\} \cup \{T_m = i\} \cup \bigcup_{j=1}^{i-1} \{T_m = j \cap \{Y_i(w) = 1 | Y_j(w) = 1\}\}. \quad (3.10)$$

Substituting the events by the corresponding probabilities, we obtain

$$\mu(w) = \sum_{k=1}^{m-1} \mathbb{P}_\mu(T_k = i) + \mathbb{P}_\mu(T_m = i) + \sum_{j=1}^{i-1} \mathbb{P}_\mu(T_m = j)\mathbb{P}_\mu(Y_i(w) = 1 | Y_j(w) = 1).$$

In the last sum, the conditional probability is equal to $\mu(w)$ for all j if the occurrence at position i does not overlap with j. For all other cases, we can use the overlap bit ϵ. Solving for $\mathbb{P}_\mu(T_k = i)$ leads to the recurrence formula

$$\begin{aligned} \mathbb{P}_\mu(T_m = i) &= \mu(w) - \sum_{k=1}^{m-1} \mathbb{P}_\mu(T_k = i) - \sum_{j=1}^{i-\ell} \mathbb{P}_\mu(T_m = j)\mu(w) \\ &\quad - \sum_{j=i-\ell+1}^{i-1} \mathbb{P}_\mu(T_m = j)\epsilon_{\ell-(i-j)}(w) \prod_{\kappa=\ell-(i-j)}^{\ell} \mu(w_\kappa). \end{aligned} \quad (3.11)$$

Thus, we can recursively compute the occurrence time probabilities and, therefore, also the exact count distribution in Eq. (3.9). Unfortunately, the computation is time consuming.

We have to fill a matrix of size $O(mn)$ and each entry takes $O(m+(n-\ell)+\ell^2)$ time leading to a complexity of $O(mn(m+n+\ell^2))$. Using hash tables for the intermediate sums and the product, one can decrease the complexity to $O(mn)$ at the expense of memory usage. However, using generating functions simplifies the recurrence by removing many terms (see Section 4). This allows a faster computation. Nevertheless, complexity always depends on the sequence length. Approximations of the count distribution can circumvent this problem as we will see below the following example.

Example 3.3. *Again, we consider the words v = 'GCCAA' and w = 'CGCGC' in an i.i.d. sequence with equi-probable nucleotide distribution of length $n = 10000$. Figure 3.2 shows the exact count densities and distribution functions for both words. The upper panel contains the densities. At a first glimpse, both trajectories look similar. Their maximum is at 9. However, the probability at the maximum is smaller for w while the corresponding probabilities for less than seven and more than 14 hits are higher. Thus, it has a higher variance as the last example already concluded. The lower panel of Fig. 3.2 depicting the distributions confirms this observation since the distribution function for w is not as steep as for v.*

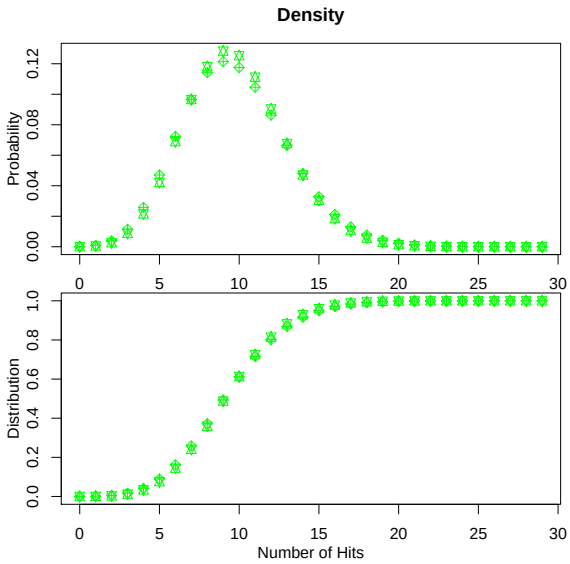

Figure 3.2: Densities (upper panel) and distribution (lower panel) functions of the number of hits for the word 'GCCAA' (circles) and 'CGCGC' (crosses) in an i.i.d. sequence with equi-probable nucleotide distribution and length $10,000$.

Although the distributions look very similar, the tails of the distribution, which are important for p-value calculations, differ almost in one order of magnitude. Therefore, the p-value to observe at least 29 hits of v is $2.8 \cdot 10^{-07}$ while the corresponding p-value for w is $1.9 \cdot 10^{-06}$. Differences between both distributions are due to the self-overlap of w leading to strong position dependencies. This illustrates that incorporation of these dependencies is highly important. □

Conditional Approach

In Zhang *et al.* (2007), a new approach for computating the exact count distribution is proposed. The two main ideas are to compute the probability for the number of occurrences recursively for smaller regions and conditioning on sequence prefixes of the smaller region. The published approach uses a Markov sequence model of order 1. Hence, we simplify the approach to the i.i.d. model. First, we compute the probability to observe at least one occurrence. In a second step, we extend this to deal with at least k occurrences which is equivalent to the count distribution.

At Least One Occurrence Let $F_i(w)$ denote the probability $\mathbb{P}_\mu(N_{n-i+1}(w) \geq 1)$ to observe at least one occurrence of w in a sequence of length $n-i+1$. We compute this probability by conditioning on a given prefix

$$F_i(w) := \mathbb{P}_\mu(N_{n-i+1}(w) \geq 1) = \sum_{a \in \mathfrak{A}} \mathbb{P}_\mu(N_{n-i+1}(w) \geq 1 | X_1 = a)\mu(a) = \sum_{a \in \mathfrak{A}} f_{i,a}(w)\mu(a), \tag{3.12}$$

where $f_{i,a}(w)$ denotes the conditional probability of $\{N_{n-i+1}(w) \geq 1 | X_1 = a\}$ with $a \in \mathfrak{A}$. One can derive these conditional probabilities by incrementing the length of the given prefix. Thus, we extend our definition to $f_{i,v}(w) := \mathbb{P}_\mu(N_{n-i+1}(w) \geq 1 | X_1 \dots X_{|v|} = v)$ with $v \in \mathfrak{A}^+$ such that it can also deal with words. This yields

$$f_{i,v}(w) = \sum_{a \in \mathfrak{A}} f_{i,va}(w)\mu(a), \tag{3.13}$$

where va denotes the concatenation of v and a. However, one does not need to compute $f_{i,v}(w)$ for all $w \in \mathscr{P}(\mathfrak{A}^n)$ where $\mathscr{P}(\cdot)$ denotes the power set. As long as $|v| \leq |w|$, an occurrence of w at position 1 is only possible if v matches a prefix of w. Furthermore, if no suffix of v matches a prefix of w for $|v| \leq |w|$ there cannot be an occurrence of w before position $i + |v|$. In these cases, the probability for $\{N_{n-i+1}(w) > 0 | X_1 \dots X_{|v|} = v\}$ is equal to the probability for $\{N_{n-i+|v|+1}(w) > 0\}$. Next, we incorporate the possibility of a suffix of v which matches the prefix of w. This yields with u being the longest suffix of v which is also a prefix of w and $i' = i + |v| - |u|$

$$f_{i,v}(w) = \begin{cases} F_{i'}(w) & \text{if } |u| = 0, \\ f_{i',u}(w) & \text{otherwise.} \end{cases}$$

Thus, if u is not the empty word ($|u| = 0$) we consider u as a prefix of the sequence region. Finally, if $v = w$ the probability conditioned on v being a prefix is 1. Furthermore, for small i the word w is longer than the sequence region. Hence, we obtain two stop criteria for the recursion

$$f_{i,v}(w) = \begin{cases} 0 & \text{if } n - i + 1 < |w|, \\ 1 & \text{if } w = v. \end{cases}$$

Therefore, $|v| > |w|$ does not need to be considered in the recursive formula. In addition, one can employ a dynamic programming algorithm to compute $f_{i,v}(w)$ and $F_i(w)$ efficiently (for details, see Zhang *et al.*, 2007).

At Least k occurrences We can extend this approach to compute the probability to observe at least k occurrences. We define

$$F_i^{(k)}(w) := \mathbb{P}_\mu(N_{n-i+1}(w) \geq k) = \sum_{a \in \mathfrak{A}} f_{i,a}^{(k)}(w)\mu(a), \tag{3.14}$$

with $f_{i,v}^{(k)}(w) = \mathbb{P}_\mu(N_{n-i+1}(w) \geq k | X_1 \dots X_{|v|} = v)$ for $v \in \mathfrak{A}^+$. In this recursive formula, we only have to change Eq. (3.13) to

$$f_{i,v}^{(k)}(w) = \begin{cases} \sum_{a \in \mathfrak{A}} f_{i,va}^{(k-1)}(w)\mu(a) & \text{if } w = va, \\ \sum_{a \in \mathfrak{A}} f_{i,va}^{(k)}(w)\mu(a) & \text{otherwise.} \end{cases} \tag{3.15}$$

Thus, we compute the event of at least k occurrences by the events of $k-1$ occurrences in smaller regions and conditioned on a prefix of the sequence region.

3.2.2 Position Independence

The consideration of position dependencies between the occurrence indicators $Y_i(w)$ is the main reason for the high complexity of the exact count distribution. Ignoring these dependencies leads to very simple approximations which are fast to compute but not very accurate. Here, we present the binomial approximation and its asymptotic counterpart, the Poisson approximation.

Binomial Distribution

Assuming all $Y_i(w)$ to be independent, then the count distribution $N_n(w)$ is a sum of $n - \ell + 1$ i.i.d. Bernoulli random variables with success probability $\mu(w)$ as parameter. Hence, the probability to observe m occurrences on a sequence of length $n - \ell + 1$ implies that $n-\ell-m+1$ positions do not have an occurrence. Considering all possible permutations of the positions of the occurrences attains

$$\mathbb{P}_\mu(N_n(w) = m) = \binom{n-\ell-1}{m} \mu(w)^m \left(1 - \mu(w)\right)^{n-\ell-m+1}$$

which is the binomial distribution with parameters $n - \ell + 1$ and $\mu(w)$.

Poisson Distribution

Assuming that the length of the word w is small in comparison to the sequence length, we can substitute $n-\ell+1$ by n. Then, the binomial distribution converges to a Poisson distribution if the expected number of occurences $\mu(w)n$ remains constant ($O(1)$) for sequence length $n \to \infty$. This implies that $\mu(w) = O(n^{-1})$ and $\log n = O(\ell)$. We obtain the Poisson distribution for the number of counts with $\vartheta = (n-\ell+1)\mu(w)$

$$\mathbb{P}_\mu\left(N_n(w) = m\right) = \frac{(\vartheta)^m\, e^{\vartheta}}{m!}.$$

Example 3.4. *In Fig. 3.3, several count distributions are shown for the words v = 'GCCAA' and w = 'CGCGC' in an i.i.d. sequence with equi-probable nucleotide distribution of length $n = 10000$. The exact distribution serves as reference. For the non-self-overlapping word v, the binomial as well as the Poisson distribution yield an accurate approximation. In general, both distributions are very similar showing that the sequence length is sufficiently long such that the words occur rarely. In contrast to the approximation for v, the probabilities for w do not match the exact distribution but are equal to the values for v. Since we assume an equi-probable nucleotide distribution, both words have the same occurrence probability. As neither the binomial nor the Poisson approximation is capable to consider the self-overlap, it is not surprising that both words obtain the same distribution.*

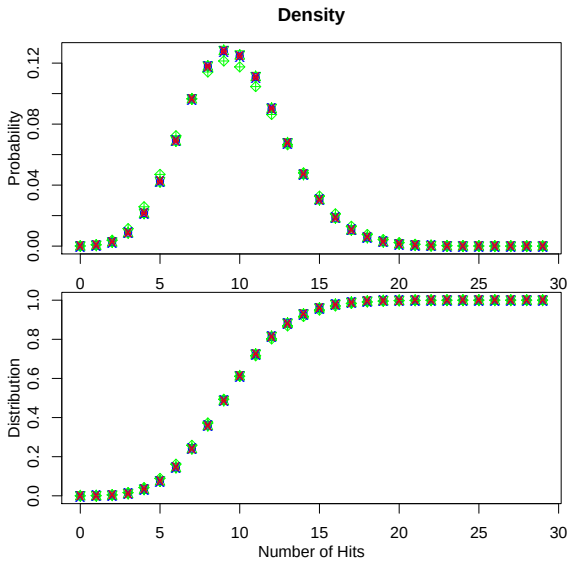

Figure 3.3: Densities (upper panel) and distribution (lower panel) of the number of hits for the word 'GCCAA' (circles) and 'CGCGC' (crosses) in an i.i.d. sequence with equi-probable nucleotide distribution and length $10,000$. Green indicates the exact calculation while blue labels the binomial approximation and red corresponds to the Poisson approximation.

Again, the differences between the distributions are very subtle but can be important for significance calculation. The p-value for $N_i(v) \geq 29$ for the binomial approximation is $3.2 \cdot 10^{-07}$ and for the Poisson approximation $3.2 \cdot 10^{-07}$ in comparison to the true value of $2.8 \cdot 10^{-07}$. Already for the non self-overlapping word, the p-values are underestimated. For the word w, the differences are more significant: The p-values from the binomial and

the Poisson distribution do not change as discussed above but the exact distribution yields a p-value of $1.9 \cdot 10^{-06}$*. Hence, the p-value is underestimated by an order of magnitude.* □

Chen-Stein Error Bounds The assumption of independence does not hold. Hence, the approximation contains an error. Using the Chen-Stein method, we can estimate the error. Robin *et al.* (2005) compute the Chen-Stein error bounds for non-overlapping words but for higher-order Markov models. Here, we give the explicit bounds for any word w but only in an i.i.d. sequence. From the above derivation, we know that the Poisson distribution has mean ϑ. Hence, we compute the total variation distance (see Eq. (3.2)) of

$$d_{\mathrm{TV}}\left(\mathcal{L}(N_n(w)), \mathcal{P}(\vartheta)\right) \leq \frac{1 - e^{-\vartheta}}{\vartheta} \left(b_1 + b_2\right).$$

Based on the index set $I = \{1, \ldots, n\}$, we define the set B_i of the neighborhood to contain all depending random variables. Since the word w has length ℓ, the set B_i for $Y_i(w)$ contains $\{i-\ell+1, i-\ell+2, \ldots, i+\ell-1\}$ for $\ell \leq i \leq n-\ell+1$. The other neighborhood sets are smaller due to boundary effects at the beginning and the end of the sequence. For convenience, we ignore this such that $|B_i| = 2\ell - 1$ for all i. The first bound b_1 is easy to compute:

$$b_1 = \sum_{i \in I} \sum_{j \in B_i} \mathbb{E}[Y_i(w)]\mathbb{E}[Y_j(w)] = (n - \ell + 1)(2\ell - 1)\mu(w)^2.$$

The second bound b_2 contains the second moments between Y_i which we have already defined to be $\gamma_d(w)$ in Eq. (3.7). Hence, we obtain

$$b_2 = \sum_{i \in I} \sum_{j \in B_i \setminus \{i\}} \mathbb{E}[Y_i(w)Y_j(w)] = 2 \sum_{i \in I} \sum_{d=1}^{\ell-1} \gamma_d(w) = 2(n - \ell + 1)(G(w) - \mu(w))$$

where the last step is done by defining $G(w) = \sum_{d=0}^{\ell-1} \gamma_d(w)$.

Now, we can analyze the asymptotics of the boundary under the assumptions of the Poisson approximation. Since $\log n = O(\ell)$ and $\mu(w) = O(n^{-1})$, we obtain for $b_1 = O(n^{-1} \log n)$, thus, $\lim_{n \to \infty} b_1 = 0$. The bound b_2 is zero if the word w is not self overlapping. Since $\vartheta^{-1}(1 - e^{-\vartheta}) \in [0, 1]$, the total variation distance $d_{\mathrm{TV}}(\mathcal{L}(N_n(w)), \mathcal{P}(\vartheta))$ tends to zero for non-overlapping words. For overlapping words, we can bound b_2 only by $O(\ell)$. In these cases, we can still explicitly compute the total variation distance to correct significance values. Thus, we can state that self-overlaps violate the requirements for an accurate Poisson approximation since such words occur in clumps. Then, a more appropriate model for the number of occurrences is a compound Poisson distribution (see Section 3.3.3) or one can use a Poisson distribution for the number of clumps (see Section 3.3.2).

3.2.3 Normal Approximation

The Poisson distribution is based on the assumption that the word only occurs rarely ($\mu(w)n = O(1)$) in the sequence. For very short words, this is not true. In this case, one can use a central limit theorem for a normal approximation (for a proof, see Waterman, 2000, Chapter 12). Here, one assumes that the expected number of occurrences tends to infinity with sequence length $n \to \infty$. Since the normal distribution is completely defined by the expected value and the variance, it suffices to compute both parameters asymptotically for $n \to \infty$. Computation of the asymptotic mean is straight-forward:

$$\lim_{n\to\infty} n^{-1}\mathbb{E}[N_n(w)] = \lim_{n\to\infty} n^{-1}\mathbb{E}[\sum_{i=1}^{n-\ell+1} Y_i(w)] = \mu(w).$$

For the variance, we start with the exact formula in Eq. (3.8). With $n \to \infty$ one obtains

$$\lim_{n\to\infty} n^{-1}\mathbb{V}[N_n(w)] = \mu(w) - \mu(w)^2 + 2\sum_{d=1}^{\ell-1}\gamma_d(w) - 2(\ell-1)\mu(w)^2.$$

Since $\gamma_0(w) = \mu(w)$, we can move the first term into the sum and accordingly change the index of the sum from $d = 1$ to $d = 0$. Due to the factor 2 we have to subtract $\mu(w)$, as well. Similarly, we proceed with $\mu(w)^2$ obtaining

$$\lim_{n\to\infty} n^{-1}\mathbb{V}[N_n(w)] = 2\sum_{d=0}^{\ell-1}\gamma_d(w) - 2\ell\mu(w)^2 - \mu(w)(1-\mu(w)).$$

The last term $-\mu(w)(1-\mu(w))$ stems from $-\mu(w)$ which we had to add for the index shift and the $\mu(w)^2$ to change the factor $\ell - 1$ to ℓ. Considering multiple words, we will see that this transformation is convenient. Substituting $G(w)$ into the equation, we obtain for the asymptotic variance

$$\sigma^2 := \lim_{n\to\infty} n^{-1}\mathbb{V}[N_n(w)] = 2G(w) - 2\ell\mu(w)^2 - \mu(w)(1-\mu(w)).$$

The number of counts $n^{-1}N_n(w)$ is now asymptotically normal distributed with mean $\mu(w)$ and variance $n^{-1}\sigma^2$. If $\sigma \neq 0$ we obtain asymptotically

$$\frac{\sqrt{n}\left(n^{-1}N_n(w) - \mu(w)\right)}{\sigma} \sim \mathcal{N}(0,1).$$

This approximation assumes long sequences and that the word occurs frequently. Due to the reduction to a standard normal distribution, one can apply known statistics.

Example 3.5. *The words v = 'GCCAA' and w = 'CGCGC' on an i.i.d. sequence with equi-probable nucleotide distribution have respectively an asymptotical mean of* 0.00098 *and* 0.00098 *and an asymptotic variance of* 0.00097 *and* 0.0011*. Thus, the variance for the self-overlapping word w is higher than for v. As we have seen using the exact approach, this is correct. Hence, the parameters of the normal approximation do reflect position dependencies. Multiplying both the asymptotic mean and the asymptotic variance by the sequence length $n = 10,000$, obtains the two parameters for the count distribution based on a normal approximation. Figure 3.4 compares the approximation with the exact approach. In general, the approximations differ from the exact solution. However, for both words the approximations have similar accuracy due to the incorporation of the position dependencies.*

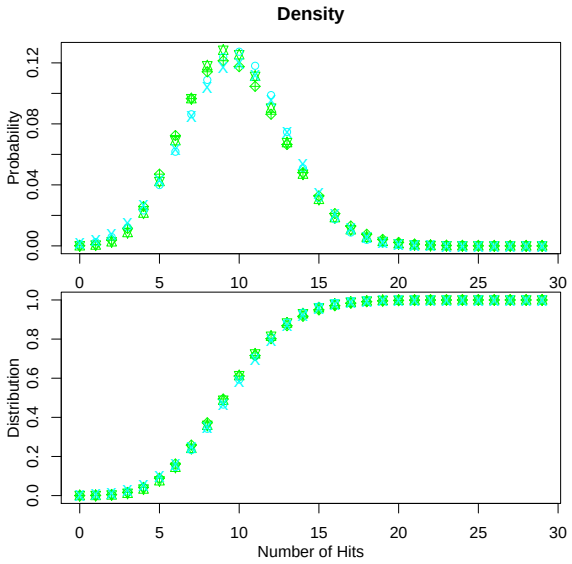

Figure 3.4: Densities (upper panel) and distribution (lower panel) of the number of hits for the word 'GCCAA' (circles) and 'CGCGC' (crosses) in an i.i.d. sequence with equi-probable nucleotide distribution and length $10,000$. Green indicates the exact calculation while light blue labels the normal approximation after applying the continuity correction.

The differences also influence the significance values. One retrieves a p-value of $7.5 \cdot 10^{-10}$ *to observe at least* 29 *occurrences of v. In comparison to the exact p-value* $2.8 \cdot 10^{-07}$*, there is a difference of several orders of magnitude. For word w, the difference is similar (p-value for normal approximation is* $6.5 \cdot 10^{-09}$ *and the exact one is* $1.9 \cdot 10^{-06}$*). Again, the accuracy does not change between v and w. However, especially the tails of the distribution are significantly underestimated since the word only occurs rarely. This can be explained as the normal approximation is not accurate in the tail.* □

3.3 Clumps for Single Words

In the last section, we have seen that self-overlapping words tend to have overlapping aggregations of occurrences. We call such an overlapping aggregation a *clump* (for an illustration, see Fig. 3.5). Especially, the Poisson approximation encounters problems with

words occurring in clumps. There are two solutions within the Poisson regime. First, one can consider clumps instead of occurrences. Due to our definition, clumps cannot overlap. Thus, a Poisson approximation for the distribution of the number of clumps yields accurate results. Second, one can explicitly model the clumps using a compound Poisson distribution.

Figure 3.5: The left panel contains four occurrences of a word (boxes) on a sequence (straight line) which are organized into two clumps with size three respectively one. The right panel illustrates the occurrences of a different word with the same number of clumps and clump sizes.

After presenting a formal definition of clumps, we discuss the difficulties in dealing with clumps. Then, we show how to compute the exact distribution of the number of clumps. Subsequently, we describe a simple binomial/Poisson approximation and extend the latter one to a more sophisticated Poisson approximation. Then, we introduce the compound Poisson distribution to model clumps explicitly. Finally, we give a asymptotic normal approximation.

Formal Definition Formally, we define a clump to start at an occurrence which does not overlap to the left with another occurrence. The size of a clump is the number of overlapping occurrences within one clump (see Fig. 3.5). We introduce an indicator random variable $\tilde{Y}_i(w)$ for the start of an occurrence of a clump of word w at position i by

$$\tilde{Y}_i(w) := Y_i(w) \prod_{d=1}^{\ell-1} (1 - Y_{i-d}(w)) \,. \tag{3.16}$$

The probability of $\{\tilde{Y}_i(w) = 1\}$ is not easy to compute due to the dependencies of the $Y_i(w)$s. We denote the number of clumps in an i.i.d. sequence of length n by $\tilde{N}_n(w) = \sum_{i=1}^{n-\ell+1} \tilde{Y}_i(w)$. The random indicator $\tilde{Y}_1$ involves the usually unobservable letters X_i for $-\ell < i < 1$. We ignore those since the corresponding error can be efficiently bounded in terms of the total variation distance for rare words on long sequences (Reinert and Schbath, 1999).

Intuitively, it might appear easier to deal with clumps instead of word occurrences due to the removal of self-overlaps. Instead, clumps are more difficult to handle. We present some difficulties and introduce notation to formalize them.

(Principal) Periods By its definition, a clump occurs if there are no preceding overlapping occurrences of the word. Thus, one can compute the probability of a clump occurrence by considering all possible preceding overlaps. Since some overlaps induce further overlaps, the problem is to consider each possible overlap exactly once. To capture this notion of overlap, we introduce the concept of the *period* and *principal period*.

Example 3.6. *The word 'CGCGC' can overlap with itself at a distance of two and four. For an overlap of distance four, the overlap for a distance of two is automatically induced:*

	$i-4$	$i-3$	$i-2$	$i-1$	i	$i+1$	$i+2$	$i+3$	$i+4$
occurrence at position i					C	G	C	G	C
overlapping occurrence	C	G	C	G	C				
induced occurrence			C	G	C	G	C		

□

The period of a word w is a distance $0 < \eta < \ell$ such that an occurrence at i can overlap with an occurrence at $i + \eta$. The set $\Upsilon(w)$ of all periods is defined by

$$\Upsilon(w) := \{\eta : \epsilon_\eta(w) = 1, 0 < \eta < \ell\}. \tag{3.17}$$

Obviously, if $\Upsilon(w) = \{\}$ then w is not self-overlapping. As we have seen in the last example, overlaps at some periods induce further occurrences. To remove such periods, we define the set $\Upsilon'(w)$ of principal periods containing all periods which are not multiples of the minimal period. The *root* of a period is the non-overlapping prefix of the preceding occurrence (for a more formal discussion, see Lothaire, 1983; Rivals and Rahmann, 2003).

Example 3.7. *The word $w =$ 'CGCGC' has periods $\Upsilon(w) = \{2, 4\}$ and the set of principal periods $\Upsilon'(w) = \{2\}$ since the period $\eta = 4$ is a multiple of the minimal period $\eta = 2$. The (principal) root of $\eta = 2$ is 'CG'.* □

Probability of a Clump The probability $\tilde{\mu}(w) := \mathbb{P}_\mu(\tilde{Y}_i(w) = 1)$ of a clump at one position can be computed by considering the probabilities for preceding overlapping occurrences (Schbath, 1995a,b) - namely using a probability $\omega(w)$ for self-overlap. A self-overlap occurs if any of the principal roots occur just before an occurrence of w. Since two different principal roots can never occur at the same time (Schbath, 1995a), the events are disjoint and their probabilities can be summed up:

$$\omega(w) := \mathbb{P}_\mu(\tilde{Y}_i(w) = 0 | Y_i(w) = 1) = \sum_{\eta \in \Upsilon'(w)} \prod_{\kappa=1}^{\eta} \mu(w_\kappa). \tag{3.18}$$

A clump occurs if there is an occurrence of w and no preceding self-overlapping occurrences. Hence, the probability $\tilde{\mu}(w)$ for a clump is the probability of an occurrence $\mu(w)$ and no self-overlap $1 - \omega(w)$ yielding

$$\tilde{\mu}(w) := \mathbb{E}[\tilde{Y}_i(w)] = \mathbb{P}_\mu(\tilde{Y}_i(w) = 1 | Y_i(w) = 1)\mathbb{P}_\mu(Y_i(w) = 1) = [1 - \omega(w)]\,\mu(w). \tag{3.19}$$

As the second term is always ≤ 1, the probability of a clump must be smaller than the probability of a word occurrence.

Example 3.8. *We consider the word w = 'ACA' in a two-letter alphabet with μ('A') = $p \in (0,1)$ and μ('C') = $1-p$. The probability for w is $\mu(w) = (1-p)p^2$. A clump of w only starts if there are no preceding overlapping occurrences. Since $\Upsilon(w) = \{2\}$ which is also the set of principal periods, an overlap can only occur two positions before, more precisely if there is an 'AC'. 'AC' occurs with a probability of μ('AC') = $(1-p)p$. We obtain $\omega(w) = (1-p)p$ as there are no further overlap possibilities. The probability of the clump is $\tilde{\mu}(w) = [1-\omega(w)]\mu(w) = (p^2-p^3)(1-p+p^2)$.* □

More Dependencies The indicator $Y_i(w)$ for a word occurrence only depends on the indicators $Y_{i+d}(w)$ for $|d| < \ell$ which can form an overlapping occurrence. All other $Y_j(w)$ are independent of $Y_i(w)$. In contrast, clumps cannot overlap due to their definition. Hence, the joint probabilities $\mathbb{P}_\mu(\tilde{Y}_i(w) = 1, \tilde{Y}_{i+d}(w) = 1) = 0$. Since the definition for $\tilde{Y}_i(w)$ (see Eq. (3.16)) comprises the $\ell - 1$ preceding positions of i, those positions can overlap with another clump and, therefore, introduce dependencies between $\tilde{Y}_i(w)$ and $\tilde{Y}_j(w)$ for $0 < |i-j| < 2\ell - 1$. In other words, one only ensures independence if there is a gap of at least $\ell - 1$ positions between the end of the preceding clump and the start of the next clump.

Example 3.9. *We consider the same word as in the last example, w = 'ACA'. We show that the clump indicators $\tilde{Y}_i(w)$ and $\tilde{Y}_{i-d}(w)$ are not independent for $0 < d < 2\ell - 1$ with $\ell = 3$.*

In the last example, we concluded that no 'AC' is allowed in front of a clump of w. We denote this by an 'X' and 'Y', which are not allowed to be 'A' respectively 'C' at the same time. The occurrences of w depending on d are shown here:

d	$i-6$	$i-5$	$i-4$	$i-3$	$i-2$	$i-1$	i	$i+1$	$i+2$
					X	Y	A	C	A
1				X	Y	A	C	A	
2			X	Y	A	C	A		
3		X	Y	A	C	A			
4	X	Y	A	C	A				

For $0 < d < \ell$, two clumps at $i-d$ and i overlap, thus, the probability for the joint event is 0 and, therefore, not the product of the single event probabilities $\tilde{\mu}(w)^2 > 0$. Considering $d = 3$, still the positions for 'XY' which are supposed to be unequal to 'AC' do overlap with 'CA' from the preceding clump. Hence, the position cannot be 'AC', thus, we obtain the joint probability $\tilde{\mu}(w)\mu(w) \neq \tilde{\mu}(w)^2$. For $d = 4$, 'X' overlaps with the last 'A' from the preceding occurrence. Thus, only 'Y' is not allowed to be a 'C' which has probability $1-p$. We obtain for the joint probability $\tilde{\mu}(w)(1-p)\mu(w) \neq \tilde{\mu}(w)^2$. In contrast, $d > 3$ does not imply any letters for 'X' or 'Y'. Hence, we have to prohibit 'AC' for 'XY' and obtain for the joint probability $\tilde{\mu}(w)(1-\omega)\mu(w) = \tilde{\mu}(w)^2$. The main difficulty in the computation of the exact count distribution is the caluclation of the joint events for $\ell \leq d < 2\ell$ □

3.3.1 Exact Distribution

The exact count distribution of the number of clumps has only recently been published (Stefanov *et al.*, 2007) for Markov sequences of order one derived by generating functions. To our knowledge, there are no explicit formulae which are not based on generating functions in the literature. The same holds for the variance. Here, we derive such formulae for the exact variance and the exact word count distribution (see Section 3.2.1) in the i.i.d. sequence model. After stating the mean and the variance of the distribution, we present explicit recurrence formulae.

Expected Value

The expected value of number of clumps is based on the probability $\tilde{\mu}(w)$ to observe a clump at one position

$$\mathbb{E}[\tilde{N}_n(w)] = \mathbb{E}[\sum_{i=1}^{n-\ell+1} \tilde{Y}_i(w)] = (n-\ell+1)\tilde{\mu}(w). \tag{3.20}$$

Using Eq. (3.19) for the probability of the clump, the expected value can easily be computed.

Variance

The variance of the number of clumps is hard to compute due to the introduced dependencies by the requirement of no preceding overlapping occurrence (see Ex. 3.9). First, we derive the variance considering these dependencies. However, they only have minor influence on the results. Thus, we simplify the formulae by ignoring them in a second step. The variance can be written similar to Eq. (3.5) as

$$\mathbb{V}[\tilde{N}_n(w)] = \sum_{i=1}^{n-\ell+1} \mathbb{V}[\tilde{Y}_i(w)] + 2 \cdot \sum_{i=1}^{n-\ell+1} \sum_{j=i+1}^{n-\ell+1} \mathbb{C}\mathrm{ov}[\tilde{Y}_i(w), \tilde{Y}_j(w)].$$

The term in the first sum is $\tilde{\mu}(w) - \tilde{\mu}(w)^2$. The sum over the covariances now involves dependencies. First, we re-structure the sum such that the position of $\tilde{Y}_j(w)$s are expressed in terms of the distance d to $\tilde{Y}_i(w)$. This yields

$$\mathbb{V}[\tilde{N}_n(w)] = (n-\ell+1)\left(\tilde{\mu}(w) - \tilde{\mu}(w)^2\right) + 2 \cdot \sum_{i=1}^{n-\ell+1} \sum_{d=1}^{n-\ell-i+1} \mathbb{C}\mathrm{ov}[\tilde{Y}_i(w), \tilde{Y}_{i+d}(w)]. \tag{3.21}$$

The critical term is the covariance. We can decompose it according to the definition of the covariance:

$$\mathbb{C}\mathrm{ov}[\tilde{Y}_i(w), \tilde{Y}_{i+d}(w)] = \mathbb{P}_\mu(\tilde{Y}_i(w) = 1, \tilde{Y}_{i+d}(w) = 1) - \tilde{\mu}(w)^2. \tag{3.22}$$

For the number of word occurrences, the joint probability becomes $\mu(w)^2$ vanish the covariance for $d \geq \ell$ due to independence of the occurrence indicators. In the overlapping case, the joint probability is computed by the overlap probability $\gamma_d(w)$. This is different for clumps. Starting with the overlapping case, overlaps of clumps are not possible. Thus, the covariance becomes $-\tilde{\mu}(w)^2$. For $\ell \leq d < 2\ell$, the $\tilde{Y}_i(w)$ and $\tilde{Y}_{i+d}(w)$ are no longer independent due to the involvement of the preceding $Y_{i-d}(w)$ in the definition of $\tilde{Y}_i(w)$. They can overlap with the random indicators covered by the preceding clump. In these cases, we cannot use $1 - \omega(w)$ for no self-overlap but have to compute this probability explicitly. In fact, if the overlap is possible, we only have to skip the positions which are already covered by the preceding occurrence (see Ex. 3.9). The self-overlap probability $\omega_d(w)$ for a clump at $i + d$ and a preceding clump at i for $d \geq \ell$ is the probability of no clump at $i + d$ given the clump at i and an occurrence at $i + d$. In other words, $\omega_d(w)$ is the probability that an occurrence is no clump given a preceding clump. We obtain

$$\begin{aligned} \omega_d(w) &:= \mathbb{P}_\mu(\tilde{Y}_{i+d}(w) = 0 | Y_{i+d}(w) = 1, \tilde{Y}_i(w) = 1) \\ &= \sum_{\eta \in \Upsilon'(w)} \epsilon_{d-\eta}(w) \prod_{\kappa=1+\max(\eta-d+\ell,0)}^{\eta} \mu(w_\kappa) \end{aligned} \tag{3.23}$$

where $\epsilon_{d-\eta}(w)$ is equal to 0 if the word w does no allow such an overlap. The maximum ensures that we correctly incorporate principial periods, which do not overlap with the preceding occurrence. The periods are always smaller than the word length $\eta < \ell$. Thus, for $d \geq 2\ell$ we obtain $d - \ell > \eta$, thus, $\omega_d(w) = \omega(w)$. Furthermore, we obtain $\omega_d(w) := 1$ for $0 < d < \ell$ such that the complementary event, the occurrence at $i + d$ is a clump, has probability 0. We can write for the joint probability of a clump at i and $i + d$

$$\begin{aligned} \mathbb{P}_\mu(\tilde{Y}_{i+d}(w) = 1, \tilde{Y}_i(w) = 1) &= \mathbb{P}_\mu(\tilde{Y}_{i+d}(w) = 1 | Y_{i+d}(w) = 1, \tilde{Y}_i(w) = 1) \\ &\quad \cdot \mathbb{P}_\mu(Y_{i+d}(w) = 1 | \tilde{Y}_i(w) = 1)\tilde{\mu}(w) \\ &= [1 - \omega_d(w)]\mu(w)\tilde{\mu}(w). \end{aligned} \tag{3.24}$$

We can substitute the probability of an occurrence at $i + d$ given the clump at i by $\mu(w)$ since for $d < \ell$ the probability $1 - \omega_d(w) = 0$ and for $d \geq \ell$ the occurrence is independent of the preceding clump. Based on this formula, we can compute the covariance in Eq. (3.22) for $d > 0$ by

$$\begin{aligned} \mathbb{C}\mathrm{ov}[\tilde{Y}_i(w), \tilde{Y}_{i+d}(w)] &= \tilde{\mu}(w) \left[1 - \omega_d(w)\right] \mu(w) - \tilde{\mu}(w)^2 \\ &= \tilde{\mu}(w)\varpi_d(w) \end{aligned}$$

with $\varpi_d(w) := [1 - \omega_d(w)]\mu(w) - \tilde{\mu}(w)$. In case $\omega_d(w) = \omega(w)$, we obtain $\varpi_d(w) = 0$ and, thus, the covariance becomes 0. This occurs for large d since there no principal period is large enough that the corresponding preceding positions overlap with the preceding occurrence. We obtain for the variance

$$\mathbb{V}[\tilde{N}_n(w)] = (n-\ell+1)\left(\tilde{\mu}(w) - \tilde{\mu}(w)^2\right) + 2\tilde{\mu}(w) \sum_{i=1}^{n-\ell+1} \sum_{d=1}^{n-\ell-i+1} \varpi_d(w).$$

Although this is a compact formula, many terms in the sums yield 0 or $-\tilde{\mu}^2$. Hence, we can decompose the corresponding sums. Substituting $\omega_d(w) = \omega(w)$ for $d > 2\ell-1$ resepectively $\omega_d(w) = 0$ for $d < \ell$, yields $\varpi_d(w) = 0$ respectively $\varpi_d(w) = -\tilde{\mu}(w)$ leading to

$$\begin{aligned} \mathbb{V}[\tilde{N}_n(w)] &= (n-\ell+1)\left[\tilde{\mu}(w) - \tilde{\mu}(w)^2\right] \\ &\quad +2(n-3\ell+1)\tilde{\mu}(w)\left[\left(\sum_{d=\ell}^{2\ell-1} \varpi_d(w)\right) - (\ell-1)\tilde{\mu}(w)\right] \\ &\quad +2\tilde{\mu}(w)\left(\sum_{k=\ell}^{2\ell-1}\sum_{d=\ell}^{k} \varpi_d(w)\right) - 3\ell(\ell-1)\tilde{\mu}(w)^2. \end{aligned} \tag{3.25}$$

The first line corresponds to the variances of $\tilde{Y}_i(w)$, the second line contains the covariances for all i and j where $i - j > 2\ell - 1$. The third line summarizes the remaining covariances. This equation is easier to analyze as the previous equation in terms of asymptotics. We need this for the normal approximation (see Section 3.3.4).

Example 3.10. *We compare the mean and the variance of the clumps of the words $v =$ 'GCCAA' and $w =$ 'CGCGC' in an i.i.d. sequence with equi-probable nucleotide distribution of length $n = 10000$. The non self-overlapping word v has an empty set of periods. Hence, neither the expected value nor the variance should change for clumps in comparison to word occurrences. Indeed, $\mathbb{E}[N_n(v)] = \mathbb{E}[\tilde{N}_n(v)] = 9.8$ and $\mathbb{V}[N_n(v)] = \mathbb{V}[\tilde{N}_n(v)] = 9.7$. The situation differs for the self-overlapping word w with principal period 2. Here, the expected value for the number of clumps is $\mathbb{E}[\tilde{N}_n(w)] = 9.2$ where the expected number of occurrences is $\mathbb{E}[N_n(w)] = 9.8$. Since one clump contains one or more occurrences, the number of clumps on the sequence has to be lower in average than the number of occurrences. This is reflected by the lower expected value. The variance $\mathbb{V}[\tilde{N}_n(w)] = 9.1$ is also smaller than $\mathbb{V}[N_n(w)] = 11.0$ due to the missing self-overlap of the clump. It is also interesting to compare the values for v and w. The expected number of clumps for w is smaller than for v. The reason is that w has the same expected number of occurrences as v but occurs in clumps. Thus, the number of clumps has to be smaller while for v the number of clumps is equal to the number of occurrences.* □

Count Distribution

The count distribution of the number of clumps is derived similar to the exact count distribution of word occurrences. As for the variance calculation, the main differences are the additional dependencies. Therefore, also this expression becomes more complex. We consider the corresponding decomposition of Eq. (3.10) visualized in Fig. 3.1 for the occurrence of a clump. This means, we decompose the event of an occurrence of a clump instead of a word. Thus, we substitute the probability $\mu(w)$ by $\tilde{\mu}(w)$. Let $\tilde{T}_m$ denote the position of the mth clump. In the recurrence formula for $\mathbb{P}_\mu(T_m = i)$ in Eq. (3.11), we have to

substitute the word occurrences by the clump occurrences and consider the dependencies. Thus, we obtain

$$\mathbb{P}_\mu(\tilde{T}_m = i) = \tilde{\mu}(w) - \sum_{k=1}^{m-1} \mathbb{P}_\mu(\tilde{T}_k = i) - \sum_{j=1}^{i-\ell} \mathbb{P}_\mu(\tilde{T}_m = j)[1 - \omega_{i-j}(w)]\mu(w).$$

The formula for the number of counts is correspondingly:

$$\mathbb{P}_\mu(\tilde{N}_n(w) = m) = \sum_{i=1}^{n-\ell+1} \mathbb{P}_\mu(\tilde{T}_m(w) = i) - \sum_{i=1}^{n-\ell+1} \mathbb{P}_\mu(\tilde{T}_{m+1}(w) = i).$$

Using this formula, we can recursively compute the probability for the number of clumps. However, the drawbacks remain, which we already discussed for number of counts (especially the computational inefficiency). Hence, one might want to use approximations which we present in the following.

Example 3.11. *Again, we consider the words v = 'GCCAA' and w = 'CGCGC' in an i.i.d. sequence with equi-probable nucleotide distribution of length $n = 10000$. Figure 3.6 shows the exact count densities for occurrences and clumps. The upper panel contains the densities for the number of occurrences. Both trajectories are very similar except that w achieves a higher variance. In contrast, in the lower panel, the trajectory for the number of clumps for w is shifted towards smaller numbers while the density for v does not change. In fact, the reason is the same as given in Ex. 3.10: The clumps for the non self-overlapping word v always contain exactly one occurrence. Thus, the number of clumps cannot differ from the number of occurrences. However, the self-overlapping word w usually obtains clumps with size > 1. Since the sequence still contains as many occurrences as the non self-overlapping word, the number of clumps has to be smaller for w.*

The shift of the distribution for w is also reflected in the p-values for the number of clumps. The p-value to observe at least 29 hits of v is $2.8 \cdot 10^{-07}$ while the the same number of clumps for w yield a p-value of $7.7 \cdot 10^{-08}$. Since the distribution for w is shifted towards smaller number of clumps, one naturally obtains a better p-value than observing the same number of clumps of v. □

3.3.2 Position Independence

As for the number of word occurrences, a simple approach to compute the distribution for the number of clumps assumes independence between positions. Correspondingly, the number of clumps has a binomial distribution. The probability for an occurrence is $\tilde{\mu}(w)$. However, one can also approximate this probability to avoid computing the set of principal periods by

$$\tilde{\mu}^*(w) := \mu(w)\left[1 - \mu(w)\right]^{\ell-1}.$$

In the same line, we obtain a Poisson approximation $\mathcal{P}(\vartheta)$ with parameter $\vartheta = (n-\ell+1)\tilde{\mu}(w)$ or $\vartheta^* = (n - \ell + 1)\tilde{\mu}^*(w)$ depending on the level of accuracy. For the choice of ϑ, we can compute Chen-Stein error bounds.

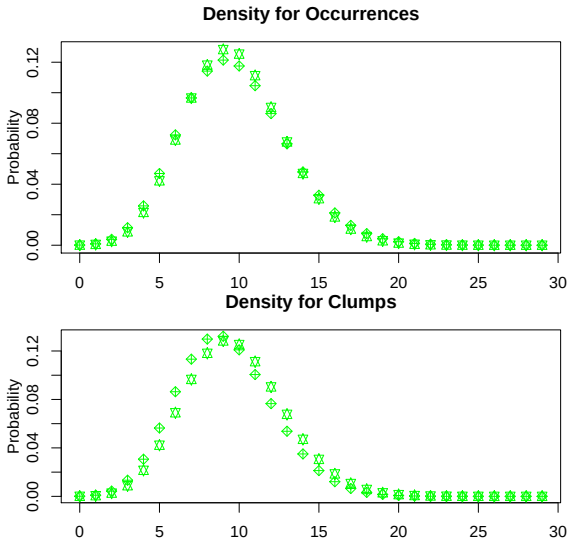

Figure 3.6: Densities for number of occurrences (upper panel) and clumps (lower panel) for the word 'GCCAA' (circles) and 'CGCGC' (crosses) in an i.i.d. sequence with equi-probable nucleotide distribution and length $10,000$.

Chen-Stein Error Bounds Given the approximation of the distribution of the number of clumps by a Poisson distribution $\mathcal{P}(\vartheta)$ with $\vartheta = (n-\ell+1)\tilde{\mu}(w)$, we can compute bounds for the approximation error. Following Reinert *et al.* (2005) but simplifying to the i.i.d. sequence model, we compute the total variation distance

$$d_{\mathrm{TV}}\left(\mathcal{L}(\tilde{N}_n(w)), \mathcal{L}(\mathcal{P}(\vartheta))\right).$$

where the bound components are defined below Eq. (3.3). First of all, we have to define the index set I which contains all random variable indices $\{1,\ldots,n-\ell+1\}$. The neighborhoods with local dependencies are chosen to be the indices with dependent variables. We have seen that the dependencies stretch over $2\ell-2$ positions to the left. For simplicity, we define the neighborhood symmetrically and obtain $B_i := \{i-2\ell+2, i-2\ell+3, \ldots, i+2\ell-2\}$. Again, we ignore boundary effects. Hence, the $\tilde{Y}_i$s are independent if they are not belonging to their neighborhoods. Therefore, $b_3 = 0$ so we can use the improved bound in Eq. (3.4)

$$d_{\mathrm{TV}}\left(\mathcal{L}(\tilde{N}_n(w)), \mathcal{L}(\mathcal{P}(\vartheta))\right) \leq \frac{1-e^{-\vartheta}}{\vartheta}\left(b_1+b_2\right).$$

The first bound b_1 is similar to the word counting bound:

$$b_1 = \sum_{i\in I}\sum_{j\in B_i}\mathbb{E}[\tilde{Y}_i(w)]\mathbb{E}[\tilde{Y}_j(w)] = (n-\ell+1)4(\ell-1)\tilde{\mu}(w)^2 \leq (n-\ell+1)4(\ell-1)\mu(w)^2.$$

Note that this bound is similar to the result in Reinert and Schbath (1999) although derived differently. For the second bound b_2, we obtain slightly better bounds since we only consider one word. Keeping in mind that the joint probability for overlapping clumps is 0, yields

$$b_2 = \sum_{i \in I} \sum_{j \in B_i \setminus \{i\}} \mathbb{E}[\tilde{Y}_i(w)\tilde{Y}_j(w)] = 2(n-\ell+1) \sum_{d=\ell}^{2\ell-2} \tilde{\mu}(w)[1-\omega_d(w)]\mu(w).$$

Analzying the asymptotics, we again assume the word occurs rarely. Thus, $\log n = O(\ell)$ and $\mu(w) = O(n^{-1})$, thus, b_1 is bounded by $O(n^{-1}\log n)$ similar to counting word occurrences. However, b_2 which could not be bounded efficiently for self-overlapping words improves for clumps. Since $1-\omega(w)$ and $1-\omega_d(w)$ are always between 0 and 1 and the sum involves $O(\ell)$ terms, we obtain the same bound as for b_1: $b_2 = O(n^{-1}\log n)$. Hence, for $n \to \infty$, the approximation error vanishes also for self-overlapping words. Therefore, one might choose a Poisson approximation for the clump counts instead of approximating the word occurrences for self-overlapping words.

Example 3.12. *Figure 3.7 compares the exact count distribution with the compound Poisson approximation for the words v = 'GCCAA' and w = 'CGCGC' in an i.i.d. sequence with equi-probable nucleotide distribution of length $n = 10000$. First of all, in all panels, the binomial and the Poisson approximations are very similar. Thus, we combine both for the discussion. The upper left panel shows that the exact distribution is well approximated by the binomial/Poisson distribution. However, the approaches considering the self-overlap (lower panel) slightly improves the approximation. For the self-overlapping word 'CGCGC' (right panels), the differences are significantly higher: The naive approximations (upper right panel) over-estimate the number of clumps. This is not surprising since the naive approach does not consider the overlap. Hence, it cannot adjust for larger clumps leading to a smaller number of clumps to conserve the expected value. In contrast, the approach considering self-overlap (lower right panel) yield accurate approximations for the number of clumps.*

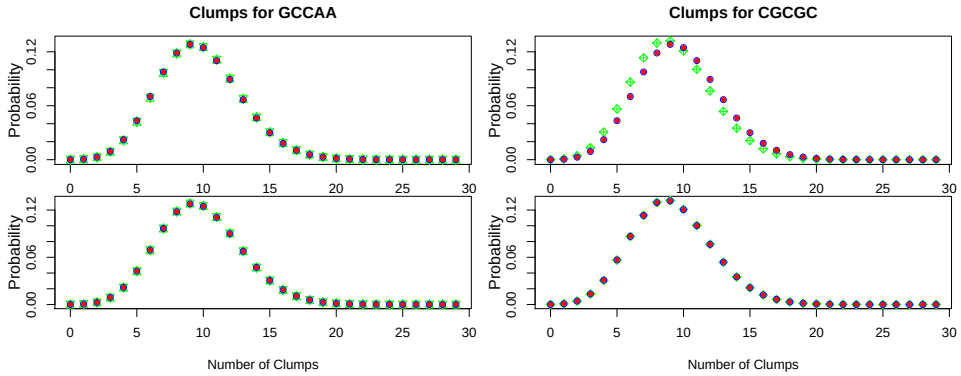

Figure 3.7: Densities for the number of clumps. Upper panel contains the naive binomial (blue circles) and Poisson (red circles) approximations. The lower panel shows the binomial (blue circles) and Poisson (red circles) approximations considering the self-overlap. The green symbol indicates the exact distribution. The left panel contains the word 'GCCAA' and the right panel the word 'CGCGC' in an i.i.d. sequence with equi-probable nucleotide distribution and length $10,000$.

	GCCAA	CGCGC
exact	$2.8 \cdot 10^{-07}$	$7.7 \cdot 10^{-08}$
binomial*	$3.0 \cdot 10^{-07}$	$3.0 \cdot 10^{-07}$
Poisson*	$3 \cdot 10^{-07}$	$3 \cdot 10^{-07}$
binomial	$3.2 \cdot 10^{-07}$	$9 \cdot 10^{-08}$
Poisson	$3.2 \cdot 10^{-07}$	$9.2 \cdot 10^{-08}$

Table 3.1: p-values to observe at least 29 hits for naive (indicated by a star) and standard binomial/Poisson approximations.

The p-values to observe at least 29 *hits are given in Table 3.1. The stars indicate the approximations based on $\tilde{\mu}^*$ for the binomial and ϑ^* for the Poisson distribution. The p-values for the non-overlapping word 'GCCAA' are very accurate. Although the estimates of the naive approximations are somewhat better, the differences are negligibly small. For the word 'CGCGC' only the approximations where the self-overlap is taken into account achieve good results. The naive estimates are almost one magnitude of order too high.* □

3.3.3 Compound Poisson Approximation

Occurrences of self-overlapping words tend to aggregate in clumps. This explains why the Poisson approximation for the number of occurrences does not converge to the true distribution (see Chen-Stein error bounds in Section 3.2.2). One can solve the problem by explicitly modelling the clumps using a compound Poisson distribution.

The compound Poisson distribution is the distribution of a random sum $\sum_{i=1}^{V} U_i$ where V is a Poisson random variable (for a detailled discussion of the compound Poisson distribution called Poisson-stopped-sum distributions, see Johnson *et al.*, 1995, Chapter 9). Furthermore, one assumes U_i to be independently and identically distributed. As we have seen before this section, the number of clumps $\tilde{N}_n(w)$ can be approximated by a Poisson distribution. Denoting the clump size of clump i by Z_i, we obtain for the number of counts

$$N_n(w) = \sum_{i=1}^{\tilde{N}_n(w)} Z_i.$$

Thus, we only have to find the distribution of the clump size. A clump of size 1 occurs if no successive occurrence overlaps with the first occurrence of the clump. The probability for this event is $1 - \omega(w)$. Hence, observing a clump of size 2 means that we first observe an overlapping occurrence (probability $\omega(w)$), which has no successive overlapping occurrence (probability $1 - \omega(w)$). Thus, we obtain a geometric distribution for the clump size of w denoted by $Z(w)$:

$$\mathbb{P}_\mu(Z(w) = k) = [1 - \omega(w)]\, \omega(w)^{k-1}.$$

Since $Z_i \sim Z$, we can now compute the compound Poisson distribution for the number of occurrences $N(w)$. In fact, the compound Poisson distribution consisting of a Poisson random sum of geometrically distributed random variables is called Pólya-Aeppli distribution (Johnson *et al.*, 1995, Chapter 9.7) first described by Pólya (1930). The probability to observe no occurrence means that no clump is observed. The corresponding probability is given by the Poisson probability $\mathbb{P}_\mu(\tilde{N}_n(w) = 0) = e^{-\vartheta}$ with $\vartheta = \mathbb{E}[\tilde{N}_n(w)] = (n-\ell+1)\tilde{\mu}(w)$. The probability for x occurrences is slightly more complicated since the number of clumps $\tilde{x}$ is not given. However, x occurrences can be in 1 to x clumps. Depending on the number of clumps, the number of different clump sizes differ. For example, if x occurrences are contained in either 1 or x clumps, each clump has the same size. However, partitioning the x occurrences into $\tilde{x}$ clumps, there are $\binom{x-1}{\tilde{x}-1}$ different possibilities: Consider occurrences denoted by X and a partitioning (bounds at $\wedge$) into clumps like

$$\left| \begin{array}{ccccccccccc} X & & X & & X & \dots & X & & X & & X \\ & & & \wedge & & & & & & \wedge & \end{array} \right|$$

Labeling the occurrences by 1 to x, each bound corresponds to one of these numbers (if we define that the bound occurs to the right of the corresponding occurrence). In the above sketch, the first bound has number 2. Now, we can think of an urn containing all numbers from 1 to $x-1$ and we draw $\tilde{x}-1$ bounds. We have to subtract 1 to ensure that the last clump contains at least one hit. Therefore, we only draw $\tilde{x}-1$ bounds such that the last clump contains occurrences from this bound till the last occurrence. The $x-1$ balls in the urn ensure that there is at least one further occurrence after the last bound. From combinatorics, it is clear that these are $\binom{x-1}{\tilde{x}-1}$ possibilities. Combining the geometric distribution incorporating the different possibilities with the Poisson distribution yields for $x > 0$

$$\mathbb{P}_\mu(N_n(w) = x) = \sum_{\tilde{x}=1}^{x} \binom{x-1}{\tilde{x}-1} \frac{\vartheta^{\tilde{x}} e^{-\vartheta}}{\tilde{x}!} \left[1-\omega(w)\right]^{\tilde{x}} \omega(w)^{x-\tilde{x}}. \tag{3.26}$$

The exponents for $\omega(w)$ respectively $1-\omega(w)$ reflect the number of required overlaps resp. non-overlaps for all $\tilde{x}$ clumps. Using $\omega(w)^{x-\tilde{x}} = \omega(w)^x \omega(w)^{-\tilde{x}}$ and including the definition for $x = 0$ yields as final expression for the Pólya-Aeppli distribution (for a different derivation, see Johnson *et al.*, 1995, Chapter 9.7)

$$\mathbb{P}_\mu(N_n(w) = x) = \begin{cases} e^{-\vartheta} & : x = 0 \\ e^{-\vartheta}\omega^x \sum_{\tilde{x}=1}^{x} \binom{x-1}{\tilde{x}-1} \frac{\left[\vartheta(1-\omega)\omega^{-1}\right]^{\tilde{x}}}{\tilde{x}!} & : x > 0 \end{cases}.$$

Note that this formula is only valid for words allowing self-overlap ($\omega(w) > 0$). Otherwise, one might use Eq. (3.26) or equivalently the Poisson approximation in Section 3.3.2.

Example 3.13. *Figure 3.8 compares the exact count distribution with the compound Poisson approximation for the words v = 'GCCAA' and w = 'CGCGC' in an i.i.d. sequence with equi-probable nucleotide distribution of length $n = 10000$. The upper panel contains the densities for the number of occurrences, the cumulated distributions are shown in the lower panel. In each panel, there is almost no difference between both trajectories. Thus, the*

compound Poisson approximation considering self-overlaps by explicit modelling of clump (sizes) is a fairly good approximation in both cases. The number of clumps are not shown since they are already discussed in Ex. 3.12.

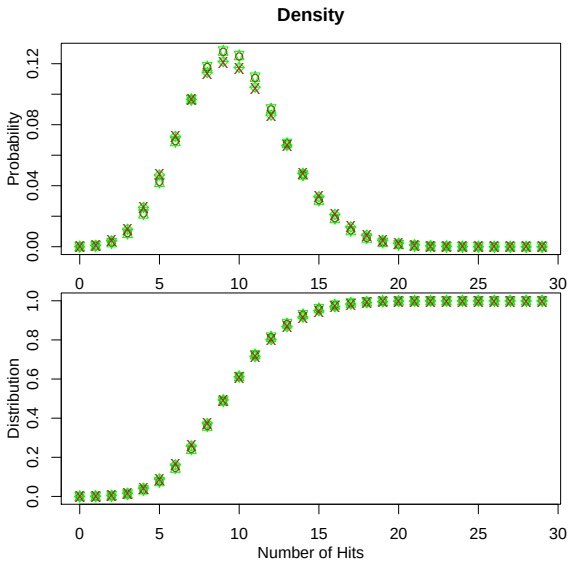

Figure 3.8: Densities (upper panel) and cumulated dsitributions (lower panel) for the number of occurrences for the words 'GCCAA' (circles) and 'CGCGC' (crosses) in an i.i.d. sequence with equi-probable nucleotide distribution and length $10,000$. Green indicates the exact calculation while brown labels the compound Poisson approximation.

The p-value to observe at least 29 *hits of* v *is* $3.2 \cdot 10^{-07}$ *in comparison to the exact p-value of* $2.8 \cdot 10^{-07}$. *Also the approximation for the self-overlapping word is very accurate: The compound Poisson distribution approximated the exact p-value of* $1.9 \cdot 10^{-06}$ *by* $2.1 \cdot 10^{-06}$. *Thus, the compound Poisson distribution is a good choice for a fast and accurate approximation of the number of occurrences.* □

Chen-Stein Error Bounds One can also compute Chen-Stein error bounds for the approximation (Reinert and Schbath, 1998, 1999). First of all, one can show that the compound Poisson distribution $\sum_{i=1}^{\tilde{N}_n(w)} Z_i(w)$, which sums up the clump sizes $Z_i(w)$, is similar to a compound Poisson distribution, which counts the number of clumps of size k (see Section 3.5.1 for a similar approach). Let $\tilde{Z}_k(w)$ denote the number of clumps of size k for word w. Then, we obtain the approximated compound Poisson distribution $\sum_{k\geq 1} k\tilde{Z}_k(w)$ where $\tilde{Z}_k(w)$ are independent Poisson random variables. The parameter of these processes are the expected value of the number of k-clumps. The probability $\tilde{\mu}_k(w)$ to observe a clump of size k is computed by the fact that the clump starts with probability $[1-\omega(w)]\mu(w)$, has $k-1$ overlapping occurrences with probability $\omega(w)^{k-1}$ and no subsequent overlap with probability $1-\omega(w)$. Hence, we obtain

$$\tilde{\mu}_k(w) := [1-\omega(w)]^2 \mu(w)\omega(w)^{k-1}. \tag{3.27}$$

Then, the expected value and, thus, the parameter of the independent Poisson process $\tilde{Z}_k(w)$ is

$$\vartheta_k := \mathbb{E}[\tilde{Z}_k(w)] = (n - \ell + 1)\tilde{\mu}_k(w). \tag{3.28}$$

With $\vartheta := \tilde{\mu}(w)$, we can compute the probability for number of occurrences by (Kemp, 1967)

$$\begin{aligned} \mathbb{P}_\mu(N_n(w) = 0) &= e^{-\vartheta}, \\ \mathbb{P}_\mu(N_n(w) = x + 1) &= \frac{\vartheta}{x+1} \sum_{x'=0}^{x} (x + 1 - x') \frac{\vartheta_{x+1-x'}}{\vartheta} \mathbb{P}_\mu(N_n(w) = x'). \end{aligned}$$

The calculation of the error bounds is complicated by many technical difficulties, thus, we refer the reader to Reinert and Schbath (1999) and only report the main results. The neighborhoods for the Chen-Stein bounds are defined such that all dependent random variables are included. The bound b_1 is easy to compute. However, the second moments for b_2 involves iterations over all possible concatenated (overlapping) words for each k-clump. The probabilities are bounded by considering the set of words which precede and succeed a k-clump.

The total variation distance between the number of occurrences $N_n(w)$ and the compound Poisson distribution $\mathcal{CP}$ is

$$\begin{aligned} d_{\mathrm{TV}}\left(\mathcal{L}(N_n(w)), \mathcal{CP}\right) \leq\ & (n - \ell + 1)\left[2(\ell - 1)\mu(w)\tilde{\mu}(w) + (4\ell - 3)\tilde{\mu}(w)^2)\right] \\ & + 4(n - \ell + 1)\mu(w)^2(\ell - 1) + (\ell - 1)(\mu(w) - \tilde{\mu}(w)). \end{aligned}$$

The error for the number $\tilde{N}_n(w)$ of clumps approximated by a Poisson distribution $\mathcal{P}(\vartheta)$ with $\vartheta = (n - \ell + 1)\tilde{\mu}(w)$ is bounded by

$$\begin{aligned} d_{\mathrm{TV}}\left(\mathcal{L}(\tilde{N}_n(w)), \mathcal{P}(\vartheta)\right) \leq\ & (n - \ell + 1)\left[2(\ell - 1)\mu(w)\tilde{\mu}(w) + (4\ell - 3)\tilde{\mu}(w)^2)\right] \\ & + 4(n - \ell + 1)\mu(w)^2(\ell - 1). \end{aligned}$$

Asymptotically, we again use the rare word assumption leading to $\log n = O(\ell)$ and $\mu(w) = O(n^{-1})$. The first line and the first term of the second line yield $O(n \log n \cdot n^{-2}) = O(n^{-1} \log n)$ converging to 0 for large n. The additional term for the compound Poisson approximation involves $\ell(\mu(w) - \tilde{\mu}(w))$. Since $0 < \tilde{\mu}(w) \leq \mu(w)$, it can be bound by $O(n^{-1} \log n)$. Hence, both bounds are $O(n^{-1} \log n)$. Therefore, the approximation error converges to 0 for large n.

3.3.4 Normal Approximation

The normal approximation needs the expected value and the variance as parameters. Based on the exact formulae (see Section 3.3.1), it is straight-forward to derive the asymptotic distribution. To our knowledge, a normal approximation for the number of clumps has not yet been published. As for the number of occurrences, we use the asymptotic expected value and variance. The asymptotic expected value is

$$\lim_{n\to\infty} n^{-1}\mathbb{E}[\tilde{N}_n(w)] = \tilde{\mu}(w).$$

The asymptotic covariance can be derived from Eq. (3.25). In fact, the third line of Eq. (3.25) vanishes since it is $O(\ell)$. Hence, we obtain

$$\tilde{\sigma}^2 := \lim_{n\to\infty} n^{-1}\mathbb{V}[\tilde{N}_n(w)] = \tilde{\mu}(w)\left(1-\tilde{\mu}(w)+2\left(\sum_{d=\ell}^{2\ell-1}\varpi_d(w)\right)-2(\ell-1)\tilde{\mu}(w)\right). \quad (3.29)$$

The number of counts $n^{-1}\tilde{N}_n(w)$ is asymptotically normal distributed with mean $\tilde{\mu}(w)$ and variance $n^{-1}\tilde{\sigma}^2$. For $\tilde{\sigma} \neq 0$, we obtain the asymptotic distribution

$$\frac{\sqrt{n}\left(n^{-1}\tilde{N}_n(w)-\tilde{\mu}(w)\right)}{\tilde{\sigma}} \sim \mathcal{N}(0,1)$$

for long sequences and the assumption that the word occurs frequently.

Example 3.14. *Figure 3.9 compares the exact count distribution for clumps with its limiting normal approximation for the words v = 'GCCAA' and w = 'CGCGC' in an i.i.d. sequence with equi-probable nucleotide distribution of length $n = 10000$. The upper panel contains the densities for the number of occurrences. The limiting normal distribution does not fit well to the exact distribution - neither for the word v nor for w. In Ex. 3.5, we have seen that the normal approximation for the number of occurrences incorporates the self-overlap but is nevertheless a weak approximation. The reason is that the words only occur rarely. We can draw the same conclusion for the number of clumps since both approximations - for v and w - are equally weak.*

The p-value to observe at least 29 hits of v respectively w is $6.4e-10$ respectively $5e-11$ in comparison to the exact p-values of $2.8e-07$ and $7.7e-08$. Obviously, the limiting normal approximation cannot be used for p-value approximation for rare words. □

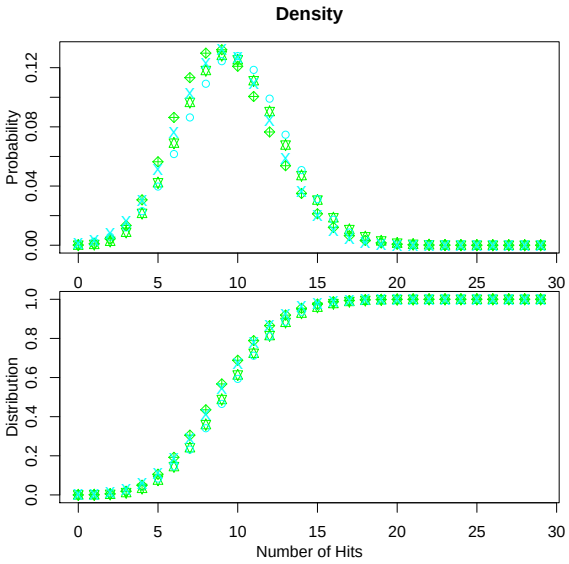

Figure 3.9: Densities (upper panel) and distribution (lower panel) of the number of clumps for the word 'GCCAA' (circles) and 'CGCGC' (crosses) in an i.i.d. sequence with equi-probable nucleotide distribution and length $10,000$. Green indicates the exact calculation while light blue labels the limiting normal approximation.

3.4 Multiple Words

DNA motifs are usually not described by only one word but a set $\mathcal{W} = \{w^{(1)}, \ldots, w^{(q)}\}$ of q different words. Any occurrence of $w \in \mathcal{W}$ is considered to be an occurrence of the DNA motif. Thus, the number of occurrences in a sequence of length n is given by

$$N_n(\mathcal{W}) = \sum_{w \in \mathcal{W}} N_n(w).$$

Of course, the count random variables are not independent. Thus, we have to derive new formulae to compute the count distribution. For simplicity, we assume the same length for all words $|w| = \ell \quad \forall_{w \in \mathcal{W}}$. For DNA motifs modeled by PFMs this is always the case. Furthermore, extension to different lengths is possible but heavily complicates notation and formulae.

First, we derive formulae for the exact expected value and the variance. Calculation of the corresponding count distribution is computationally very demanding. The main problem is introduced by possible overlaps. The occurrence probability depends on a preceding overlapping hit. For set of words, one has to consider all possible predecessors. This makes the classical recursive formula (Robin *et al.*, 2005) for the exact distribution high-dimensional (quadratic in the number of words). However, we present the conditional approach (Zhang *et al.*, 2007) since its extension to multiple words is straight-forward. The distribution can also be derived from generating functions, which are presented - although only for single words and clumps - in Chapter 4, or can be computed based on language decomposition

approaches (Régnier, 2000; Régnier and Denise, 2004) and probability arithmetic automata (Marschall and Rahmann, 2008). Subsequently, we present the independence model based on the binomial/Poisson distribution. We close with the asymptotic normal approximation (Bender and Kochman, 1993; Waterman, 2000).

3.4.1 Exact Count Distribution

First and Second Moments

The expected value of the sum of random count variables is straight-forward since dependencies do not matter. The expected value is the sum of the individual expected values, yielding

$$\mathbb{E}[N_n(\mathcal{W})] = (n - \ell + 1) \sum_{w \in \mathcal{W}} \mu(w).$$

As always, calculation of the variance is more complicated due to the dependencies. First, we decompose the variance into the co-/variances of the words:

$$\mathbb{V}[N_n(\mathcal{W})] = \sum_{w \in \mathcal{W}} \sum_{v \in \mathcal{W}} \mathbb{C}\mathrm{ov}[N_n(w), N_n(v)]. \tag{3.30}$$

The variances $\mathbb{C}\mathrm{ov}[N_n(w), N_n(w)]$ are already known from Eq. (3.8). The covariances between different words are computed similarily: First, we extend the definition of the overlap bit for two words for $0 \leq d < \ell$ to

$$\epsilon_d(w, v) := \begin{cases} 1 & \text{if } \forall_{0<\kappa\leq\ell-d} \quad w_{\kappa+d} = v_\kappa \\ 0 & \text{otherwise} \end{cases} \tag{3.31}$$

Again, we set $\epsilon_d(w, v) = 1$ for $d \geq \ell$. As we show in the subsequent example, the overlap bit is not necessarily symmetric. An exception is, of course, the self overlap $\epsilon_d(w) = \epsilon_d(w, w)$.

Example 3.15. *The overlap bit vector $\epsilon_d(w, v)$ for the word w = 'ACA' and v = 'CAA' is given by*

d	A	C	A			$\epsilon_d(w, v)$
0	C	A	A			0
1		C	A	A		1
2			C	A	A	0

In contrast, the overlap bit $\epsilon_d(v, w)$ is

d	C	A	A			$\epsilon_d(v, w)$
0	A	C	A			0
1		A	C	A		0
2			A	C	A	1

□

Based on the overlap bit vector, we can now define the overlap probability for w and v by

$$\gamma_d(w,v) := \mathbb{P}_\mu(Y_i(w)=1, Y_{i+d}(v)=1) = \mu(w)\epsilon_d(w,v)\prod_{\kappa=\ell-d+1}^{\ell}\mu(v_\kappa). \tag{3.32}$$

Also $\gamma_d(w,v)$ are not necessarily symmetric.

Now, we can start decomposing the covariance into its indicator random variables. Since the words are assumed to be different but of equal length, we cannot have an occurrence of w and v at the same position. Otherwise, w could be a prefix or suffix of v. Furthermore, non-overlapping occurrences are independent. Thus, we obtain for $w \neq v$

$$\begin{aligned}\mathbb{C}\mathrm{ov}[N_n(w),N_n(v)] &= \sum_{i=1}^{n-\ell+1}\sum_{j=1}^{n-\ell+1}\mathbb{C}\mathrm{ov}[Y_i(w),Y_j(v)] \\ &= -(n-\ell+1)\mu(w)\mu(v) \\ &\quad +\sum_{d=1}^{\ell-1}(n-\ell-d+1)[\gamma_d(w,v)+\gamma_d(v,w)-2\mu(w)\mu(v)].\end{aligned} \tag{3.33}$$

Conditional Approach

The conditional approach (Zhang *et al.*, 2007) for single words presented in Section 3.2.1 can be extended to deal with multiple words. The main difference is that one has to consider all possible prefixes of the words in $\mathcal{W}$. Only some of the definitions change slightly. For the probability of at least k occurrences of $\mathcal{W}$ in a sequence of length $n-i+1$, we obtain

$$F_i^{(k)}(\mathcal{W}) := \mathbb{P}_\mu(N_{n-i+1}(\mathcal{W}) \geq k) = \sum_{a\in\mathfrak{A}} f_{i,a}^{(k)}(\mathcal{W})\mu(a),$$

with $f_{i,v}^{(k)}(\mathcal{W}) = \mathbb{P}_\mu(N_{n-i+1}(\mathcal{W}) \geq k | X_1 \ldots X_{|v|} = v)$ for $v \in \mathfrak{A}^+$. For the main recursion, one obtains

$$f_{i,v}^{(k)}(\mathcal{W}) = \begin{cases} \sum_{a\in\mathfrak{A}} f_{i,va}^{(k-1)}(\mathcal{W})\mu(a) & \text{if } va \in \mathcal{W} \\ \sum_{a\in\mathfrak{A}} f_{i,va}^{(k)}(\mathcal{W})\mu(a) & \text{otherwise} \end{cases}.$$

The remaining formulae have to be changed accordingly.

3.4.2 Position Independence

The calculation of the dependencies between the random indicators can be circumvented by assuming position independence. In this case, we retrieve a fairly easy formula for the count distribution. Since the words in $\mathcal{W}$ are assumed to be different, we can construct a random indicator for set of words by

$$Y_i(\mathcal{W}) := \sum_{w \in \mathcal{W}} Y_i(w).$$

Due to the assumed independencies between the occurrences, we can directly compute the probability for $Y_i(\mathcal{W})$:

$$\mu(\mathcal{W}) := \mathbb{P}_\mu(\mathcal{W}) = \sum_{w \in \mathcal{W}} \mu(w).$$

Then, we can use the binomial distribution and obtain

$$\mathbb{P}_\mu(N_n(\mathcal{W}) = m) = \binom{n-\ell+1}{m} \mu(\mathcal{W})^m (1-\mu(\mathcal{W}))^{n-\ell-m+1}.$$

Furthermore, we can apply a Poisson approximation with parameter $\vartheta = (n-\ell+1)\mu(\mathcal{W})$. Although it is already clear that the approximation cannot work for self-overlapping words (see Section 3.2.2), we derive the bounds since they cannot be found in the literature.

Chen-Stein Error Bound The Chen-Stein error bounds can be derived using a similar neighborhood as for the single word case. Instead of using a single index, we define a tuple (i, w) containing the position i and the word $w \in \mathcal{W}$. The index set is defined by $I := \{1, \dots, n-\ell+1\} \times \mathcal{W}$. The neighborhood for (i, w) contains the tuples with indices for overlapping occurrences for all words $B_{(i,w)} := \{i-\ell+1, i-\ell+2, \dots, i+\ell-1\} \times \mathcal{W} \subset I$. We bound the total variation distance by

$$d_{\mathrm{TV}}\left(\mathcal{L}(N_n(\mathcal{W}), \mathcal{P}(\vartheta)\right) \leq \frac{1-e^{-\vartheta}}{\vartheta}(b_1 + b_2).$$

The bound b_1 contains the product of the first moments for all the neighborhoods

$$\begin{aligned} b_1 &= \sum_{(i,w)\in I} \sum_{(j,v)\in B_{(i,w)}} \mathbb{E}[Y_i(w)]\mathbb{E}[Y_j(v)] = \sum_{i=1}^{n-\ell+1} \sum_{w\in\mathcal{W}} \sum_{d=-\ell+1}^{\ell-1} \sum_{v\in\mathcal{W}} \mu(w)\mu(v) \\ &= \sum_{i=1}^{n-\ell+1} \sum_{w\in\mathcal{W}} (2\ell-1)\mu(w)\mu(\mathcal{W}) = (n-\ell+1)(2\ell-1)\mu(\mathcal{W})^2. \end{aligned}$$

We sum over all tuples and for each tuple we consider the neighborhood comprising the overlapping positions to the left and to the right for all words. Using $\mu(\mathcal{W}) = \sum_{w\in\mathcal{W}} \mu(w)$

and counting the sum iterations yields above result. The second bound b_2 involves second moments which become 0 if $i = j$ and $w \neq v$ leading to

$$\begin{aligned} b_2 &= \sum_{(i,w)\in I} \sum_{(j,v)\in B_{(i,w)}\setminus\{(i,w)\}} \mathbb{E}[Y_i(w)Y_j(v)] \\ &= \sum_{i=1}^{n-\ell+1} \sum_{w\in\mathcal{W}} \sum_{v\in\mathcal{W}} \sum_{d=1}^{\ell-1} (\gamma_d(w,v) + \gamma_d(v,w)) \end{aligned}$$

where $\gamma_d(w, v)$ denotes the probability for the joint occurrence of w followed by v in a distance of d.

Under the rare word assumption ($\log n = O(\ell)$ and $\mu(\mathcal{W}) = O(n^{-1})$), we can bound b_1 and b_2. For b_1, we obtain $b_1 = O(n^{-1} \log n)$. Considering b_2, we have to bound the probability for joint occurrences $\gamma_d(w, v)$. Without any assumptions about the (self-)overlapping structure of the words, one obtains $\gamma_d(w, v) = O(n^{-1})$. Thus, we obtain $b_2 = O(\ell)$. Hence, the approximation error does not necessarily converge to zero. However, if all the words are neither self-overlapping nor overlap with any other word in $\mathcal{W}$, the joint probabilities become 0. In this case, the error of the approximation vanishes.

3.4.3 Normal Approximation

The asymptotic normal distribution can also be computed for multiple words (Bender and Kochman, 1993). The first derivation of the multi-dimensional normal approximation was derived based on the δ-method (Lundstrom, 1990) which is described in Waterman (2000). Subsequently, explicit formulae for the asymptotic covariance matrix were published (Prum *et al.*, 1995; Schbath *et al.*, 1995) as well as formulae for the general Markov case (Reinert and Schbath, 1998).

In Section 3.4.1, the formulae for the exact mean and variance are derived. Based on them, we compute the asymptotic mean and variance. Again, the asymptotic mean is easy to compute:

$$\lim_{n\to\infty} n^{-1}\mathbb{E}[N_n(\mathcal{W})] = \lim_{n\to\infty} n^{-1} \sum_{i=1}^{n-\ell+1} \sum_{w\in\mathcal{W}} \mathbb{E}[Y_i(w)] = \mu(\mathcal{W}).$$

Thus, the asymptotic mean of the sum of the occurrences of a set of words $\mathcal{W}$ is the sum of the asymptotic means of the words. Furthermore, the asymptotic mean of the words is the probability of an occurrence at one position.

The variance can be decomposed into the pair-wise co-variances similar to Eq. (3.30)

$$\lim_{n\to\infty} n^{-1}\mathbb{V}(N_n(\mathcal{W})) = \lim_{n\to\infty} n^{-1} \sum_{w\in\mathcal{W}} \sum_{v\in\mathcal{W}} \mathbb{C}\mathrm{ov}[N_n(w), N_n(v)].$$

Hence, we have to compute the asymptotic covariance based on Eq. (3.33). However, we do not assume $v \neq w$ here, thus, we add the second moment $\mathbb{E}[Y_i(w)Y_i(v)] = \mu(w)$ if $w = v$ and otherwise 0 by multiplying with $\epsilon_0(w, v)$ which is 1 if $w = v$ and otherwise 0. We obtain

$$\begin{aligned}
\sigma^2_{v,w} &:= \lim_{n\to\infty} n^{-1}\,\mathbb{C}\mathrm{ov}[N_n(w), N_n(v)] \\
&= \epsilon_0(v,w)\mu(w) - \mu(w)\mu(v) + \sum_{d=1}^{\ell-1} [\gamma_d(w,v) + \gamma_d(v,w) - 2\mu(w)\mu(v)] \\
&= \mu(w)\mu(v) - \epsilon_0(v,w)\mu(w) + \sum_{d=0}^{\ell-1} [\gamma_d(w,v) + \gamma_d(v,w) - 2\mu(w)\mu(v)] \\
&= G(w,v) + G(v,w) - 2\ell\mu(w)\mu(v) - \mu(w)\left[\epsilon_0(w,v) - \mu(v)\right].
\end{aligned}$$

First, we incorporate the first terms into the sum by changing the index from $d = 1$ to $d = 0$. However, the sum contains γ_d twice, thus, we have to subtract $\mu(w)$ if $w = v$. Furthermore, the sum also contains $-\mu(w)\mu(v)$ twice, hence, we add this, again. Finally, we substitute the sum over γ_d by $G(w, v) = \sum_{d=0}^{\ell-1} \gamma_d(w, v)$.

In fact, the number of counts of words $\mathbf{N}_n := (N_n(w^{(1)}), N_n(w^{(2)}), \ldots, N_n(w^{(q)}))$ is a multi-dimensional normal distribution. Let the asymptotic expected values be

$$\boldsymbol{\mu} := (\mathbb{E}[Y_i(w^{(1)})], \mathbb{E}[Y_i(w^{(2)})], \ldots, \mathbb{E}[Y_i(w^{(q)})]) \tag{3.34}$$

and the asymptotic covariance matrix denoted by

$$\Sigma := (\sigma^2_{v,w})_{v,w\in\mathcal{W}}. \tag{3.35}$$

Then, the count vector $n^{-1}\mathbf{N}_n$ is asymptotically normal with mean $\boldsymbol{\mu}$ and covariance matrix $n^{-1}\Sigma$. Furthermore, if $\det(\Sigma) \neq 0$, we obtain the multi-dimensional standard normal distribution by

$$\sqrt{n}\Sigma^{-1/2}\left(n^{-1}\mathbf{N}_n - \boldsymbol{\mu}\right) \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$$

were $\mathbf{0}$ is a q-dimensional vector filled with 0s and $\mathbf{1}$ is a $q \times q$ matrix with diagonal entries 1 and otherwise 0. This approximation assumes long sequences and that the word occurs frequently.

Finally, the summed number of occurrences is retrieved by multiplying a q-dimensional column vector $t = (1, 1, \ldots, 1)^T$ from the right to the count vector $\mathbf{N}_n$. This yields $\mathbf{N}_n t \sim \mathcal{N}(\boldsymbol{\mu}t, n^{-1}t^T\Sigma t)$ leading to $n^{-1}N_n(\mathcal{W}) \sim \mathcal{N}(\mu(\mathcal{W}), n^{-1}\sigma^2)$ with $\sigma^2 := \sum_{w\in\mathcal{W}}\sum_{v\in\mathcal{W}}\sigma^2_{w,v}$.

3.5 Clumps for Sets of Words

For a set of words, we can differentiate between two different types of clumps. Homogeneous clumps are overlapping occurrences of the same word. Two homogeneous clumps of the same word cannot overlap. In contrast, two homogeneous clumps of two different words might overlap. The second type of clump is called heterogeneous since it can contain any combination (including multiplicities) of words of the set. By definition, heterogeneous clumps cannot overlap.

Example 3.16. *The set of words is given by* $\mathcal{W} = \{$*'GCCAA'*, *'CGCGC'*$\}$*. The sequence 'ACGCGCGCCAAT' contains one homogeneous clump of the word 'CGCGC' of size* 2 *starting at position* 2*. Furthermore, there is one homogeneous clump for 'GCCAA' of size* 1 *starting at position* 7*. However, if we consider heterogeneous clumps, the whole sequence only contains one clump of size* 3 *starting at position* 2 *and ending at position* 11*.* □

In Section 3.3 about clumps for single words, we can only consider homogeneous clumps. There, the random indicator for a clump of a single word w is defined by $\tilde{Y}_i(w) := Y_i(w)\prod_{d=1}^{\ell-1} Y_{i-d}(w)$. This definition can be adopted for homogeneous clumps in the framework of multiple words. However, definition of heterogeneous clumps of a set of words $\mathcal{W}$ is slightly more complicated. Under the assumption that words have equal lengths, we obtain

$$\breve{Y}_i(\mathcal{W}) := \left(\sum_{w\in\mathcal{W}} Y_i(w)\right)\prod_{d=1}^{\ell-1}\left(1-\sum_{w\in\mathcal{W}} Y_{i-d}(w)\right). \tag{3.36}$$

The definition ensures that no predecessor of any word in $\mathcal{W}$ overlaps with the initial occurrence at position i of a word in $\mathcal{W}$ starting the clump. Note that usually $\breve{Y}_i(\mathcal{W}) \neq \sum_{w\in\mathcal{W}} \tilde{Y}_i(w)$. The number of counts of homogeneous clumps is $\tilde{N}_n(\mathcal{W}) := \sum_{w\in\mathcal{W}} \tilde{N}_n(w)$ which is also usually different from the number of heterogeneous clumps given by $\breve{N}_n(\mathcal{W}) := \sum_{i=1}^{n-\ell+1} \breve{Y}_i(\mathcal{W})$.

The first part considers homogeneous clumps. After deriving the first two moments, we present the independence approach with Chen-Stein error bounds. Then, we describe the more elaborate compound Poisson approximation (Reinert and Schbath, 1999). However, this approach is not capable of considering overlaps between clumps of different words. Finally, we derive a normal approximation for homogeneous clumps. The second part considers heterogeneous clumps - consisting of one or more different or equal words. After deriving the formula for the expected value, we present the important compound Poisson approach (Roquain and Schbath, 2007) to compute occurrences and clumps by considering all possible overlaps.

3.5.1 Homogeneous Clumps

Expected Value and Variance

For homogeneous clumps, the calculation of the expected value is the sum of the expected value of the single word case. One obtains

$$\mathbb{E}[\tilde{N}_n(\mathcal{W})] = \sum_{w\in\mathcal{W}} \mathbb{E}[\tilde{N}_n(w)] = (n-\ell+1)\sum_{w\in\mathcal{W}} \tilde{\mu}(w).$$

The variance can be decomposed into the covariances of the random indicators by

$$\begin{aligned}
\mathbb{V}[\tilde{N}_n(\mathcal{W})] &= \sum_{w\in\mathcal{W}}\sum_{v\in\mathcal{W}} \mathbb{C}\mathrm{ov}[\tilde{N}_n(w), \tilde{N}_n(v)] \qquad (3.37)\\
&= \sum_{w\in\mathcal{W}}\sum_{v\in\mathcal{W}}\sum_{i=1}^{n-\ell+1}\sum_{j=1}^{n-\ell+1} \mathbb{C}\mathrm{ov}[\tilde{Y}_i(w), \tilde{Y}_j(v)]\\
&= \sum_{w\in\mathcal{W}}\sum_{v\in\mathcal{W}}\sum_{i=1}^{n-\ell+1} \mathbb{C}\mathrm{ov}[\tilde{Y}_i(w), \tilde{Y}_i(v)]\\
&\quad + \sum_{w\in\mathcal{W}}\sum_{v\in\mathcal{W}}\sum_{d=1}^{n-\ell-i+1} \left[\mathbb{C}\mathrm{ov}[\tilde{Y}_i(w), \tilde{Y}_{i+d}(v)] + \mathbb{C}\mathrm{ov}[\tilde{Y}_i(v), \tilde{Y}_{i+d}(w)]\right].
\end{aligned}$$

From the second to the last line, we change the index j to distance d from position i. Furthermore, we separate the covariances, which have distance 0. These are easy to compute. For $w = v$, we obtain $\tilde{\mu}(w) - \tilde{\mu}(w)^2$ and for $w \neq v$ the covariance yields $-\tilde{\mu}(w)\tilde{\mu}(v)$ since two clumps of different words cannot start at the same position. The remaining covariance terms can be computed by considering the overlap probabilities. For clumps of single words, we introduced the self-overlap probability $\omega_d(w)$ in Eq. (3.23). For computation of the covariance between two different words, we have to extend the definition such that the preceding clump can be based on a different word. Applying the same reasoning, we obtain

$$\begin{aligned}
\omega_d(w|v) &:= \mathbb{P}_\mu(\tilde{Y}_{i+d}(w) = 0 | Y_{i+d}(w) = 1, \tilde{Y}_i(v) = 1)\\
&= \sum_{\eta\in\Upsilon'(w)} \epsilon_{d-\eta}(v, w) \prod_{\kappa=1+\max(\eta-d+\ell,0)}^{\eta} \mu(w_\kappa).
\end{aligned}$$

The only modification is that the $d - \eta$ has to be a valid overlap between v followed by w. The joint probability of a clump of words v at position i and a clump of words w at position $i + d$ is analogous to Eq. (3.24)

$$\mathbb{P}_\mu(\tilde{Y}_{i+d}(w) = 1, \tilde{Y}_i(v) = 1) = (1 - \omega_d(w|v))\mu(w)\tilde{\mu}(v).$$

Note that the joint probability is not symmetric in terms of the order of the clump of w and v. The covariance term is the joint probability minus the product of the single event probabilities. Defining $\varpi_d(w|v) := [1 - \omega_d(w)]\mu(w) - \tilde{\mu}(w)$ yields

$$\mathbb{C}\mathrm{ov}[\tilde{Y}_i(v), \tilde{Y}_{i+d}(w)] = \tilde{\mu}(v)\varpi_d(w|v).$$

Substituting these terms into Eq. (3.37) for the variance finalizes the calculation. Furthermore, we present the formula for pair-wise covariance of the number of clumps which we need for the normal approximation. Starting point is the corresponding Eq. (3.25) for the number of counts. We separate the covariance terms such that it is easier to compute the asymptotic covariance (see Section 3.5.1)

$$\begin{aligned}\mathbb{C}\mathrm{ov}[\tilde{N}_n(w), \tilde{N}_n(v)] &= (n-\ell+1)\left[\epsilon_0(w,v)\tilde{\mu}(w) - \tilde{\mu}(w)\tilde{\mu}(v)\right] \\ &\quad +(n-3\ell+1)\left(\sum_{d=\ell}^{2\ell-1}\tilde{\mu}(w)\varpi_d(v|w) + \tilde{\mu}(v)\varpi_d(w|v)\right) \\ &\quad -2(n-3\ell+1)(\ell-1)\tilde{\mu}(w)\tilde{\mu}(v) \\ &\quad +\left(\sum_{k=\ell}^{2\ell-1}\sum_{d=\ell}^{k}\tilde{\mu}(w)\varpi_d(v|w) + \tilde{\mu}(v)\varpi_d(w|v)\right) - 3\ell(\ell-1)\tilde{\mu}(w)\tilde{\mu}(v).\end{aligned} \tag{3.38}$$

Position Independence

Under the assumption that occurrences are independent of each other, a simple approximation is obtained. With

$$\tilde{\mu}^*(\mathcal{W}) := \sum_{w\in\mathcal{W}} \tilde{\mu}^*(w)$$

one can approximate the expected value of the number of homogeneous clumps. However, the approximation is only reasonable for very weakly (self-) overlapping sets of words $\mathcal{W}$. Besides the binomial approximation with $\tilde{\mu}^*(\mathcal{W})$ as success probability and $n-\ell+1$ trials, we obtain a Poisson approximation with parameter $\vartheta^* = (n-\ell+1)\tilde{\mu}^*(\mathcal{W})$. A more severe Poisson approximation is based on the expected value such that we obtain the parameter $\vartheta = \tilde{\mu}(\mathcal{W})$.

Chen-Stein Error Bounds The Chen-Stein error bounds for a Poisson $\mathcal{P}(\vartheta)$ approxmiation for the number of homogeneous clumps of a set of words $\mathcal{W}$ is given in Reinert and Schbath (1998, 1999). Since the cited approaches also bound the approximation error for a compound Poisson approximation (see next section), they are more complicated than needed. Therefore, we derive the bounds explicitly. As usually, the total variation distance is given by

$$d_{\mathrm{TV}}\left(\mathcal{L}(\tilde{N}_n(\mathcal{W}), \mathcal{P}(\vartheta)\right) \leq \frac{1-e^{-\vartheta}}{\vartheta}(b_1+b_2).$$

The Chen-Stein bounds are derived by combining the bounds for clumps of single words and the multiple word occurrences case. The index set is defined by $I := \{1,\ldots,n-\ell+1\}\times\mathcal{W}$. To capture all dependencies in the neighborhoods, we have to include all $2\ell-2$ positions to the left. Again, we define them symmetrically by $B_{(i,w)} := \{i-2\ell+2, i-2\ell+3,\ldots,i+2\ell-2\}\times\mathcal{W}$. We obtain the bound b_1

$$\begin{aligned} b_1 &= \sum_{(i,w)\in I} \sum_{(j,v)\in B_{(i,w)}} \mathbb{E}[\tilde{Y}_i(w)]\mathbb{E}[\tilde{Y}_j(v)] = \sum_{i=1}^{n-\ell+1} \sum_{w\in\mathcal{W}} \sum_{d=-2\ell+2}^{2\ell-2} \sum_{v\in\mathcal{W}} \tilde{\mu}(w)\tilde{\mu}(v) \\ &= 4(n-\ell+1)(\ell-1)\tilde{\mu}(\mathcal{W})^2. \end{aligned}$$

The second bound is slightly more complicated to compute. On the one hand, two clumps of the same word cannot overlap. However, they can depend on each up to a distance of $d = 2\ell - 2$. On the other hand, two clumps of different words can overlap, thus, we have to consider the overlap probabilities. For distances $d \geq \ell$, they are independent. First, we separate these three cases and obtain

$$\begin{aligned} b_2 &= \sum_{(i,w)\in I} \sum_{(j,v)\in B_{(i,w)}\setminus\{(i,w)\}} \mathbb{E}[\tilde{Y}_i(w)\tilde{Y}_j(v)] = \sum_{i=1}^{n-\ell+1} \sum_{w\in\mathcal{W}} \sum_{v\in\mathcal{W}} \sum_{d=-2\ell+2}^{2\ell-2} \mathbb{E}[\tilde{Y}_i(w)\tilde{Y}_{i+d}(v)] \\ &= b_{21} + b_{22} + b_{23} \end{aligned}$$

with

$$\begin{aligned} b_{21} &:= 2\sum_{i=1}^{n-\ell+1} \sum_{w\in\mathcal{W}} \sum_{d=\ell}^{2\ell-2} \left[\tilde{\mu}(w)\left(1-\omega_d(w)\right)\mu(w)\right], \\ b_{22} &:= \sum_{i=1}^{n-\ell+1} \sum_{w\in\mathcal{W}} \sum_{v\in\mathcal{W}\setminus\{w\}} \sum_{d=1}^{\ell-1} \Big(\mathbb{E}[\tilde{Y}_i(w)\tilde{Y}_{i+d}(v)] + \mathbb{E}[\tilde{Y}_i(v)\tilde{Y}_{i+d}(w)]\Big), \\ b_{23} &:= 2\sum_{i=1}^{n-\ell+1} \sum_{w\in\mathcal{W}} \sum_{v\in\mathcal{W}\setminus\{w\}} \sum_{d=\ell}^{2\ell-2} \tilde{\mu}(w)\tilde{\mu}(v). \end{aligned}$$

Now, we can simplify the expressions. For b_{21}, we can use $\tilde{\mu}(w) \leq \mu(w)$ and $0 \leq \omega_d(w) < 1$ to obtain

$$b_{21} \leq 2(n-\ell+1)(\ell-2)\mu(\mathcal{W})^2.$$

The bound b_{22} still contains the second moments. Since we can bound the probability for a clump by the probability for an occurrence, we obtain the inequality $\mathbb{E}[\tilde{Y}_i(w)\tilde{Y}_{i+d}(v)] \leq \gamma_d(w,v) \leq \mu(w)$. Hence, b_{22} can be bounded by

$$b_{22} \leq 4(n-\ell+1)(\ell-1)(q-1)\mu(\mathcal{W}).$$

Finally, b_{23} is easy to calculate and yields $b_{23} \leq 2(n-\ell+1)(\ell-2)\mu^2(\mathcal{W})$.

Under the rare word assumption, the asymptotics for b_1 are similar to the multiple word case $b_1 = O(n^{-1}\log n)$. We reassemble the bound b_2 by its parts. b_{21} yields $O(n^{-1}\log n)$,

$b_{22} = O(\log n)$ and $b_{23} = O(n^{-1} \log n)$. Hence, we can sum up the asymptotics to $b_2 = O(\log n + n^{-1} \log n)$. As we have already seen, the second term converges to 0. However, the first term grows with $O(\log n)$, thus, the approximation does not converge to 0 but tends to $+\infty$. In contrast, if we assume that the words in $\mathcal{W}$ do not overlap with each other, we attain an improved bound $b_{22} = 0$ and, thus, an overall error of $O(n^{-1} \log n)$ which converges to 0.

In summary, the Poisson approximation for the number of clumps only works for a limited number of words, which are not allowed to overlap. For overlapping words, the error does not converge to 0. The reason is that we do not model the overlap structure between words accurately. Hence, one might use a compound Poisson approximation, which considers these dependencies.

Compound Poisson Approximation

The main result for a compound Poisson approximation is published in Reinert and Schbath (1998) for the general Markov case (for an overview, see Reinert *et al.*, 2000, 2005). The special case of an independent sequence model substantially simplifies the formulae (Reinert and Schbath, 1999). After presenting the approach of the independent seequence model, we will see that two problems emerge. Therefore, we describe an improved approach circumventing these problems (Roquain and Schbath, 2007) by considering heterogeneous clumps (see Section 3.5.2).

Classical Approach The classical approach is similar to the compound Poisson distribution for a single word (see Section 3.3.3) based on the probability for clumps of size k. First, we define the set of periods for two words using the overlap bit $\epsilon_d(w, v)$

$$\Upsilon(w, v) := \{\eta \in \{1, \ldots, \ell - 1\} : \epsilon_\eta(w, v) = 1\} \tag{3.39}$$

Hence, the set of periods $\Upsilon(w, v)$ contains the possible overlaps of v occuring after w. The proof also requires an assumption

A1 $\forall_{w \neq v}\, v$ is not a substring of any homogeneous 2-clump of w.

This assumption ensures that any word v can only partly overlap any clump of any word w. Otherwise, v could occur within a clump. Thus, observing such a clump would automatically imply an occurrence of v. Such strong dependencies are excluded by assumption **A1**.

Example 3.17. *The two words v = 'CACA' and v = 'ACAC' do not fulfil assumption* **A1** *since the 2-clump 'CACACA' contains a complete occurrence of v starting at position 2.* □

To state the bound for the compound Poisson approximation, we have to introduce some notation:

$$\begin{aligned} M(w,v) &:= \mathbf{1}(w \neq v) \sum_{\eta \in \Upsilon(w,v)} \frac{1}{\mu(v_1,\ldots,v_{\ell-\eta})}, \\ R(w,v) &:= (n-\ell+1)\left[(\ell-1)(\mu(w)\tilde{\mu}(v)+\tilde{\mu}(w)\mu(v))+(4\ell-3)\tilde{\mu}(w)\tilde{\mu}(v)\right], \\ T(w,v) &:= (n-\ell+1)\mu(w)\mu(v)\left[4(\ell-1)+M(w,v)+M(v,w)\right]. \end{aligned}$$

We consider the compound Poisson random variable $\sum_{k\geq 1} k \sum_{w\in\mathcal{W}} \tilde{Z}_k(w)$ where $\tilde{Z}_k(w)$ denotes the number of k-clumps of word w, see Eq. (3.28), and has the expected value

$$\mathbb{E}[\tilde{Z}_k(w)] = (n-\ell+1)\tilde{\mu}_k(w).$$

Hence, we want to bound the total variation distance between the number of all counts $\sum_{w\in\mathcal{W}} N_n(w)$ and the corresponding compound Poisson process $\mathcal{CP}$ as described above. The proof (Reinert and Schbath, 1999) is similar to the one-dimensional case. In fact, one computes the second moments by considering each k-clump with any possible words occuring just before and after the clump. Distinguishing the two cases that a k-clump of a word w does respectively does not overlap with a k'-clump of a word v, yields the bounds

$$d_{\mathrm{TV}}\left(\mathcal{L}\left(\sum_{w\in\mathcal{W}} N_n(w)\right), \mathcal{CP}\right) \leq \sum_{v,w\in\mathcal{W}} \left[R(w,v)+T(w,v)\right] + 2\sum_{w\in\mathcal{W}} (\ell-1)\left[\mu(w)-\tilde{\mu}(w)\right].$$

Next, we compute the asymptotics for constant number of words. The bound for $R(w,v)$ is $O(n^{-1}\log n)$, for $T(w,v)$ one obtains $O(n^{-1}\log n)+O(\sum_{w\neq v} n^{-1}(M(w,v)+M(v,w)))$, and for the last term $O(n^{-1}\log n)$. Obviously, the total variation distance is strongly influenced by $M(w,v)+M(v,w)$ catching the between-word overlapping structure. The more overlaps are possible, the bigger they become. If the words do not overlap, the set of periods is empty, thus, $M(w,v) = M(v,w) = 0$. Therefore, the total variation distance converges to 0 for non-overlapping words but does not necessarily do so for overlapping words.

Normal Approximation

Based on the exact variance, we can derive a normal approximation for frequent words in long sequences. In vector notation, we define the asymptotic expected value by

$$\tilde{\boldsymbol{\mu}} := \lim_{n\to\infty} n^{-1}\mathbb{E}[(\tilde{N}_n(w^{(1)},\ldots,\tilde{N}_n(w^{(q)})] = (\tilde{\mu}(w^{(1)}),\ldots,\tilde{\mu}(w^{(q)})). \tag{3.40}$$

The variance given in Eq. (3.38) asymptotically becomes

$$\begin{aligned}\sigma^2_{v,w} &:= \lim_{n\to\infty} n^{-1}\,\mathbb{C}\mathrm{ov}[\tilde{N}_n(w),\tilde{N}_n(v)] \\ &= \left(\sum_{k=\ell}^{2\ell-1}\sum_{d=\ell}^{k}\tilde{\mu}(w)\varpi_d(v|w)+\tilde{\mu}(v)\varpi_d(w|v)\right) \\ &\quad +\tilde{\mu}(w)\left[\epsilon_0(w,v)+\tilde{\mu}(v)\right]-2\ell\tilde{\mu}(w)\tilde{\mu}(v).\end{aligned}$$

The number of counts of homogeneous clumps $\tilde{\mathbf{N}}_n := (\tilde{N}_n(w^{(1)}),\ldots,\tilde{N}_n(w^{(q)}))$ is a multi-dimensional normal distribution. Let the asymptotic covariance matrix be

$$\Sigma := (\sigma^2_{v,w})_{v,w\in\mathcal{W}}.$$

Then, the count vector $n^{-1}\tilde{\mathbf{N}}_n$ is asymptotically normal with mean $\tilde{\boldsymbol{\mu}}$ and covariance matrix $n^{-1}\Sigma$. For $\det(\Sigma)\neq 0$, we obtain the multi-dimensional standard normal distribution by

$$\sqrt{n}\Sigma^{-1/2}\left(n^{-1}\tilde{\mathbf{N}}_n-\tilde{\boldsymbol{\mu}}\right)\sim\mathcal{N}(\mathbf{0},\mathbf{1}).$$

Similar to the other normal approximations, we can transform the count vector to be χ^2 distributed. Also, the summed number of clumps is obtained by multiplying the q-dimensional column vector $t=(1,1,\ldots,1)^T$ from the right to the count vector $\tilde{\mathbf{N}}_n$ yielding $n^{-1}\tilde{\mathbf{N}}_n t\sim\mathcal{N}(\tilde{\boldsymbol{\mu}}t,n^{-1}t^T\Sigma t)$. Hence, $\tilde{N}_n(\mathcal{W})\sim\mathcal{N}(\tilde{\mu}(\mathcal{W}),n^{-1}\sigma^2)$ with $\sigma^2:=\sum_{w\in\mathcal{W}}\sum_{v\in\mathcal{W}}\sigma^2_{w,v}$.

3.5.2 Heterogeneous Clumps

Calculation of moments for heterogeneous clumps is much more complicated due to more dependencies. Therefore, we only show how to compute the expected value such that the additional dependencies become obvious. In a second step, we present a compound Poisson approximation (Roquain and Schbath, 2007).

First Moment

Before we can state the expected value of the number of heterogeneous clumps, we have to compute the probability $\breve{\mu}(\mathcal{W})$ of an occurrence of a heterogeneous clump. In contrast to homogeneous clumps, heterogeneous clumps start if any of the words $\mathcal{W}$ occurs without any overlapping occurrence of *any* word in $\mathcal{W}$ (see Eq. (3.36)). For homogeneous clumps, the preceding occurrences are disjoint since two principial roots of the same word can never occur at the same time (Schbath *et al.*, 1995). Thus, we obtain the self-overlap probability $\omega(w)$ by summing over all principal roots (see Eq. (3.18)). For heterogeneous clumps, this is not possible since the preceding occurrences are not necessarily disjoint. Consider the following example:

Example 3.18. *Given the set of words* $\mathcal{W} = \{$*'AAC', 'ACA', 'CAA'*$\}$*, a clump at position* i *can start by an occurrence of any of the words which does not overlap with any preceding occurrence of* $\mathcal{W}$*. A clump starting with 'CAA' does not allow a preceding 'AA' as this would be an overlapping occurrence of 'AAC' at position* $i-2$*. Furthermore, an occurrence of 'ACA' would be implied at position* $i-1$*.* □

The problem can be solved by computing the *principal* periods between two words (Chryssaphinou *et al.*, 2001; Roquain and Schbath, 2007) with respect to $\mathcal{W}$:

$$\Upsilon'_{\mathcal{W}}(w,v) := \begin{cases} \Upsilon'(w) & \text{if } w = v, \\ \{\eta \in \Upsilon(w,v) : \forall_{u\in\mathcal{W}} \forall_{j\in\Upsilon(w,u)} \eta - j \notin \Upsilon(u,v)\} & \text{otherwise.} \end{cases} \tag{3.41}$$

Thus, the set of prinicipal periods contains the possible overlaps of w followed by v such that there is no occurrence of $\mathcal{W}$ inbetween. Hence, the principal periods are disjoint among $\mathcal{W}$. Therefore, we can use them similarily to the principal periods of one word. However, the multi dimensionality yields a matrix Ω with entries $\omega_{w,v}$ for the possible overlaps between w and a subsequent v. We define $\omega_{w,v}$ by

$$\omega_{w,v} := \sum_{\eta\in\Upsilon'(w,v)} \prod_{\kappa=1}^{\eta} \mu(v_\kappa). \tag{3.42}$$

and retrieve as overlap matrix $\Omega = (\omega_{w,v})_{w,v\in\mathcal{W}}$. The probability $\breve{\mu}_k(\mathcal{W})$ to observe a k-clump means that one observes an occurrence without any previous overlapping occurrence (which is $\mathbf{1} - \Omega$ with $\mathbf{1}$ being the identity matrix of dimension $q \times q$), $k-1$ overlapping occurrences (Ω^{k-1}) and, finally, no further overlapping occurrence ($\mathbf{1} - \Omega$). With $\boldsymbol{\mu}(\mathcal{W}) = (\mu(w))_{w\in\mathcal{W}}$, this yields

$$\breve{\mu}_k(\mathcal{W}) = \|\Omega^{k-1}(\mathbf{1} - \Omega)^2 \boldsymbol{\mu}(\mathcal{W})\|_1 \tag{3.43}$$

where $\|\cdot\|_1$ denotes the 1-norm defined by $\|\boldsymbol{z}\|_1 := \sum_{i=1}^{d} |z_i|$ for all $\boldsymbol{z} \in \mathbb{R}^d$.

Hence, we obtain for the expected value of the number of counts of heterogeneous clumps $\mathbb{E}[\breve{N}_n(\mathcal{W})] = (n - \ell + 1)\sum_{k\geq 1} k\breve{\mu}_k(\mathcal{W})$. Note that Reinert *et al.* (2005, Section 1.6.2) also report the expected value of the number of heterogeneous (mixed) clumps based on Chryssaphinou *et al.* (2001). Erroneously, the stated formula calculates the expected value of the number of renewal counts and not of heterogeneous clumps. However, both counts are asymptotically similar (Chryssaphinou *et al.*, 2001). Renewal counts which are related to heterogeneous clumps, are described extensively in literature (Breen *et al.*, 1985; Biggins and Cannings, 1987; Chryssaphinou and Papastavridis, 1988; Tanushev and Arratia, 1997).

Compound Poisson Approximation

The compound Poisson approach considering homogeneous clumps (see Section 3.5.1) suffers from two major drawbacks: First, the assumption **A1** about the overlapping structure of the set of words might not always be fulfilled. Second, the bound for the approximation does not converge to zero for strongly overlapping words. The reason for the second disadvantage is that the overlaps between different words are not considered in the compound Poisson approximation. In contrast, dealing with heterogeneous clumps incorporates such dependencies. Based on this idea, a recently published approach solves these problems (Roquain and Schbath, 2007) which is presented in the following.

In Eq. (3.43), the probability $\breve{\mu}_k(\mathcal{W})$ for the occurrence of a heterogeneous k-clump is stated. Denoting the number of heterogeneous k-clumps by $\breve{Z}_k(\mathcal{W})$ yields as expected value

$$\mathbb{E}[\breve{Z}_k(\mathcal{W})] = (n - \ell + 1)\breve{\mu}_k(\mathcal{W}).$$

Based on this expected value, one can define a compound Poisson distribution $\mathcal{CP}$ similar to the homogeneous case by $\sum_{k\geq 1} k\breve{Z}_k(\mathcal{W})$. The total variation distance for the i.i.d. sequence model can be bounded by

$$d_{\mathrm{TV}}\left(\mathcal{L}(N_n(\mathcal{W})), \mathcal{CP}\right) \leq 18(n - \ell + 1)\ell\mu(\mathcal{W})^2 + 2\ell\mu(\mathcal{W}), \tag{3.44}$$

which slightly improves the bound for the Markov case in Roquain and Schbath (2007). The bound is proven by using Chen-Stein with an index set for random indicator variables for each position and for each k-clump. Iterating over the possible words which are a k-clumps yields above bounds (for the proof, see Roquain and Schbath, 2007). For the number $\breve{N}_n(\mathcal{W})$ of clumps, one obtains the bound

$$d_{\mathrm{TV}}\left(\mathcal{L}(\breve{N}_n(\mathcal{W})), \mathcal{P}\right) \leq 18(n - \ell + 1)\ell\mu(\mathcal{W})^2$$

where $\mathcal{P}$ denotes the Poisson distribution for $\sum_{k\geq 1} \breve{Z}_k(\mathcal{W})$. Under the rare word condition, both bounds have the same asymptotic behaviour since both summands in Eq. (3.44) yield $O(n^{-1}\log n)$. Hence, the approximation error converges to 0 independent of the (self) overlap of the words.

Chapter 4

Generating Functions

Generating functions can be used to represent an infinite sequence of numbers in a closed form. Since discrete probability distributions on $\mathbb{N}$ may be infinite, they are often presented as generating functions, as well as recursive formulae. As we have seen in the previous chapter, the exact formula for the count distribution is a complicated recursive formula. Here, we derive generating functions for the number of counts of words and clumps, which are easier to understand and to manipulate. After summarizing basic properties and transformations of generating functions, we focus on single words. After deriving the formulae for the absence probability based on the waiting time of the first occurrence, we can compute the count probability distribution (Goulden and Jackson, 1983; Fudos *et al.*, 1996; Koutras, 1997; Régnier and Szpankowski, 1998; Robin and Daudin, 1999; Noonan and Zeilberger, 1999; Rahmann, 2000). For word counts, our exposition is based on Rahmann (2000). Finally, we consider clumps following the presentation in Stefanov *et al.* (2007). Derivation of generating functions for set of words is more complicated since the order of word occurrences matter. Hence, we don't cover this and refer to Robin and Daudin (2001); Bassino *et al.* (2007).

4.1 Preliminaries

A generating function is a formal power series (Niven, 1969). It is used to represent finite and infinite sequences. Let $g = \langle g_n \rangle = \langle g_0, g_1, g_2, \ldots \rangle$ for $n \geq 0$ denote an infinite sequence, one can encode the elements of the sequence g_n as coefficients of a polynomial using a 'dummy' variable z:

$$g(z) = \sum_{n \geq 0} g_n z^n.$$

The coefficient g_n of $g(z)$ is also denoted by $[z^n]g(z)$. There are two different views to look at generating functions. On the one hand, $g(z)$ can be seen as a function of a complex variable z. Then, convergence of $g(z)$ becomes an issue. On the other hand, one can focus on the sequence represented by the formal power series (Graham *et al.*, 1994, Chapter 7). In this case, one can manipulate $g(z)$ without considering convergence. See Wilf (1994) for a detailled exposition of both views.

To derive the count distribution, we only need a few basic closed forms and manipulations. For the sequence $g = \langle 1, 1, 1, \ldots \rangle$, one obtains the closed formula $1/(1-z)$ since

$$\begin{aligned}(1-z)\cdot\sum_{n\geq 0} z^n &= 1+z+z^2+z^3+\ldots \\ &\quad -z-z^2-z^3-\ldots \\ &= 1.\end{aligned}$$

A basic manipulation is to shift a sequence $\langle g_0, g_1, \ldots\rangle$ by k positions to the right and filling the k left most positions with zeros: $\langle 0,0,\ldots,0,g_0,g_1,\ldots\rangle$. The respective power series operation is multiplication by z^k. Thus, we obtain

$$g(z)\cdot z^k = \sum_{n\geq k} g_{n-k} z^n.$$

Furthermore, it is obvious to see that for two generating functions $g(z)$ and $f(z)$ we obtain for the sum $g(z)+f(z)$ the sum of the coefficients $\langle g_n + f_n\rangle$.

The final operation we need is the convolution $h(z) = g(z)f(z)$. Calculating the product of both sums and ordering the result by the powers of z, we obtain $h_n = \sum_{k=0}^{n} g_{n-k} f_k$. A special case occurs if $f(z) = 1/(1-z)$. In this case, the resulting sequence contains the cumulative sums:

$$\frac{g(z)}{1-z} = \sum_{n\geq 0}\left[\left(\sum_{k=0}^{n} g_k\right) z^n\right].$$

For more manipulations and elementary generating functions, see Comtet (1974)

Probability Generating Function As mentioned in the introduction, one can use a generating function to represent a discrete probability distribution. For a random variable X which takes values in $\mathbb{N}_{\geq 0}$ with $\mathbb{P}(X=n) = g_n$, the probability generating function $g(z)$ is defined by

$$g(z) := \sum_{n\geq 0} g_n z^n$$

with the additional requirement that the probabilities sum up to 1 which means that $g(1) = 1$.

Example 4.1. *Let X be a* shifted *geometrically distributed random variable with parameter p, thus, $\mathbb{P}(X=n) = (1-p)^n p$. Hence, the probability generating function $g(z)$ is*

$$g(z) = \sum_{n\geq 0}(1-p)^n p z^n = \frac{p}{1-(1-p)z}.$$

The last step is obained by shifting p in front of the sum, using $(1-p)^n z^n = [(1-p)z]^n$ and then applying $\sum_{n\geq 0} x^n = (1-x)^{-1}$ for $x = (1-p)z$.

□

The probability generating function $h(z)$ of the sum of two independent discrete random variables X and Y with probability generating functions $g(z)$ and $f(z)$ is the product $h(z) = g(z)f(z)$. This can be easily seen by writing the probability for $X + Y = n$. Given that $Y = k$, we neccessarily obtain $X = n - k$. Since one has to sum up the probabilities for all such decompositions of n, one yields $h_n = \sum_{k=0}^{n} g_{n-k} f_k$, which is the convolution of both distributions.

One can also express a sum with a random number of terms as a probability generating function. Let X_i be i.i.d. random variables with probability generating function $g(z)$ with coefficients g_n and Y with probability generating function $f(z)$ with coefficients f_n. If Y is independent of all X_i, the random variable $\sum_{y=1}^{Y} X_y$ has the probability generating function

$$\sum_{n\geq 0} f_n \left[\sum_{i\geq 0}^{n} g_i z^i\right]^n = f\left(g(z)\right).$$

Here, we use that the sum of n i.i.d. random variables is the nth power of its probability generating function since one has to perform n convolutions.

4.2 Single Word Count Distribution

We consider a word w of length ℓ. For simplified notation, we drop the index/parameter w if the word is obvious from context. For derivation of generating functions, it is more appropriate to define the indicator random variables for an occurrence to be 1 at the last position of the occurrence. Therefore, we define

$$\underline{Y}_i := \begin{cases} 0 & : 1 \leq i < \ell \\ Y_{i-\ell} & : \ell \leq i \leq n \end{cases} \tag{4.1}$$

The main difference is that the indicators at the very beginning of the sequence are 0 instead of the indicators at the last $\ell - 1$ position of the sequence. This simplifies recursive calculation of certain probabilities. We obtain for the occurrence probability $\mathbb{P}_\mu(\underline{Y}_i = 1) = \mu(w)$ for $i \geq \ell$ where $\mu(w)$ denotes the probability for word w.

4.2.1 Absence Probability

The absence probability a_n is the probabiltiy that there is no occurence of the word in the sequence of length n:

$$a_n := \mathbb{P}_\mu(\sum_{i=1}^{n} \underline{Y}_i = 0)$$

We can express the absence probabiltiy by using the waiting time till the 1st occurence: If the first occurence is after the nth position, the word is absent. Let W be the random

variable for the position of the first occurence (in fact, it is the position where the first hit ends). Due to the new definition of an occurrence, $\underline{T}_1$ cannot be less than ℓ. Defining $t_0 = 0, t_n := \mathbb{P}_\mu(\underline{T}_1 = n)$ for $n > 0$, we obtain for the absence probability

$$a_n = \mathbb{P}_\mu(\underline{T}_1 > n) = 1 - \mathbb{P}_\mu(\underline{T}_1 \leq n) = 1 - \sum_{i=0}^{n} t_i. \tag{4.2}$$

Thus, given the probability distribution for the first occurence, we can simply compute the absence probability. Therefore, we introduce the return probability in the next paragraph to be able to derive formulae for the waiting time afterwards.

Return Probability

The return probability r_n is the probability to observe an occurrence n steps after another occurrence. There can also be further occurrences in between. Thus, we can define independently of i

$$r_n := \mathbb{P}_\mu(\underline{Y}_{i+n} = 1 | \underline{Y}_i = 1) = \mathbb{P}_\mu(\underline{Y}_{i+n} = 1 | \underline{T}_1 = i).$$

Due to the i.i.d. property of the sequence model, the occurrence probabilities are independent of the waiting time if the occurrences do not overlap, thus, $\mathbb{P}_\mu(\underline{Y}_{i+n} = 1 | \underline{T}_1 = i) = \mathbb{P}_\mu(\underline{Y}_{i+n}) = \mu(w)$ for $n \geq \ell$. For overlapping occurrences, we can use the overlap bit ϵ_d defined in Eq. (3.6). Since the return probability for $n = 0$ is obviously 1, we obtain for the return probabilities

$$r_n = \begin{cases} 1 & \text{if } n = 0, \\ \epsilon_n \prod_{\kappa=\ell-n+1}^{\ell} \mu(w_\kappa) & \text{if } 1 \leq n < \ell, \\ \mu(w) & \text{otherwise.} \end{cases}$$

Capturing the overlap in the correlation polynomial $c(z)$ with $c(z) := \sum_{n=0}^{\ell-1} \epsilon_n z^n$, the first ℓ return probabilities in $d(z) := \sum_{n=0}^{\ell-1} r_n z^n$, we obtain for the generating function $r(z) := \sum_{n\geq 0} r_n z^n$ the expression

$$r(z) = d(z) + \frac{\mu(w) z^\ell}{1 - z}. \tag{4.3}$$

Note that the second term is derived by shifting the generating function $\sum_{n\geq 0} \mu(w) z^n = (1 - z)^{-1} \mu(w)$ by ℓ positions to the right (multiplying with z^ℓ).

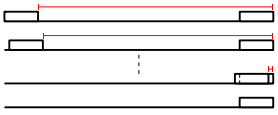

Figure 4.1: Decomposition of the event of an occurrence at position n by the waiting time and the return probability. The upper row shows a waiting time of ℓ for the first occurrence and a *returned* occurrence after $n-\ell$ positions. In the second row, the waiting time is $\ell+1$ and the return after $n-\ell-1$ steps. One has to consider all such events till the waiting time is n (last row).

Waiting Time

The probability for the waiting time for the first occurrence at the first $\ell-1$ positions equals to 0 since the first occurrence is not before position ℓ due to the definition of the occurrence indicators $\underline{Y}_i$. Hence, we obtain $t_0 = t_1 = \ldots = t_{\ell-1} = 0$. Based on the results for the return probability, we can compute the remaining waiting time probabilities

We decompose the event $\{\underline{Y}_n = 1\}$ of an occurrence at position n. The occurrence at n can be expressed in terms of the waiting time (see Fig. 4.1 for an illustration): For all $\ell \leq i \leq n$, the first occurrence is at position i and another occurrence *returns* after $n-i$ positions. For $i = n$, we consider the possibility that there is no occurrence before n. Hence, we obtain for the event $\{\underline{Y}_n = 1\} \equiv \bigcup_{i=\ell}^{n} \{\underline{Y}_n = 1, \underline{T}_1 = i\}$. Conditioning the events on $\underline{T}_1 = i$, one obtains for $n \geq \ell$

$$\mathbb{P}_\mu(\underline{Y}_n = 1) = \sum_{i=0}^{n} \mathbb{P}_\mu(\underline{Y}_n = 1 | \underline{T}_1 = i) \mathbb{P}_\mu(\underline{T}_1 = i) = \sum_{i=0}^{n} r_{n-i} t_i. \tag{4.4}$$

Hence, we can recursively compute t_n by solving above equation for the last term in the sum:

$$t_n = \mu(w) - \sum_{i=0}^{n-1} r_{n-i} t_i.$$

Defining the generating function for the waiting time by $t(z) := \sum_{n\geq 0} t_n z^n$, we can express it by considering that the sum in Eq. (4.4) is the convolution of $r(z)$ and $t(z)$. Furthermore, Eq. (4.4) equals to 0 for $n < \ell$. Hence, we can use for the left hand side $(1-z)^{-1}\mu(w)z^\ell$ and obtain

$$\frac{\mu(w)z^\ell}{1-z} = r(z)t(z).$$

Solving for $t(z)$, we obtain

$$t(z) = \frac{\mu(w)z^\ell}{r(z)(1-z)}. \tag{4.5}$$

Absence Probability

In Eq. (4.2), the absence probability is defined in terms of the waiting time. Based on the absence probability $a_n = 1 - \sum_{i=0}^{n} t_i$ in a sequence of length n, we obtain the generating function $a(z) := \sum_{n\geq 0} a_n z^n$ for the absence probability taking the product of $1 - \sum_{i=0}^{n} t_i$ and $1 - z$:

$$a(z) = \frac{1 - t(z)}{1 - z}. \tag{4.6}$$

Example 4.2. *We consider the word $w =$'ACA' in a two-letter alphabet with $\mu($'A'$) = p$. Hence, the occurrence probability is $\mu(w) = p^2(1-p)$. The correlation polynomial is given by $c(z) = 1 + z^2$ since the word overlaps at position 0 and 2. The generating function for the return probability is*

$$r(z) = 1 + p(1-p)z^2 + \frac{p^2(1-p)z^3}{1-z}.$$

The generating function for the waiting time yields

$$t(z) = \frac{p^2(1-p)z^3}{(1-z)\left(1 + p(1-p)z^2 + \frac{p^2(1-p)z^3}{1-z}\right)} = \frac{p^2(1-p)z^3}{1 - z - (p^2-p)z^2 - (p - 2p^2 + p^3)z^3}.$$

For the generating function of the absence probability $a(z) = (1-t(z))/(1-z)$, one obtains

$$a(z) = \frac{1 + (p - p^2)z^2}{1 - z + (p - p^2)z^2 + (-p + 2p^2 - p^3)z^3}.$$

Solving for $[z^n]a(z)$ for $0 \leq n < 8$ yields

$$\begin{array}{lll} a_0 = a_1 = a_2 = 1, & a_3 = 1 - p^2 + p^3, & a_4 = 1 - 2p^2 + 2p^3, \\ a_5 = 1 - 3p^2 + 4p^3 - 2p^4 + p^5, & & a_6 = 1 - 4p^2 + 6p^3 - 3p^4 + p^6, \\ a_7 = 1 - 5p^2 + 8p^3 - 4p^4 + p^7. & & \end{array}$$

Obviously, the probability for no occurrence in a sequence with length less than 3 is 1 since the word is larger than the sequence. For a sequence of length 3, the absence probability is the complement of the occurrence probability: $1 - \mu(w) = a_3$. A sequence of length 4 can contain the word either at position 3 or 4. Since the occurrences cannot overlap, they cannot occur at the same time, therefore, they are disjoint. Hence, we obtain $a_4 = 1 - 2\mu(w)$. For longer sequences, it is more difficult to manually derive the probabilities, however, the generating function $a(z)$ contains all probabilities.

□

4.2.2 Number of Occurrences

Calculation of the absence probability is based on the waiting time till the first occurrence. Similarily, the count distribution can be computed by considering the waiting times till the kth occurrence. We derive the corresponding generating functions using the inter-occurrence time. This leads to the generating function for the probability distribution of the number of counts.

Inter-Occurrence Time

The inter-occurrence time is the distance between two successive occurrences. Denoting the waiting time till the kth occurrence by $\underline{T}_k$, the inter-occurrence time is defined for $k \geq 1$

$$V_k := \underline{T}_{k+1} - \underline{T}_k. \tag{4.7}$$

Note that V_k are identically and independently distributed. However, they are usually not equal in distribution to the waiting time $\underline{T} = \underline{T}_1$ till the first occurrence.

Simililarily to the computation of the return probabilities, we decompose the event of an occurrence at position $m+n$ given the first occurrence at m into the events with a first occurrence at m with a successive occurrence at $m+i$ and a further (not necessarily successive) occurrence at $m+n$. This yields

$$\{\underline{Y}_{m+n} = 1 | \underline{T} = m\} \equiv \bigcup_{i=1}^{n} \{\underline{Y}_{m+n} = 1, V_1 = i | \underline{T} = m\}.$$

In probabilities and, in a second step, using the independence between $\underline{T}$ and V_1, one obtains

$$\begin{aligned} \mathbb{P}_\mu(\underline{Y}_{m+n} = 1 | \underline{T} = m) &= \sum_{i=1}^{n} \mathbb{P}_\mu(\underline{Y}_{m+n} = 1 | V_1 = i, \underline{T} = m) \mathbb{P}_\mu(V_1 = i | \underline{T} = m) \\ &= \sum_{i=1}^{n} \mathbb{P}_\mu(\underline{Y}_{m+n} = 1 | \underline{Y}_{m+i} = 1) \mathbb{P}_\mu(V_1 = i). \end{aligned}$$

Since the left hand side is the return probability after n steps, we have with $v_n = \mathbb{P}_\mu(V_i = n)$ for $n > 0$

$$r_n = \sum_{i=1}^{n} r_{n-i} v_i.$$

Note that $v_0 = 0$ by definition because a successive occurrence cannot have a distance of 0. Hence, we obtain for the generating function $v(z) := \sum_{n \geq 0} v_n z^n$ the equation $r(z) - 1 = r(z)v(z)$ since $r_0 = 1$. Solving for $v(z)$ yields

$$v(z) = \frac{r(z)-1}{r(z)} = 1 - \frac{1}{r(z)}. \tag{4.8}$$

Waiting Time till kth Occurrence

Due to the independence and the identical distribution of V_i, we can state the waiting time till the kth occurrence by

$$\underline{T}_k = \underline{T} + \sum_{j=1}^{k-1} V_j \stackrel{d}{=} \underline{T} + (k-1)V_1,$$

where $\stackrel{d}{=}$ denotes equality in distribution. Since the probability for the sum of independent discrete random variables can be computed by the product of the corresponding probability generating functions, one directly obtains with $t_n^{(k)} := \mathbb{P}_\mu(\underline{T}_k = n)$ for $k \geq 1$ and $t^{(k)}(z) := \sum_{n\geq 0} t_n^{(k)} z^n$

$$t^{(k)}(z) = t(z)\left[v(z)\right]^{k-1}. \tag{4.9}$$

One can also write $t^{(k+1)}(z) = t^{(k)}(z)v(z)$. However, for the previous formula, we can plug in $t(z)$ and $v(z)$

$$t^{(k)}(z) = \frac{\mu(w)z^\ell\left[r(z)-1\right]^{k-1}}{(1-z)\left[r(z)\right]^k}.$$

Based on the generating function of the waiting time, we can derive generating functions of the number of counts.

Number of Occurrences

Let $\underline{N}_n = \sum_{i=1}^n \underline{Y}_i$ be the random variable for the number of occurrences in a sequence of length n. To compute its probability distribution, we use the same duality principle as in the previous chapter: The probability to observe at least k occurrences in a sequence of n is equivalent to the probability that the waiting time till the kth occurrence is at most n, in formula, $\{\underline{N}_n \geq k\} \equiv \{\underline{T}_k \leq n\}$. Based on the above generating functions, we obtain for the probability of k occurrences $\mathbb{P}_\mu(\underline{N}_n = k) = \mathbb{P}_\mu(\underline{T}_k \leq n) - \mathbb{P}_\mu(\underline{T}_{k+1} \leq n)$ and, therefore

$$f^{(k)}(z) = \frac{t^{(k)}(z) - t^{(k+1)}(z)}{1-z} = t^{(k)}(z)\frac{1-v(z)}{1-z}. \tag{4.10}$$

Note that this is no longer a probability generating function since the coefficient $z^n[f^{(k)}(z)]$ corresponds to the probability for k occurrences in a sequence of length n.

Example 4.3. *Continuing Ex. 4.2, we can compute the inter-arrival time and the waiting times till the kth occurrence. For the inter-arrival time, we obtain*

$$v(z) = \frac{pz^2\left(1-p-(1-2p+p^2)z\right)}{1-z+(p-p^2)z^2-(p-2p^2+p^3)z^3}$$

and for the waiting time till the kth occurrence

$$t^{(k)}(z) = \frac{\left[\frac{pz^2\left(1-p-(1-2p+p^2)z\right)}{1-z+(p-p^2)z^2-(p-2p^2+p^3)z^3}\right]^k pz}{1-(1-p)z}.$$

Finally, the probability for k counts in a sequence of length n is encoded in the nth coefficient of

$$f^{(k)}(z) = \frac{\left[\frac{pz^2\left(1-p-(1-2p+p^2)z\right)}{1-z+(p-p^2)z^2-(p-2p^2+p^3)z^3}\right]^k pz}{(1-z+(p-p^2)z^2-(p-2p+p^3)z^3)\,[1-(1-p)z]}.$$

Salvy and Zimmermann (1994) show how to evaluate the coefficients.

□

4.3 Number of Homogeneous Clumps

Instead of counting overlapping occurrences, one can also count the number of clumps. The exact distribution derived by generating functions has only recently be published (Stefanov *et al.*, 2007). After defining the indicator for a clump, we derive the formulae for the inter-arrival times of clumps. Based on this, we can directly compute the count distribution for the number of clumps.

As before, we define a clump to be an occurrence without any preceding overlapping occurrence. Similar to the last section, we define the positions of an occurrence at the end of the word in the sequence. Hence, we obtain for the clump indicator

$$\tilde{\underline{Y}}_i := \underline{Y}_i \prod_{d=1}^{\ell-1} (1-\underline{Y}_{i-d}). \tag{4.11}$$

4.3.1 Inter-arrival Time

The waiting time $\tilde{T}$ to observe the first clump is equal to the waiting time T till the first occurrence. This is obvious since the first clump always starts at the position of the first occurrence. The waiting time $\tilde{T}_k$ till the kth clump for $k > 1$ are not necessarily equal to the waiting time $\underline{T}_k$ till the kth occurrence (equality holds if the word is not self-overlapping). Hence, the same holds for the inter-arrival times $\tilde{V}_k = \tilde{T}_{k+1} - \tilde{T}_k$. Since $\tilde{V}_k$ are identically distributed, we denote the inter-arrival time of clumps by $\tilde{V}$.

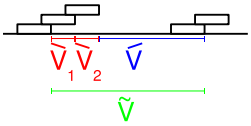

Figure 4.2: Decomposition of the inter-arrival time for a clump into (non-) overlapping inter-arrival times of occurrences..

The inter-arrival times between two clumps is composed of inter-arrival times between occurrences. In fact, the only occurrences between two clumps necessarily overlap with their preceding occurrence. Otherwise, the next clump would start at the first occurrence which does not overlap. Hence, depending on the size L of the preceding clump, the inter-arrival time to the next clump is the sum of L overlapping occurrence inter-arrival times and one non-overlapping occurrence inter-arrival time (see Fig. 4.2). Therefore, we define conditional random variables for overlapping inter-arrival times $\grave{V}_k$ and non-overlapping inter-arrival times $\acute{V}$ by

$$\grave{V}_k := (V_k | V_k < \ell), \quad \acute{V} := (V | V \geq \ell). \tag{4.12}$$

Then, we can easily express the clump inter-arrival time by its decomposition

$$\tilde{V} = \left(\sum_{k=1}^{Z} \grave{V}_k \right) + \acute{V} \tag{4.13}$$

where Z denotes the size of the preceding clump. The size of the clump is distributed according to a shifted geometric distribution $\mathbb{P}_\mu(Z = k) = \omega^k(1-\omega)$ with parameter $1-\omega$ where ω denotes the probability for a self-overlap. A self-overlap occurs if the occurrence inter-arrival time is less than ℓ. Thus, we obtain $\omega = \sum_{k=1}^{\ell-1} v_k$ where v_k denotes the probability $\mathbb{P}_\mu(V = k)$ as defined above.[1].

Generating Functions Now, we define the corresponding generating functions. The generating function $\grave{v}(z)$ for the overlapping inter-arrival times consists of the first $\ell - 1$ terms of $v(z)$. Due to $\mathbb{P}_\mu(\grave{V}) = \mathbb{P}_\mu(V|V < \ell) = \mathbb{P}_\mu(V)/\mathbb{P}_\mu(V < \ell) = \omega^{-1}\mathbb{P}_\mu(V)$, we obtain

[1]Note that this definition of ω is consistent with the definition in Chapter 3 in Eq. (3.18).

$\grave{v}(z) := \omega^{-1} \sum_{n=0}^{\ell-1} v_n z^n$. Accordingly, we have to consider the factor $1-\omega$ for the generating function $\acute{v}(z)$ for the non-overlapping inter-arrival times. By derivation from $v(z)$, we also have to remove the first $\ell - 1$ terms. Thus, we obtain

$$\acute{v}(z) := (1-\omega)^{-1} \left(v(z) - \sum_{n=0}^{\ell-1} v_n z^n \right). \tag{4.14}$$

Finally, the clump size Z is a shifted geometric distribution, thus, according to Ex. 4.1 its generating function is $g(z) := (1-\omega)(1-\omega z)^{-1}$. In Eq. (4.13), we sum up Z times $\grave{V}_k$. As described in the preliminaries, this is achieved by plugging $\grave{v}(z)$ into $g(z)$ under the assumption that Z is independent of $\grave{V}_k$. Multiplying by $\acute{v}(z)$ to consider the last term in Eq. (4.13) yields

$$\tilde{v}(z) := \frac{(1-\omega)\acute{v}(z)}{1-\omega\grave{v}(z)} = \frac{v(z) - \sum_{n=0}^{\ell-1} v_n z^n}{1 - \sum_{n=0}^{\ell-1} v_n z^n}. \tag{4.15}$$

4.3.2 Number of Counts

Based on the inter-arrival time for clumps, it is straight-forward to derive the number of counts. Again, we use the duality between the number of clumps and the waiting time till the kth clump. For the waiting time, we obtain the generating function

$$\tilde{t}^{(k)}(z) := t(z)\,[\tilde{v}(z)] \tag{4.16}$$

since the first occurrence of a clump is equal to the first occurrence of the word. Then, the generating function $\tilde{f}^{(k)}(z)$ for the number of k counts in a sequence of length n is

$$\tilde{f}^{(k)}(z) := \tilde{t}^{(k)}(z)\frac{1-\tilde{v}(z)}{1-z}. \tag{4.17}$$

Example 4.4. *Continuing the previous example, we obtain for the overlapping inter-arrival times* $\grave{v}(z) = z^2$ *since* $\omega = (p - p^2)$. *The clump inter-arrival time is*

$$\tilde{v}(z) = \frac{\left[1 - p - (1 - 2p + p^2)z + (1 - 3p + 3p^2 + p^3)z^2\right] p^2 z^3}{\left[1 - z + (p - p^2)z^2 - (p - 2p^2 + p^3)z^3\right]\left[1 - (p - p^2)z^2\right]}.$$

Now, we can compare the inter-arrival times of clumps and occurrences. For $n = 2$, *the main difference shows up: The probability for an inter-arrival time of occurrences of* 2 *is* $p - p^2$ *while for clumps the probability is* 0 *since clumps cannot overlap. For a distance of* 3, *both inter-arrival probabilities are* $p^2 - p^3$ *since no overlaps can be involved. The same is true for a distance of* 4 *yielding a probability of* $p^3 - p^4$. *However, for longer distances the probabilties differ again because the preceding clump can have size larger than* 1. *A distance*

of 5 *yields a occurrence inter-arrival probability of* $p^2 - 3p^3 + 4p^4 - 2p^5$ *while the clumps inter-arrival probability is* $p^2 - 2p^3 + 2p^4 - p^5$*. In fact, the corresponding probabilities for the inter-arrival of clumps are always larger than for occurrences. This is due to the fact that the inter-arrival probabilities for clumps have to compensate for the big difference for the distance of* 2*. Furthermore, we have already seen in the previous chapter that the distances between clumps are in average larger than for the corresponding word occurrences if the word allows self-overlaps.*

□

Chapter 5

TF Count Statistics

5.1 Introduction

TFs are often represented as PFMs. As described in Section 2.4.4, the number of compatible words of a PFM depends on the chosen threshold selection method. Here, we focus on the selection methods based on the error probabilities. In this case, the number of compatible words is asymptotically $O(|\mathfrak{A}|^{\ell})$ where $\mathfrak{A}$ is the alphabet and ℓ the length of the PFM. Hence, they cannot be enumerated efficiently. Furthermore, such a high number of words makes every word counting approach infeasible. This explicitly includes the conditional approach from Zhang *et al.* (2007) as we show in Chapter 7. Therefore, many heuristic approaches have been proposed comprising Ahab (Rajewsky *et al.*, 2002), Clover (Frith *et al.*, 2004), Consensus (Hertz and Stormo, 1999), MEME (Bailey and Elkan, 1994; Bailey and Gribskov, 1998), MotifScanner (Thijs *et al.*, 2001) and p-value calculation for multiple alignments (Nagarajan *et al.*, 2005). However, these methods are based on heuristically selected statistical criteria instead of a rigorous derivation of the count distribution. Furthermore, there are no methods which incorporate the complementary strand into the analysis which is important for palindromic PFMs.

In this chapter, we present our main contribution: we propose an approximation based on the compound Poisson distribution for the number of occurrences of a PFM without enumerating all compatible words. Furthermore, we incorporate both strands of the DNA molecules. We explicitly consider dependencies of overlapping hits. As background model, we use a symmetric i.i.d. model incorporating the average GC content of the upstream sequence, which can be justified by Chargaff's second law. We restrict ourselves to the GC content instead of base pair composition to make the computation invariant with respect to the choice of the leading strand.

A comparison of the new approach with existing approximations based on word counting (Schbath, 1995a; Waterman, 2000; Roquain and Schbath, 2007) is given in Chapter 7. In contrast to the most recent exact calculation (Zhang *et al.*, 2007), its complexity neither depends on the number of compatible words nor on the sequence length.

While repeating the statistical framework with its notation from the word count chapter, we relate the word counting approaches to TF counts. Subsequently, we derive the first two moments of the count distribution asymptotically. The third section contains the development of the count statistic approximation based on the second moments. Since we explicitly model the self-overlap of the PFM that results in dependencies, we can calculate two characteristic values for the self-overlap and the palindromicity of a PFM. In the next section,

we present a generating function formalism for TF under an equi-probable sequence model and by restriction to one strand. However, if the assumption of equi-probable nucleotide distribution is not fulfilled, the formalism still yields a feasible approximation. Finally, we introduce an algorithm to compute the important second moments.

5.2 Statistical Framework

Each PFM represents a DNA motif. It contains specific probabilities for each nucleotide at every position. We assume that the binding sites of each TF are described by only one PFM. An extension to more than one PFM can be obtained by choosing the representative PFM of a cluster (see Chapter 10). The position specific scoring matrix (PSM) $\Psi^A_{\kappa,a}$ of TF A is chosen to be the log-likelihood ratios of the nucleotide distribution of the PFM and the background probabilities $\mu(a)$ for every position κ and for nucleotides $a \in \mathfrak{A}$. We denote the length of the PSM by ℓ_A.

Using the PSM, we can assign a score to every position of the potential binding site depending on the observed nucleotide. Sliding a window of length ℓ_A over a random sequence $X_1 \dots X_n$ and summing up the scores in each window, yields a random score $S_j(A)$ for every position j of the sequence

$$S_j(A) = s_A(X_j \dots X_{j+\ell-1}) := \sum_{\kappa=0}^{\ell-1} \Psi^A_{\kappa,X_{j+\kappa}}. \tag{5.1}$$

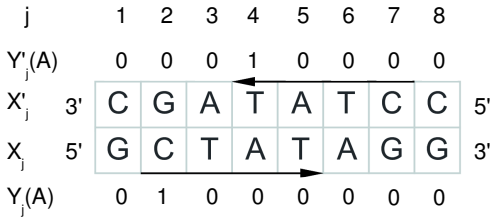

Figure 5.1: Example for our notation: The lower 5'-3' strand is the leading strand. Given a motif CTAT, there are two overlapping occurrences on the shown sequence region. $Y_1(A) = Y'_3(A) = 1$ indicate a hit starting at position 1 on the leading strand and another hit ending at position 3 on the complementary strand. This definition of $Y_j(A)$ and $Y'_j(A)$ simplifies notation.

Considering the complementary strand requires additional notation: In general, we use the same variables with a prime for this purpose. We call the strand of the corresponding gene the $5' - 3'$ strand. Correspondingly, the complementary strand is called the $3' - 5'$ strand. In contrast to the $5' - 3'$ strand, we assign a hit to the position at the complementary strand where the actual hit ends (see Figure 5.1). This means, that $S'_j(A)$ refers to the score of the nucleotides $X'_{j+\ell-1} \dots X'_j$ where X'_k denotes the complementary letter of X_k.

We call a position a *hit* if the corresponding window yields a score s higher than a certain threshold t_A. Denoting a hit at position j by the indicator random variable $Y_j = 1$, we

obtain the definition: $Y_j(A) := \mathbf{1}[S_j(A) \geq t_A]$. Similarily, a detected binding site on the complementary strand at position j is denoted by $Y'_j(A) = 1$ (see Figure 5.1).

The probability of an occurrence in random sequences can be calculated using the score distribution (see Section 2.4.2). One obtains

$$\mathbb{P}_\mu(Y_j(A) = 1) = \mathbb{P}_\mu(S_j(A) \geq t_A) = \alpha_A.$$

All positions including the complementary strand are identically distributed. However, due to possible overlaps the positions are not independent. This is the main difficulty to compute the second moment and the count distribution.

The number of counts is defined by summing over all positions and both strand. Thus, we obtain for a sequence of length n

$$N_n(A) := \sum_{j=1}^{n-\ell_A+1} \left(Y_j(A) + Y'_j(A)\right).$$

5.2.1 Relation to Word Count Approaches

Instead of assigning a score to each position of a sequence, we can determine for each word $a \in \mathfrak{A}^{\ell_A}$ its score $s_A(a)$. Each word corresponding to a hit ($s_A(a) \geq t_A$) is called a compatible word of A. The set of all compatible words is denoted by $\mathcal{A}$. We introduce indicator random variables $Y_j(a)$ which are 1 if the word at position j of the sequence is a and otherwise 0. Since a hit of TF A at position j occurs if the word at position j of the sequence is in $\mathcal{A}$, we obtain the equivalence

$$Y_j(A) \equiv \sum_{a \in \mathcal{A}} Y_j(a),$$

since hits are necessarily disjoint at each position.

5.3 First Two Moments

Before we present a new approximation of the count distribution, we compute the exact expected value, as well as the asymptotic variance. Then, we extend the variance to a pair of TFs obtaining the asymptotic covariance. Later, we show how to derive a similarity measure (Chapter 9), a clustering (Chapter 10), and a quality measure (Chapter 11) based on this concept.

5.3.1 Expected Value

The expected value is fairly easy to derive since position dependencies do not matter. Given the probability of an occurrence by α_A and a sequence length n, one obtains

$$\mathbb{E}[N_n(A)] = \sum_{j=1}^{n-\ell_A+1} \big(\mathbb{E}[Y_j(A)] + \mathbb{E}[Y'_j(A)]\big) = 2(n - \ell_A + 1)\alpha_A.$$

Asymptotically with $n \to \infty$, the expected value is

$$\lim_{n\to\infty} n^{-1}\mathbb{E}[N_n(A)] = 2\alpha_A.$$

5.3.2 Variance

As usual, the second moment is more complicated to compute due to the position dependencies introduced by possible overlapping occurrences. However, we will see that one can compute the exact asymptotic variance without knowing the set of compatible words. We introduce the variance by shortly reviewing the approach for set of words as presented in Section 3.4.3 for the normal approximation.

Asymptotic Variance for Words

Before we can state the asymptotic variance, we repeat some further notation regarding the self-overlap of a word $a \in \mathfrak{A}^\ell$ (for a more detailed exposition, see Section 3.4.1). We define the probabilities $\gamma_d(a)$ for a self-overlap of word a at position d. For that, we use the overlap bit $\epsilon_d(a)$ which is 1 if the word allows the overlap ($a_d = a_1, a_{d+1} = a_2, \ldots, a_\ell = a_{\ell-d+1}$) and otherwise 0. We obtain the overlap probability under the background model μ:

$$\gamma_d(a) := \mathbb{P}_\mu(Y_j(a) = 1, Y_{j+d}(a) = 1) = \epsilon_d(a) \cdot \alpha_a \cdot \prod_{\kappa=\ell-d+2}^{\ell} \mu(a_\kappa), \tag{5.2}$$

where $\mu(a_\kappa)$ is the nucleotide probability of a_κ. For correspondence with the TF setting, we denote $\mathbb{P}_\mu(Y_j(a) = 1)$ by α_a instead of $\mu(a)$. We capture the overlap probabilities for each d in the overlap sum $G(a)$ defined as

$$G(a) := \sum_{d=0}^{\ell-1} \gamma_d(a).$$

The variance of the counts of A on a sequence of length n is the sum over all covariances between the hit indicator random variables:

$$\mathbb{V}[N_n(a)] = \sum_{i=1}^{n-\ell+1} \sum_{j=1}^{n-\ell+1} \mathbb{C}\mathrm{ov}[Y_i(a), Y_j(a)].$$

Since the covariance between non-overlapping hits is zero, we only have to consider overlapping hits. The covariance for overlapping hits is given by

$$\begin{aligned} \mathbb{C}\mathrm{ov}[Y_i(a), Y_{i+d}(a)] &= \mathbb{E}[Y_i(a) \cdot Y_{i+d}(a)] - \mathbb{E}[Y_i(a)] \cdot \mathbb{E}[Y_{i+d}(a)] \\ &= \gamma_d(a) - \alpha_a^2, \end{aligned}$$

for $0 \leq d < \ell$. Hence, we can express the asymptotic variance by

$$\lim_{n\to\infty} n^{-1}\mathbb{V}[N_n(a)] = 2G(a) - 2\ell\alpha_a^2 - \alpha_a\left(1 - \alpha_a\right).$$

The first term corresponds to the self-overlap probability of a. The second term contains the product of the expected values of the two indicator random variables. Lastly, we subtract α_a which is added twice with $G(a)$ (overlap at $d = 0$) and add α^2 since we added it 2ℓ instead of $2\ell - 1$ times.

Asymptotic Covariance for Words

Since a TF can be defined by occurrences of a set of words, we introduce the asymptotic covariance before dealing with TFs. Hence, we have to introduce the overlap probabilities for two different words a and a'. We extend the definition of the overlap bit $\epsilon_d(a, a')$ which is 1 if the words allow the overlap ($a_d = a'_1, a_{d+1} = a'_2, \ldots, a_\ell = a'_{\ell-d+1}$) and otherwise 0. Note that we assume without loss of generality, both words have equal length $\ell = |a| = |a'|$. Hence, we obtain the overlap probability

$$\gamma_d(a, a') := \mathbb{P}_\mu(Y_j(a) = 1, Y_{j+d}(a') = 1) = \epsilon_d(a, a') \cdot \alpha_a \cdot \prod_{\kappa=\ell-d+2}^{\ell} \mu(a'_\kappa).$$

Similarily, we capture the overlap probabilities for each d in the overlap sum $G(a, a')$ defined as

$$G(a, a') := \sum_{d=0}^{\ell-1} \gamma_d(a, a').$$

The covariance between the counts of a and a' on a sequence of length n is the sum over all covariances between the hit indicator random variables:

$$\mathbb{C}\mathrm{ov}[N_n(a), N_n(a')] = \sum_{i=1}^{n-\ell+1} \sum_{j=1}^{n-\ell+1} \mathbb{C}\mathrm{ov}[Y_i(a), Y_j(a')].$$

Applying the same reasoning as before, we can express the asymptotic covariance between a and a':

$$\lim_{n\to\infty} n^{-1}\,\mathbb{C}\mathrm{ov}[N_n(a), N_n(a')] = G(a,a') + G(a,a') - 2\ell\alpha_a\alpha_{a'} - \alpha_a\left(\epsilon_0(a,a') - \alpha_{a'}\right).$$

The two first terms correspond to the overlap probability of a followed by a' and a' followed by a. The third term contains the product of the expected values of the two random variables. Lastly, we add the covariance for a and a' occuring at the same position.

Asymptotic Variance for TFs

We introduce the asymptotic variance by ignoring the complementary strand for the beginning. We consider one TF A with length ℓ_A and a set of compatible words $\mathcal{A}$. Obviously, the length of each word is the same within each corresponding set. The probability α_A for a hit of TF A is in terms of its set of compatible words

$$\alpha_A = \mathbb{P}_\mu(Y_j(A) = 1) = \mathbb{P}_\mu\left(\sum_{a\in\mathcal{A}} Y_j(a) = 1\right) = \sum_{a\in\mathcal{A}} \alpha_a.$$

The definition of the self-overlap probabilities for TFs follows the same reasoning. A self-overlap occurs if any word $a \in \mathcal{A}$ overlaps with any word $a' \in \mathcal{A}$. Hence, we can define the self-overlap probability for a TF. Since the events of the indicator random variables Y_j^a for $a \in \mathcal{A}$ are disjoint for fixed position j, we obtain

$$\gamma_d(A) := \mathbb{P}_\mu(Y_j(A) = 1, Y_{j+d}(A) = 1) = \sum_{a\in\mathcal{A}}\sum_{a'\in\mathcal{A}} \gamma_d(a,a').$$

The sums iterate over the set of compatible words. In fact, we can avoid the sum since the event of two overlapping occurrences means that the score S_j and S_{j+d} reach the threshold. Here, we compute the exact value for $\mathbb{P}_\mu(S_{j+k} \geq t_A, S_j(A) \geq t_A)$. We define the set of scores which are assumed to be integers due to score discretization by

$$\mathcal{S}(A) := \left\{ s : \sum_{\kappa=1}^{\ell} \min_{a\in\mathfrak{A}} \Psi_{\kappa,a}^A \leq s \leq \sum_{\kappa=1}^{\ell} \max_{a\in\mathfrak{A}} \Psi_{\kappa,a}^A \right\}. \tag{5.3}$$

Then, the scores greater than or equal to the threshold can be defined by $\mathcal{S}_t(A) := \{s \in \mathcal{S}(A) : s \geq t_A\}$. Using these definitions, we can express $\mathbb{P}_\mu(Y_{j+d}(A) = 1, Y_j(A) = 1)$ in terms of a two-dimensional score distribution

$$\gamma_d(A) = \mathbb{P}_\mu(Y_{j+d}(A) = 1, Y_j(A) = 1) = \sum_{s\in\mathcal{S}_t(A)}\sum_{s'\in\mathcal{S}_t(A)} \mathbb{P}_\mu(S_{j+d}(A) = s', S_j(A) = s). \tag{5.4}$$

The overlapping probabilities $\gamma'_d(A)$ can be computed correspondingly.

Example 5.1. *Figure 5.2 shows the 2-dimensional score distributions for a PFM with strong consensus 'ACACACACAC' (see same figure for sequence logo) for different shifts d.*

The first panel contains the distribution for a shift $d = 0$. In this case, both score random variables are completely determined by each other. Hence, both scores are equal, thus, the 2-dimensional score distribution is in fact 1-dimensional since it only contains strictly positive probabilities on the diagonal. Furthermore, one observes that also on the diagonal many scores are have probability 0 in-between possible scores. Due to the construction of the PFM, its PSM only contains two different scores (for a GC content of 50%). Hence, the scores are very granular. Therefore, the plots are mainly blue since most score values are not possible. In Figure 5.3, the corresponding plots are shown for a similar PFM with slight perturbations such that the scores are less granular.

The next panel for $d = 1$ seems to neither show correlated scores nor any probability weight, which is above both thresholds. However, the scores are negatively correlated similar to the corresponding plot in 5.3. Hence, the PFM does not allow such an overlap. Incrementing the shift by 1, yields a very positively correlated score distribution. In addition, the probability to reach both thresholds is greater than 0. Obviously, observing one occurrence 'ACACACACAC' only requires an additional 'AC' for a second occurrence with a shift $d = 2$.

The shift of $d = 3$ resp. $d = 4$ is similar to $d = 1$ and $d = 2$ and so on because every even shift allows an overlap while odd shifts prevent an overlapping occurrence. With increasing even shifts, the scores are more uncorrelated since more additional nucleotides ('AC's) have to be observed.

The last panel containing a score distribution, shows a shift of $d = 10$ which implies independence. In fact, this distribution looks very similar to the distributions for odd shifts. However, they are not equal. The reason is that the thresholds are set to $\alpha < .05$, thus, only one mismatch is allowed. On the one hand, rigorous inspection of the distribution for $d = 7$ and $d = 9$ show that they slightly differ. The probability for a joint occurrence for $d = 7$ is 0 while for $d = 9$ it is greater than zero. On the other hand, a shift of $d = 10$ is also not equal to the distribution $d = 9$ but has a weakly higher probability to reach both thresholds since the first letter is not determined by the preceding overlapping hit.

□

Considering the scores as state space, the $S_j(A)$s become a first-order Markov chain (Fu and Koutras, 1994) because the score only depends on the score of the previous position. Hence, (5.4) can also be written in terms of its transition matrix to the dth power. For the sake of simplicity, we focus on the two-dimensional score distribution for each d. This distribution can be computed by the two-dimensional convolution of the position specific score distributions. An efficient dynamic programming algorithm is presented in section 5.6.

The sum of the overlap probabilities is given by

$$G(A) := \sum_{d=0}^{\ell_A - 1} \gamma_d(A).$$

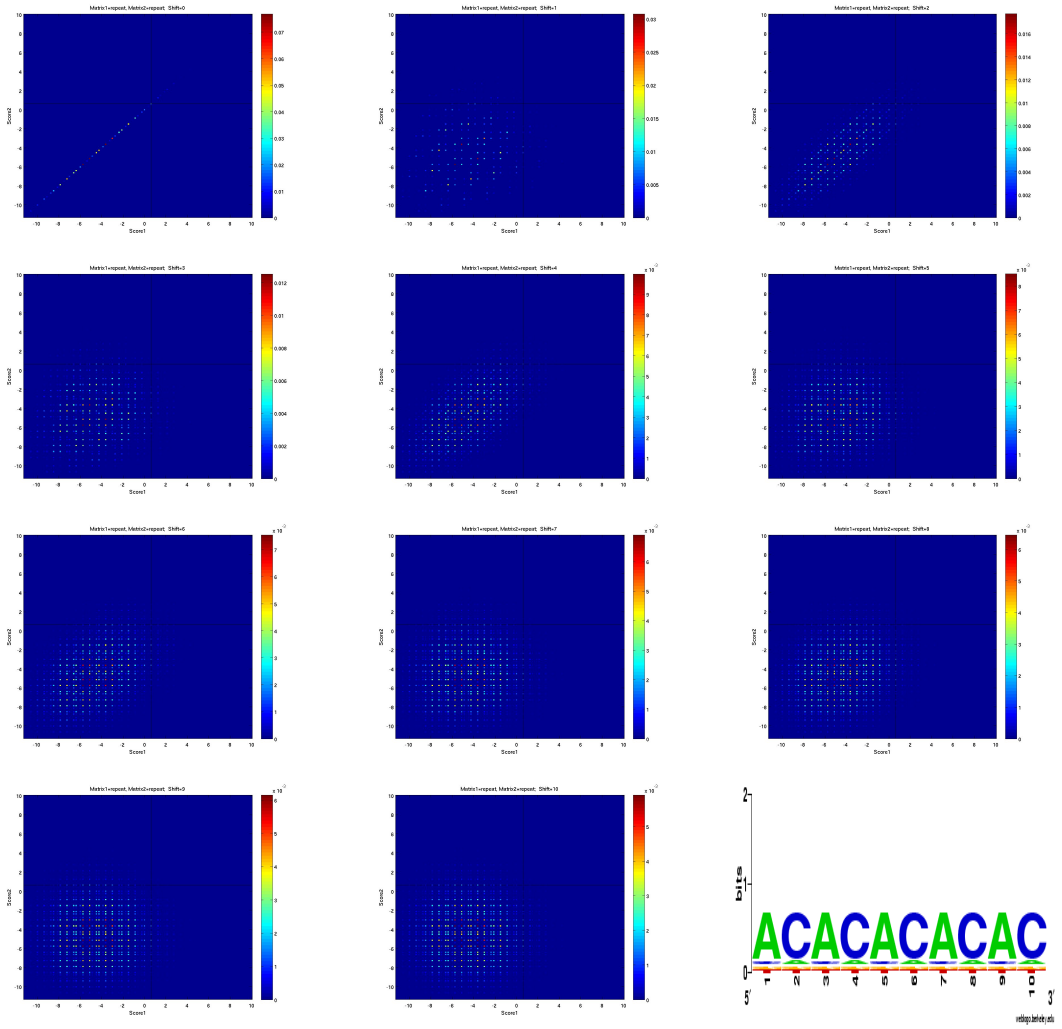

Figure 5.2: Distribution of the 2-dimensional scores for PFM 'repeat' (bottom right panel) for different shifts d. From top to bottom and left to right, the shift is increased by 1 starting with $d = 0$. The black lines denote the thresholds.

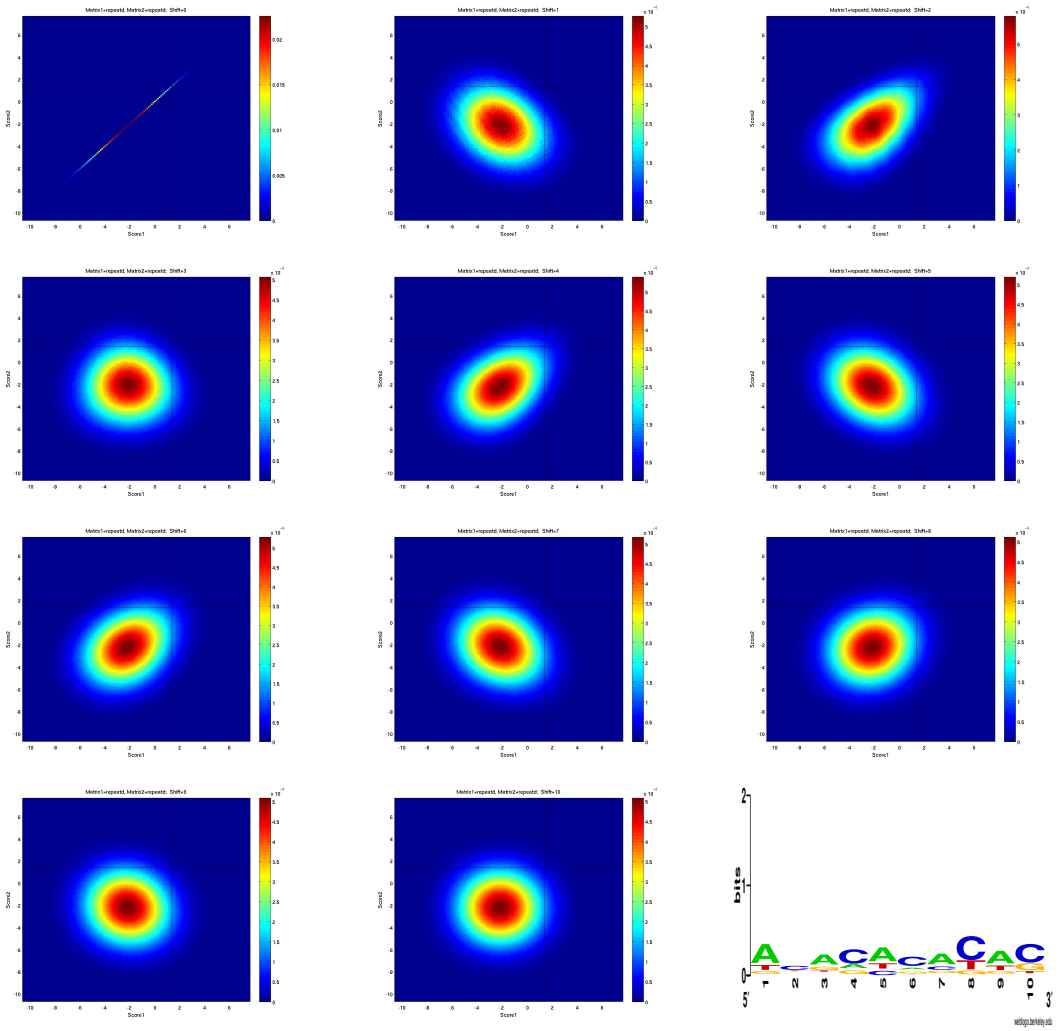

Figure 5.3: Distribution of the 2-dimensional scores for perturbed PFM 'repeat' (bottom right panel) for different shifts d. From top to bottom and left to right, the shift is increased by 1 starting with $d = 0$. The black lines denote the thresholds.

As before, we can express the number of hits for a TF by the occurrences of $a \in \mathcal{A}$. It is the sum of the number of hits for all compatible words: $N_n(A) = \sum_{a \in \mathcal{A}} N_n(a)$. Hence, we can split up the asymptotic variance for the TF into sums of asymptotic covariances of words and then rewrite it with the introduced notation:

$$\begin{aligned}\lim_{n\to\infty} n^{-1}\mathbb{V}[N_m(A)] &= \sum_{a\in\mathcal{A}}\sum_{a'\in\mathcal{A}} \lim_{n\to\infty} n^{-1}\,\mathbb{C}\mathrm{ov}[N_n(a), N_n(a')] \\ &= 2G(A) - 2\ell\alpha_A^2 - \alpha_A\,(1-\alpha_A)\,.\end{aligned}$$

Next, we introduce the asymptotic covariance and subsequently extend to incorporate the complementary strand.

Asymptotic Covariance for TFs

It is straight-forward to apply previous definitions to compute the asymptotic covariance. First, we have to define the overlap probabilities for two different TFs. An overlap occurs between TF A with length ℓ_A and B with length ℓ_B if any of the words in $\mathcal{A}$ overlap with any of the words in the set $\mathcal{B}$ of compatible words of B. Similarily to before, we obtain

$$\gamma_d(A,B) := \mathbb{P}_\mu(Y_j(A) = 1, Y_{j+d}(B) = 1) = \sum_{a\in\mathcal{A}}\sum_{b\in\mathcal{B}} \gamma_d(A,B).$$

The overlap probabilities are obtained by the corresponding two dimensional score distribution. The sum of the overlap probabilities is $G(A,B) := \sum_{d=0}^{\ell_A-1} \gamma_d(A,B)$. Based on these definition, one obtains

$$\begin{aligned}\lim_{n\to\infty} n^{-1}\,\mathbb{C}\mathrm{ov}[N_n(A), N_n(B)] &= \sum_{a\in\mathcal{A}}\sum_{b\in\mathcal{B}} \lim_{n\to\infty} n^{-1}\,\mathbb{C}\mathrm{ov}[N_n(a), N_n(b)] \\ &= G(A,B) + G(B,A) - (\ell_A + \ell_B - 1)\alpha_A\alpha_B - \gamma_0(A,B).\end{aligned}$$

Since we also want to consider reverse complementary overlaps, we have to further extend the asymptotic covariance. Denoting the reverse complementary set of words $\mathcal{A}$ by $\mathcal{A}'$ and the corresponding TF variable by A', the symmetry of the restrictive background model μ yields $\alpha_A = \alpha_{A'}$ and correspondingly for B, and $G(A',B) = G(A,B')$, $G(A',B') = G(A,B)$, $\gamma_0(A',B) = \gamma_0(A,B')$, $\gamma_0(A',B') = \gamma_o(A,B)$. Hence, we obtain the following formula for the asymptotic covariance between two TFs A and B:

$$\begin{aligned}\lim_{n\to\infty} n^{-1}\,\mathbb{C}\mathrm{ov}[N_n(A), N_n(B)] &= \lim_{n\to\infty} n^{-1}\,\mathbb{C}\mathrm{ov}[N_n(A) + N_n(A'), N_n(B) + N_n(B')] \\ &= 2\cdot\big[G(A,B) + G(A',B) + G(B,A) + G(B',A)\big] \\ &\quad -4\cdot(\ell_A + \ell_B - 1)\alpha_A\alpha_B \\ &\quad -2\left(\gamma_0(A,B) + \gamma_0(A',B)\right).\end{aligned}$$

5.4 Count Distribution

Our derivation for the count distribution uses the probability of an occurrence, as well as the self-overlap of the PFM. For example, a PFM with the consensus 'CTAACT' has a higher probability for two hits overlapping in two positions than to obtain two independent hits. Counting the number of occurrences while taking care of the self-overlap has already been discussed for a single word (Guibas and Odlyzko, 1981b; Robin and Schbath, 2001) and a small set of given words (Reinert *et al.*, 2000). As mentioned before, there are two reasons for avoiding the Chen-Stein approach proposed by Reinert *et al.* (2000); Roquain and Schbath (2007): First of all, the enumeration of all compatible words encoded in the PFM is computationally demanding and only possible for small PFMs. Second, the incorporation of the complementary strand can lead to two hits at one position. In terms of word counting, this means that the words in the set of compatible words are not necessarily different contradicting one important assumption for the (compound) Poisson approximation. Therefore, we use the discrete nature of the PFM score to compute the probabilities of overlapping hits (Pape *et al.*, 2006). Based on these probabilities, we use a generalization of the Poisson distribution to model overlaps. We couple a probability vector for the number of hits with a Poisson distribution. This is a compound Poisson distribution or a so called stopped-sum distribution (Johnson *et al.*, 1995). As we have seen by reviewing word counting approaches, this distribution is widely used in this context (see Chapter 3).

For better readability, we avoid the index/parameter A if the TF is obvious from context. As previously mentioned, the probability of an occurrence by chance in a symmetric i.i.d. sequence model is α. Hence, the indicator random variables Y_j and Y'_j have a Bernoulli distribution with $\mathbb{P}_\mu(Y_j = 1) = \mathbb{P}_\mu(Y'_j = 1) = \alpha$ and $\mathbb{P}_\mu(Y_j = 0) = \mathbb{P}_\mu(Y'_j = 0) = 1 - \alpha$. As before, the number of binding sites in a region of length n of a sequence is:

$$N_n = \sum_{j=1}^{n-\ell+1} \left(Y_j + Y'_j\right).$$

In fact, Y_j and Y'_j are defined on an infinite sequence but in practice we are concerned with finite sequences. Hence, the dependencies of Y_j and Y'_j are different at the beginning and the end of the sequence. These boundary effects are negligibly small under the rare hit assumption (Barbour *et al.*, 1992). The rare hit assumption holds because we set α and the corresponding threshold t such that only a very small fraction of all possible words reach the score threshold.

Now, we compute the distribution for the number of occurrences $\mathbb{P}_\mu(N_n)$. Although we know the probability of Y_j and we assume the symmetric i.i.d. sequence model, calculation of p is not straightforward due to dependencies between Y_js and Y'_js. The dependencies are caused by self-overlap of the PFM and by incorporation of the complementary strand both leading to overlapping binding sites.

5.4.1 Computing Probabilities of Clumps

Dealing with overlapping hits requires a refined vocabulary - similar to word counting approaches: A hit and its overlapping hits together can be defined as a clump. The size

of the clump corresponds to the number of contained overlapping hits. Also a single hit without any overlaps is called a clump of size 1. Thus, a clump is a left- and right-maximal set of overlapping hits on both strands.

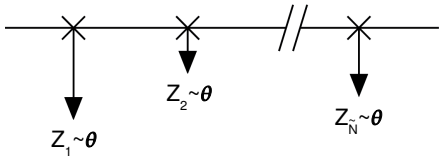

Figure 5.4: The horizontal line symbolizes the sequence. At certain position (marked by a cross), a clump occurs. The size of each clump is modelled by an i.i.d. probability vector $\boldsymbol{\theta}$. The number of clumps $\tilde{N}_n$ is distributed as a Poisson random variable $\mathcal{P}(\vartheta)$.

Now, we incorporate the notion of clumps into our definitions. We assume that the number of clumps $\tilde{N}_n$ is distributed as a Poisson random variable $\mathcal{P}(\vartheta)$ with unknown rate parameter ϑ. The size Z_i of clumps are assumed to be identically and independently distributed by an unknown probability vector $\boldsymbol{\theta}$ (see Figure 5.4) with $\mathbb{P}_\mu(Z_i = k) = \theta_k$. Both assumptions can be justified by the fact that they hold for word counting (Robin, 2002). The number of counts per sequence is given by $N_n = \sum_{i=1}^{\tilde{N}_n} Z_i$. N_n follows a compound Poisson distribution $\mathcal{CP}(\vartheta, \boldsymbol{\theta})$.

In the remaining part of this section, we show how to compute approximations of the unknown parameters of rate ϑ and the probability vector $\boldsymbol{\theta}$. First of all, we reduce the computation of ϑ to the computation of $\boldsymbol{\theta}$. Then, we start with the probability θ_1 of having exactly one hit. Based on θ_1, we recursively compute the remaining parameters. Furthermore, the analysis of the resulting formulation discovers two characteristic values describing the self-overlap of the PFM.

Computing the rate ϑ

The parameter ϑ is the rate of clump occurrences. We cannot use the probability α of a false positive directly to compute it because α is the probability for hits including overlapping hits. In contrast, we need the rate for non-overlapping hits which is equal to the rate of clumps. Using the law of total probability, we obtain:

$$\mathbb{E}[N_n] = \mathbb{E}[\mathbb{E}(N_n \mid \tilde{N}_n)] = \mathbb{E}[\tilde{N}_n \mathbb{E}(Z)] = \mathbb{E}[\tilde{N}_n]\mathbb{E}[Z] = \vartheta \mathbb{E}[Z],$$

where $Z \sim Z_i$. Thus, we can express ϑ in terms of $\boldsymbol{\theta}$: $\mathbb{E}[N_n]$ is the expected number of hits. The probability of a hit by chance is α as we defined the threshold in this way. Hence, the expected number of hits is given by the fact that a hit can occur at each sequence position on each strand, thus

$$\vartheta = \frac{\mathbb{E}[N_n]}{\mathbb{E}[Z]} = \frac{2\alpha\,(n - \ell + 1)}{\sum_{k>0} k\theta_k}. \tag{5.5}$$

Parameters for the probability vector

The probability vector $\boldsymbol{\theta} = (\theta_k)_{k>0}$ contains probabilities for the different sizes of clumps $k > 0$. Here, we show how to compute approximations $(\hat{\theta}_k)_{k>0}$ of these probabilities. At first, we focus on one strand of the sequence only. Subsequently, we extend the approach to deal with the complementary strand as well.

θ_1 corresponds to the event that there is exactly one hit at a certain position j while no overlapping hits occur given the hit at position j. Using the fact that an overlapping hit is a hit within the range of the length ℓ of the PFM, we obtain

$$\theta_1 = \mathbb{P}_\mu(Y_{j-\ell+1} = 0, \ldots, Y_{j-1} = 0, Y_{j+1} = 0, \ldots, Y_{j+\ell-1} = 0 \mid Y_j = 1). \tag{5.6}$$

The conditional probability on the right hand side of (5.6) is hard to compute because the events in the collection $(\{Y_{j+d} = 0\})_{-\ell+1 \leq d \leq \ell-1, d \neq 0}$ are not independent, given $\{Y_j = 1\}$. However, in a first order approximation we pretend that conditional independence holds and compute

$$\hat{\theta}_1 = \prod_{d=-\ell+1, d\neq 0}^{\ell-1} \mathbb{P}_\mu(Y_{j+d} = 0 \mid Y_j = 1) = \prod_{d=-\ell+1, d\neq 0}^{\ell-1} (1 - \mathbb{P}_\mu(Y_{j+d} = 1 \mid Y_j = 1)). \tag{5.7}$$

Due to the symmetric i.i.d. random sequence, we can prove the symmetry $\mathbb{P}_\mu(Y_{j-d} = 1 \mid Y_j = 1) = \mathbb{P}_\mu(Y_{j+d} = 1 \mid Y_j = 1)$ by applying the law of conditional probabilities and substituting $j = j' + d$:

$$\begin{aligned} \mathbb{P}_\mu(Y_{j-d} = 1 \mid Y_j = 1) &= \frac{\mathbb{P}_\mu(Y_{j-d} = 1, Y_j = 1)}{\mathbb{P}_\mu(Y_j = 1)} \\ &= \frac{\mathbb{P}_\mu(Y_{j'+d-d} = 1, Y_{j'+d} = 1)}{\mathbb{P}_\mu(Y_{j'+d} = 1)} \\ &= \mathbb{P}_\mu(Y_{j'+d} = 1 \mid Y_{j'} = 1). \end{aligned} \tag{5.8}$$

Symmetry follows due to the symmetric i.i.d. background model. Thus, we obtain for (5.7)

$$\hat{\theta}_1 = \left(\prod_{d=1}^{\ell-1} [1 - \mathbb{P}_\mu(Y_{j+d} = 1 \mid Y_j = 1)] \right)^2. \tag{5.9}$$

Next, we extend the approach to both strands by continuing to assume conditional independence for all hits and simplify notation by

$$\begin{aligned} \bar{\gamma}_d &:= \mathbb{P}_\mu(Y_{j+d} = 1 \mid Y_j = 1) = \alpha^{-1}\gamma_d, \\ \bar{\gamma}'_d &:= \mathbb{P}_\mu(Y'_{j+d} = 1 \mid Y_j = 1) = \alpha^{-1}\gamma'_d, \end{aligned} \tag{5.10}$$

where Y' refers to the hit random variable on the other strand and γ_d resp. γ'_d are the corresopnding joint probabilities as defined in Eq. (5.2). These terms are the probability of two overlapping hits $\mathbb{P}_\mu(Y_{j+d} = 1, Y_j = 1)$ for d as given above. In Section 5.3.2 about the variance, calculation of the overlap probabilities is outlined. Now, (5.9) becomes

$$\hat{\theta}_1 = (1 - \bar{\gamma}'_0) \prod_{d=1}^{\ell-1} (1 - \bar{\gamma}_d)^2 (1 - \bar{\gamma}'_d)^2. \tag{5.11}$$

Probability of an k-clump with $k > 1$

We recursively compute the probability $\hat{\theta}_k$ to have a clump with exactly k hits for $k > 1$. Without loss of generality, we assume that we count hits starting with $Y_1, Y'_1, Y_2, Y'_2, Y_3$ and so on. Considering a clump of size two, the first overlapping hit at a clump position j can either occur in the interval $d \in [j+1, j+\ell-1]$ on the same strand or in the interval $d \in [j, j+\ell-1]$ on the opposite strand. The idea is to cancel the probability in (5.11) for each position $j+d$ and to replace it with the probability of a hit at this position. We denote these *extension* factors ξ_d for a hit on the $5'-3'$ strand and ξ'_d for a hit on the $3'-5'$ strand. We obtain for a pair of hits for $0 < d < \ell$

$$\begin{aligned}
\mathbb{P}_\mu(Y_{j+d} = 1, Y'_{j+d} = 0, Y'_j = 0, \{Y_{j+\kappa} = 0, Y'_{j+\kappa} = 0\}_{-\ell<\kappa<d+\ell, \kappa\neq 0,d} | Y_j = 1) &\approx \hat{\theta}_1 \cdot \xi_d, \\
\mathbb{P}_\mu(Y'_{j+d} = 1, Y_{j+d} = 0, Y'_j = 0, \{Y_{j+\kappa} = 0, Y'_{j+\kappa} = 0\}_{-\ell<\kappa<d+\ell, \kappa\neq 0,d} | Y_j = 1) &\approx \hat{\theta}_1 \cdot \xi'_d, \\
\mathbb{P}_\mu(Y'_j = 1, \{Y_{j+\kappa} = 0, Y'_{j+\kappa} = 0\}_{-\ell<\kappa<\ell, \kappa\neq 0} | Y_j = 1) &\approx \hat{\theta}_1 \cdot \xi'_0.
\end{aligned}$$

For the definitions of ξ_d, ξ'_d and ξ'_0, it is important to note that one also has to replace the probability at the other strand at position d with the probability $\bar{\gamma}'_0$ for an exact palindromic hit at position d of the old hit except the new hit is on the $3'-5'$ strand. In this case (ξ'_d), an exact palindromic hit is not possible (because we would have counted the hit before). We also replace the probabilities for hits at the subsequent positions given the hit at position j with the probabilities of a hit given the hit at $j+d$. Lastly, one has to extend the positions without a hit to the positions covered by the new hit but not by the former hit. Thus, we obtain the definitions for $0 < d < \ell$

$$\begin{aligned}
\xi_d &:= \frac{\bar{\gamma}_d}{1-\bar{\gamma}_d} \cdot \frac{1-\bar{\gamma}'_0}{1-\bar{\gamma}'_d} \cdot \left(\prod_{\kappa=1}^{\ell-d-1} \frac{1-\bar{\gamma}_\kappa}{1-\bar{\gamma}_{d+\kappa}} \cdot \frac{1-\bar{\gamma}'_\kappa}{1-\bar{\gamma}'_{d+\kappa}} \right) \cdot \left(\prod_{\kappa=\ell-d}^{\ell-1} (1-\bar{\gamma}_\kappa)\left(1-\bar{\gamma}'_\kappa\right) \right), \\
\xi'_d &:= \frac{\bar{\gamma}'_d}{1-\bar{\gamma}'_d} \cdot \left(\prod_{\kappa=1}^{\ell-d-1} \frac{1-\bar{\gamma}_\kappa}{1-\bar{\gamma}_{d+\kappa}} \cdot \frac{1-\bar{\gamma}'_\kappa}{1-\bar{\gamma}'_{d+\kappa}} \right) \cdot \left(\prod_{\kappa=\ell-d}^{\ell-1} (1-\bar{\gamma}_\kappa)\left(1-\bar{\gamma}'_\kappa\right) \right).
\end{aligned}$$

For the other strand, we must not replace the exact palindromic hit because the hit is already on that strand. We have an overlapping hit if any of these encoded events occur. Thus, we can sum up the terms (see also Figure 5.5)

$$\xi := \sum_{d=1}^{\ell-1} \xi_d, \qquad \xi' := \sum_{d=1}^{\ell-1} \xi'_d, \qquad \xi'_0 := \frac{\bar{\gamma}'_0}{1-\bar{\gamma}'_0}. \tag{5.12}$$

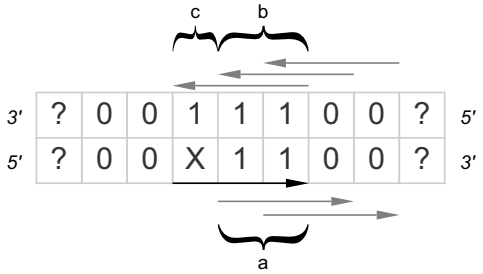

Figure 5.5: X and the black arrow indicate the given hit while 0s indicate where no hit is allowed, 1 denotes the possibility of an overlapping hit (marked by grey arrows), and ? is a hit or no hit. The letters a, b, and c label the three different type of hits: Type (a) is a hit on the same strand, type (b) hits on the complementary strand but not palindromic, and type (c) is the palindromic hit.

We split up the different types of hits into a, b, and c because they differ with respect to which hit can follow (see Figure 5.6). Type (a) encodes a hit on the $5' - 3'$ strand. The hit can be followed by a hit on the same strand (a), a hit on the complementary strand (b), or an exact palindromic hit (c). Type (b) is a hit on the $3' - 5'$ strand excluding an exact palindromic hit. After it, types (a) and (b) can follow. An exact palindromic hit is not possible as the hit itself is on the $3' - 5'$ strand. Type (c) stands for the exact palindromic hit. Everything but another exact palindromic hit can follow.

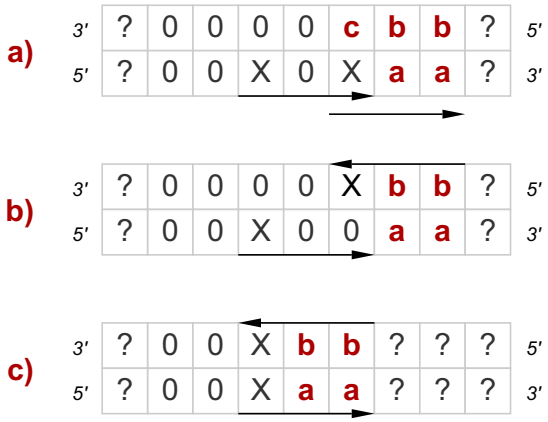

Figure 5.6: The figure shows the three different types of hits (a), (b), and (c) in a situation of two previous hits. Furthermore, for each type of hit the possible subsequent type of hit is indicated by the corresponding identifier (a, b or c). While (a) and (b) can occur after each type of hit, (c) can only occur after type (a).

Thus, an additional hit of type (a) can be preceded by a hit of type (a), (b), or (c). For (b) the same logic applies. In contrast, a palindromic hit (c) can only occur after a hit of type (a). Without these considerations, we would allow more than two hits at the same position. This gives us a linear system of recurrences to compute an approximation of $\boldsymbol{\theta}$.

Again, assuming that conditional independence holds, we have $\hat{\theta}_{k+1} := \hat{\theta}_1(a_k + b_k + c_k)$, where

$$a_1 := \xi, \qquad a_{k+1} := (a_k + b_k + c_k)\xi, \tag{5.13a}$$
$$b_1 := \xi', \qquad b_{k+1} := (a_k + b_k + c_k)\xi', \tag{5.13b}$$
$$c_1 := \xi'_0, \qquad c_{k+1} := a_k\xi'_0. \tag{5.13c}$$

5.4.2 Closed Formula

Although the above formulae suffice to compute an approximation for $\boldsymbol{\theta}$, the reformulation as a closed formula and its further analysis reveals interesting insights. At the end, we obtain two characteristic values which describe the self-overlap of the PFM.

We can write the recursive formulae (5.13a) to (5.13c) by matrix notation for $k > 0$

$$\begin{pmatrix} a_{k+1} \\ b_{k+1} \\ c_{k+1} \end{pmatrix} = \begin{pmatrix} \xi & \xi & \xi \\ \xi' & \xi' & \xi' \\ \xi'_0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} a_k \\ b_k \\ c_k \end{pmatrix} =: A \cdot \begin{pmatrix} a_k \\ b_k \\ c_k \end{pmatrix}.$$

Thus, we get the closed formula for $k \geq 0$ with $\boldsymbol{\xi} = (\xi, \xi', \xi'_0)^T$

$$\begin{pmatrix} a_{k+1} \\ b_{k+1} \\ c_{k+1} \end{pmatrix} = A^k \, \boldsymbol{\xi}.$$

Furthermore, we obtain $\hat{\theta}_{k+1}$ using the recurrence formula for $k > 0$

$$\hat{\theta}_{k+1} = (1, 1, 1) \cdot A^k \cdot \boldsymbol{\xi} \cdot \hat{\theta}_1.$$

We decompose $A = B^{-1} \, \Lambda \, B$ where B contains the eigenvectors of A and the diagonal matrix Λ the corresponding eigenvalues $\lambda_1, \lambda_2, \lambda_3$ given by

$$\lambda_{1,2} = \frac{\xi + \xi'}{2} \pm \frac{1}{2}\sqrt{w}, \quad \lambda_3 = 0,$$

with $w = (\xi + \xi')^2 + 4\xi \, \xi'_0$. Hence, we can denote $\hat{\theta}_{k+1}$ in terms of the eigenvalues

$$\hat{\theta}_{k+1} = (1 \;\; 1 \;\; 1) \, B^{-1} \, \Lambda^k \, B \, \boldsymbol{\xi} \, \hat{\theta}_1 = \left(u\lambda_1^k + v\lambda_2^k \right) \hat{\theta}_1 \tag{5.14}$$

where u and v are computed by solving the linear system

$$\begin{pmatrix} 1 & 1 \\ \lambda_1 & \lambda_2 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} \hat{\theta}_1 = \begin{pmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \end{pmatrix},$$

whereby $\hat{\theta}_2 = (\xi + \xi' + \xi'_0)\hat{\theta}_1$ using the recursive formula. Finally, we obtain the solution

$$u, v = \frac{w \pm (\xi + \xi' + 2\xi'_0)\sqrt{w}}{2w}.$$

In addition to the benefits of a closed formula, the expression in (5.14) shows that the asymptotics of the clump size only depend on the first two eigenvalues λ_1, λ_2. Obviously, the following inequalities hold: $\lambda_2 \leq 0$ and $\lambda_1 > -\lambda_2$. Hence, the series $\hat{\theta}_k$ converges to zero, $\lim_{k\to\infty} \hat{\theta}_k = 0$, if $\lambda_1 < 1$. This condition holds if most of the words of length ℓ do not exceed the threshold which is in practise always true. In most cases, we can also assume $\lambda_1 < 1$ since for $\lambda_1 > 1$ we can transform the PSM and the threshold to its complement which yields $\lambda_1 < 1$ and correspondingly use the complementary statistics.

Two Characteristic Values for PFMs

The eigenvalues can be used as descriptive values for the PFM: We call a PFM a palindrome if it allows two hits at the same position on both strands. Considering a PFM which is not a palindrome, we have $\bar{\gamma}'_0 > 0$, thus, $\xi'_0 = 0$. Hence, matrix A has only rank 1 and we only obtain one non-zero eigenvalue λ_1 and $u = 1$. Therefore, $\hat{\theta}_{k+1} = \lambda_1^k \hat{\theta}_1$ decreases exponentially. Higher values of λ_1 decelerate convergence. Since $\hat{\theta}_1$ is the probability of a clump of size one, which means that no overlap occurs, and λ_1 corresponds to the probability of an overlap, obviously $\hat{\theta}_1 = 1 - \lambda_1$ holds. Hence, the clump size has a shifted geometric distribution. This case is similar to the compound Poisson model for one word considering only one strand described by Robin (2002). Discarding the complementary strand always leads to $\lambda_2 = 0$ and then both models are equivalent.

A palindromic PFM has $\lambda_2 < 0$ since $\xi'_0 > 0$. If ξ'_0 dominates the eigenvalues we obtain $\lambda_2 \approx -\lambda_1$. In addition, it follows that $v \approx -u$. Thus, $\hat{\theta}_{k+1} \approx u[1+(-1)^k]\lambda_1^k \hat{\theta}_1$. This leads to $\hat{\theta}_k \approx 0$ for odd k. It indicates that the probability of an odd clump size is approximately equal to zero. As we assumed a palindromic PFM with a very high probability of a palindromic hit, one almost always detects a hit on both or neither strands. In summary, λ_1 describes the speed of convergence of $\hat{\theta}_k$ to zero and $-\lambda_2$ correlates with the tendency of palindromic hits.

5.4.3 The p-value for the Number of Hits

Now, we can compute the approximations of the probability vector $\boldsymbol{\theta}$ and the rate parameter ϑ using (5.5). Using the approximations for the parameters, we can compute the distribution for the number of hits $x \geq 0$. Since the number of hits X is distributed as $\mathcal{CP}(\vartheta, \boldsymbol{\theta})$ we can apply formulae for the compound Poisson distribution (Kemp, 1967):

$$\begin{aligned} \mathbb{P}_\mu(N_n = 0) &= \exp(-\vartheta), \\ \mathbb{P}_\mu(N_n = x+1) &= \frac{\vartheta}{x+1} \sum_{x'=0}^{x} (x+1-x')\theta_{x+1-x'} \mathbb{P}_\mu(N_n = x'). \end{aligned}$$

The p-value for the occurrence of $x \geq 0$ hits is computed by:

$$p = \mathbb{P}_\mu(N_n \geq x) = 1 - \sum_{x'=0}^{x-1} \mathbb{P}_\mu(N_n = x').$$

5.4.4 The p-value for the Number of Clumps

The underlying Poisson process for the count statistic is given by $\tilde{N}_n \sim \mathcal{P}(\vartheta)$. Since $\tilde{N}_n$ is the number of clumps, one can use $\mathcal{P}(\vartheta)$ to compute p-values p' for clumps as the count entity. In this case, we only need to compute the rate ϑ. Equation (5.5) can be approximated by

$$\hat{\vartheta} = \frac{2\alpha\,(n - \ell + 1)}{\sum_{k>0} k\hat{\theta}_k}.$$

Next, we show how to compute $\tilde{\vartheta}$ efficiently by substituting the sum under the assumption $-1 < \lambda_2 \leq 0 \leq \lambda_1 < 1$. Using the expression in (5.14), we obtain

$$\begin{aligned}
\sum_{k>0} k\hat{\theta}_k &= \sum_{k>0} k(u\lambda_1^{k-1} + v\lambda_2^{k-1}) \cdot \hat{\theta}_1 \\
&= \left(\frac{u}{\lambda_1} \sum_{k>0} k\lambda_1^k + \frac{v}{\lambda_2} \sum_{k>0} k\lambda_2^k \right) \cdot \hat{\theta}_1 \\
&= \left(\frac{u}{(1-\lambda_1)^2} + \frac{v}{(1-\lambda_2)^2} \right) \cdot \hat{\theta}_1.
\end{aligned}$$

Again, this equation has an interpretation in terms of the self-overlap of the PFM. In case of a non-palindromic PFM, the second term is equal to zero since $\lambda_2 = 0$ and, therefore, $v = 0$. From $\lambda_2 = 0$, it also follows that $w = (\xi + \xi')^2$, hence, $u = 1$. Since a non self-overlapping PFM has λ_1 near to zero, the above equation is equal to 1. Thus, the expected value of the clump size is equal to 1 and the rate for clumps is $\hat{\vartheta} = 2\alpha(n - \ell + 1)$. Furthermore, $\hat{\theta}_1 = 1$ contains all the weights of the probability vector. Then, $\mathcal{CP}(\hat{\vartheta}, (1, 0, \ldots)) \sim \mathcal{P}(\hat{\vartheta})$. Hence, applying the derived statistic to a non self-overlapping PFM leads to a Poisson process with rate $\hat{\vartheta}$ for the number of occurrences.

5.5 Generating Function Formalism

An important idea of the previous approach is the summing of word probabilities into TF occurrence probabilities. Using this idea, we can easily derive generating functions for the number of TF occurrences. The formalism is based on the introduction to generating functions given in Chapter 4 following the exposition in Rahmann (2000). The advantage of such a formalism is that this enables us to use all kind of statistics derived for word counting based on generating functions (e.g., see Nicodeme *et al.*, 1999; Stefanov *et al.*,

2007). However, we have to restrict the sequence model to one strand since two occurrences at one position are difficult to deal with. Furthermore, we assume that the word occurrence probabilities are equal.

However, if the assumption of equi-probable nucleotide distribution is not fulfilled the formalism still yields a reasonable approximation. For non-uniform background distribution, the scores in the PSM are strongly influenced by the background distribution since the PSM contains the likelihood ratios. Hence, positions with low background nucleotide frequencies only have high scores if they have high support from the PFM. In this case, there is a strong consensus at this position. Therefore, the set of compatible words usually contains words with similar occurrence probabilities $\mu(a)$ justifying our assumption.

5.5.1 Waiting Time and Stopping Probabilities

We define an occurrence of TF A with length ℓ and set of compatible words $\mathcal{A}$ by the last position of the hit

$$\underline{Y}_i(A) := \sum_{a \in \mathcal{A}} \underline{Y}_i(a),$$

where $\underline{Y}_i(a)$ is equal to 1 if i is the last position of an occurrence of a and otherwise 0. Note that here we use the fact that the events $\{\underline{Y}_i(a)\}_{a \in \mathcal{A}}$ are disjoint. Thus, $\mathcal{A}$ is not allowed to be a multi-set. Therefore, we have to remove the complementary strand from the analysis.

The waiting time $\underline{T}_1(A)$ until the first occurrence of A is defined based on the word waiting times $\underline{T}_1(a)$

$$\underline{T}_1(A) := \min_{a \in \mathcal{A}} \underline{T}_1(a),$$

where $\underline{T}_1(a)$ is the smallest index i for which $\underline{Y}_i(a) = 1$. We can compute the correponding probabilities by introducing stopping probabilities for $a \in \mathcal{A}$

$$w_n^{(a)} := \mathbb{P}_\mu(\underline{T}_1(A) = n, \underline{Y}_n(a) = 1) = \mathbb{P}_\mu(\underline{T}_1(A) = n, \underline{T}_1(a) = n).$$

Again, we use the disjoint property of $\{\underline{Y}_i(a)\}_{a \in \mathcal{A}}$. For $n < \ell$, the stopping probabilities are 0 since the occurrence cannot start before the sequence. Obviously, one obtains $t_n := \mathbb{P}_\mu(\underline{T}_1(A) = n) = \sum_{a \in \mathcal{A}} w_n^{(a)}$ due to the law of total probability. However, we will see that it is difficult to compute the stopping probabilities without the assumption of an equi-probable nucleotide distribution.

Return Probabilities

First, we need to define return probabilities

$$r_n^{(a,a')} := \mathbb{P}_\mu(\underline{Y}_{i+n}(a') = 1 | \underline{T}_1(a) = i) = \mathbb{P}_\mu(\underline{Y}_{i+n}(a') = 1 | \underline{Y}_i(a) = 1).$$

They are easy to compute since we know the overlap probabilities

$$r_n^{(a,a')} := \begin{cases} 1 & \text{if } n = 0 \text{ and } a = a', \\ 0 & \text{if } n = 0 \text{ and } a \neq a', \\ \epsilon_n(a,a') \prod_{\kappa=\ell-n+1}^{\ell} \mu(a'_\kappa) & \text{if } 1 \leq n < \ell, \\ \mu(a') & \text{otherwise,} \end{cases}$$

where the overlap bit $\epsilon_n(a,a')$ is 1 if $a_n = a'_1, \ldots, a_\ell = a'_{\ell-n+1}$ and otherwise 0.

Stopping Probabilities for Set of Words

Now, we can express the stopping probabilities in terms of return probabilities. We decompose the event of an occurrence by

$$\{\underline{Y}_n(a) = 1\} \equiv \bigcup_{i=0}^{n} \left\{ \bigcup_{a' \in \mathcal{A}} \{\underline{Y}_n(a) = 1, \underline{T}_1(A) = i, \underline{T}_1(a') = i\} \right\}.$$

Thus, the event of an occurrence of a at n is a partition of the events that at position i is the first occurrence of any word in $\mathcal{A}$. Of course, i can also be n which means that a is the first occurrence. Conditioning on the waiting times directly yields for $a \in \mathcal{A}$

$$\mathbb{P}_\mu(\underline{Y}_n(a) = 1) = \sum_{i=0}^{n} \sum_{a' \in \mathcal{A}} r_{n-i}^{(a',a)} w_i^{(a')}. \tag{5.15}$$

Hence, we obtain an $|\mathcal{A}|$ dimensional system of linear equations. In general, one can show that the system has a unique solution (Rahmann, 2000) if $\mathcal{A}$ is a set (in contrast to a multiset). However, the problem for TF is the large size of $\mathcal{A}$. Therefore, we avoid solving this linear system by using our assumption.

Stopping Probabilities under equi-probable Sequence Model

In this case, the stopping probabilities $w_n^{(a)}$ are equal. For all $a \in \mathcal{A}$ and $n \geq 0$, this leads to

$$w_n := w_n^{(a)}.$$

Under this assumption, above system of linear equations derived from Eq. (5.15) becomes after summing over all $a \in \mathcal{A}$

$$\sum_{a \in \mathcal{A}} \mathbb{P}_\mu(\underline{Y}_n(a) = 1) = \sum_{i=0}^{n} w_i \sum_{a \in \mathcal{A}} \sum_{a' \in \mathcal{A}} r_{n-i}^{(a',a)}. \tag{5.16}$$

The left hand side is obviously the probability of an occurrence of A. The right hand side sums the return probabilities over all possible word pairs. For the sums over the return probabilities, one obtains after changing the index $n - i$ to n for convenience

$$\sum_{a \in \mathcal{A}} \sum_{a' \in \mathcal{A}} r_n^{(a',a)} = \sum_{a \in \mathcal{A}} \sum_{a' \in \mathcal{A}} \mathbb{P}_\mu(\underline{Y}_{i+n}(a) = 1 | \underline{Y}_i(a')) = \sum_{a' \in \mathcal{A}} \frac{\sum_{a \in \mathcal{A}} \mathbb{P}_\mu(\underline{Y}_{i+n}(a) = 1, \underline{Y}_i(a'))}{\mathbb{P}_\mu(\underline{Y}_i(a'))}.$$

With above assumption that the word occurrence probabilities are equal (hence, $\mathbb{P}_\mu(\underline{Y}_i(a')) = \mu(A)/|\mathcal{A}|$), one obtains

$$\sum_{a \in \mathcal{A}} \sum_{a' \in \mathcal{A}} r_n^{(a',a)} = |\mathcal{A}| \mu(A)^{-1} \sum_{a' \in \mathcal{A}} \sum_{a \in \mathcal{A}} \mathbb{P}_\mu(\underline{Y}_{i+n}(a) = 1, \underline{Y}_i(a')).$$

The right hand term is the joint occurrence probabilities summed over all word pairs. Hence, we can substitute this by $\gamma_n(A)$ for $0 \leq n < \ell$ and otherwise by $\mu(A)^2$. Hence, the whole term becomes $|\mathcal{A}| \bar{\gamma}_n(A)$ resp. $|\mathcal{A}| \mu(A)$. Using the definition

$$r_n := \begin{cases} \bar{\gamma}_n(A) & \text{if } 0 \leq n < \ell \\ \mu(A) & \text{otherwise} \end{cases},$$

and the fact $t_n = \sum_{a \in \mathcal{A}} w_n = |\mathcal{A}| \bar{w}_n$ yields for Eq. (5.16)

$$\mu(A) = \sum_{i=0}^{n} r_{n-i} t_i.$$

Note that this equation neither contains the sum over the compatible words since the joint/conditional and occurrence probabilities can be computed by the score convolution. Furthermore, the number of compatible words $|\mathcal{A}|$ is cancelled. The equation is similar to Eq. (4.4) in Chapter 4 for word counting. Hence, with above assumption one can derive similar fundamental equations to develop further statistics. This equation can be written as generating function by

$$\frac{\mu(A)z^{\ell}}{1-z} = r(z)t(z),$$

where $r(z) = \sum_{n\geq 0} r_n z^n$ and $t(z) = \sum_{n\geq 0} t_n z^n$. Dividing by $r(z)$ directly yields the equation for the waiting time.

5.5.2 Number of Counts

With the same assumption, one obtains formulae for the waiting time till the kth occurrence, as well as for the number of counts. However, nothing changes except that the return probabilities contain the conditional occurrence probabilities for A under our assumption. For the probability v_n of the inter-arrival time between two successive occurrences, one obtains the corresponding generating function $v(z) = 1 - [r(z)]^{-1}$. The probability $t_n^{(k)}$ that the kth occurrence occurs at the nth position, is given by

$$t^{(k)}(z) := \sum_{n\geq 0} t_n^{(k)} z^n = t(z)\,[v(z)]^{k-1}\,.$$

Finally, the probability $f_n^{(k)}$ for k occurrences in a sequence of length n can be computed by

$$f^{(k)}(z) := \sum_{n\geq 0} f_n^{(k)} z^n = t^{(k)}(z)\frac{1-v(z)}{1-z}.$$

Hence, based on the score convolutions and under above assumption one can compute the corresponding generating functions as easy as for single words.

5.6 An Efficient Algorithm for Computing Overlap Probabilities

In (5.4), we have to compute the joint event of two scores greater than or equal to the threshold. Given a position j and a shift d, the two scores induce a two dimensional distribution. The first component is the score s of the PSM starting at position j. The second component is the score s' of the PSM beginning at position $j + d$. As an example, consider a PSM which only accepts 'CC'. In the case of a shift $d = 1$, the score s' can only exceed the threshold if s is above the threshold. Thus, both scores are not independent for $0 \leq d < \ell$. Since scores are the sum of the position specific scores $\Psi_{\kappa,\cdot}$ we can decompose the score into each pair of positions $j + \kappa$ and $j + d + \kappa$ which point to the same sequence position, and, thus, to the same nucleotide. Then, pairs of scores are independent. Hence, we can use a dynamic programming algorithm.

The dynamic programming approach is often used for the computation of the one dimensional score distribution (Staden, 1989; Claverie and Audic, 1996; Wu *et al.*, 2000; Rahmann, 2003; Rahmann *et al.*, 2003; Beckstette *et al.*, 2006; Touzet and Varré, 2007). We extend this approach to two dimensions similar to Liefooghe *et al.* (2006). Let $Q_i^{(d)}(s, s')$ denote

the probability for a score s at the first $i+1$ positions of the PSM and a score s' at the first $i+1-d$ positions of the PSM shifted by d positions. We compute this value by summing over the probabilities of the last step $i-1$ which yield a score s and s' after observing any nucleotide with its respective score at step i. With $\Psi_{\kappa,\cdot} := 0$ for $\kappa < 0$ or $\kappa \geq \ell$, we obtain for $0 \leq k < \ell$ and $0 < i \leq \ell + d$

$$\begin{aligned} Q_0^{(d)}(s,s') &:= \begin{cases} \text{undefined} & \text{if} \quad s \neq 0 \quad \text{or} \quad s' \neq 0, \\ 1 & \text{else,} \end{cases} \\ Q_i^{(d)}(s,s') &:= \sum_{a \in \mathfrak{A}} Q_{i-1}^{(d)}(s - \Psi_{i,a}, s' - \Psi_{i-d,a}) \cdot \mu(a). \end{aligned}$$

After the last step, $Q_{\ell+d}^{(d)}(s,s')$ contains the probability to observe score s starting at position j and score s' starting at position $j+d$. Therefore, $\mathbb{P}_\mu(S_{j+d} = s', S_j = s) = Q_{\ell+d}^{(d)}(s,s')$ and, hence, we can solve (5.4).

5.6.1 Speed Improvement

The practical running time of the algorithm can be improved significantly by some modifications. The last d steps do not modifiy the scores starting at position j since $\Psi_{\kappa,\cdot} = 0$ for $\kappa \geq \ell$. Hence, instead of the d two-dimensional convolutions, we can obtain $Q_{\ell+d-1}^{(d)}$ in one step by using the one dimensional convolution of the last d positions (see Figure 5.7). Since (5.4) sums over all scores s starting at position j, we can do the summation before adding the remaining scores:

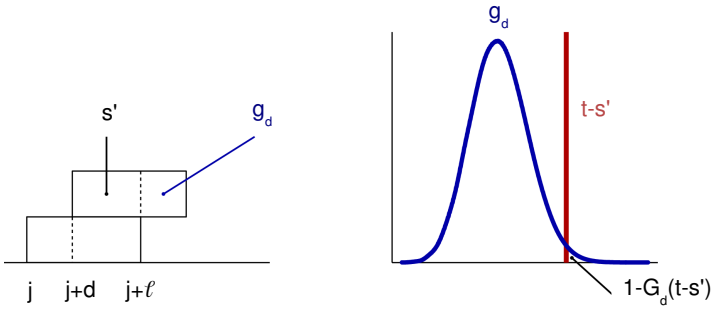

Figure 5.7: The left part of the figure shows the sequence with two overlapping hits at position j and $j+d$. The score of the overlapping part of the second hit is given by s'. The score of the non-overlapping part is a random variable whose distribution is the convolution of the position-specific scores of the remaining positions of the PFM g_d. The right part of the figure visualizes this distribution and the probability of the second hit $1 - G_d(t-s')$.

$$R^{(d)}(s') = \sum_{s \in \mathcal{S}_t} Q_\ell^{(d)}(s,s').$$

Let f_κ denote the position specific score distribution at position κ of the PSM and g_d denotes the score distribution for the non-overlapping part for a shift d:

$$g_d := f_{\ell-d+1} * \ldots * f_\ell.$$

Considering the recursion $g_{d+1} = g_d * f_{\ell-d}$, we can use a dynamic programming approach for computing the convolution. Denoting the cumulative distribution for g_d with G_d, we can rewrite (5.4) by:

$$\gamma_d = \mathbb{P}_\mu(Y_{j+d} = 1, Y_j = 1) = \sum_{s' \in \mathcal{S}_t} \left[1 - G_d(t - s')\right] R^{(d)}(s').$$

We can further improve the speed by removing scores in each step of the calculation of Q which are too small to reach the threshold (see Beckstette *et al.* (2006) for the one-dimensional case). Hence, we define intermediate thresholds t_i by

$$t_i := t - \sum_{\kappa=i+1}^{\ell-1} \max_{a \in \mathfrak{A}} \Psi_{\kappa,a}. \tag{5.17}$$

In each step i we can remove scores $s < t_i$ and $s' < t_{i-d}$. In addition, one can merge scores which will exceed the threshold t for sure. The corresponding intermediate thresholds t'_i are defined analogously to (5.17) by substituting min with max. Then, in each step i, scores $s \geq t'_i$ and $s' \geq t'_{i-d}$ can be merged.

We can further speed up the algorithm by enhancing the effect of these improvements (see Beckstette *et al.* (2006) for a similar idea). Processing positions of the PFM with high information content first, discards many scores which can't exceed the threshold at all in the first steps. In addition, indefinite positions (processed at the end) often do not change the score significantly such that either the score has already been discarded or the score surely exceeds the threshold. This reduces the size of Q significantly.

In summary, the algorithm takes advantage from both a high and a low threshold t: On the one hand, the higher the threshold, the more scores can be removed in the beginning steps because scores will not be able to exceed the threshold at all. On the other hand, a low threshold yields many scores which surely exceed the threshold. As those scores can be merged, the number of different scores (size of Q) stays low.

5.6.2 Extension to Pairs of TFs

It is straight-forward to apply the algorithm to two different PFMs. In the pre-processing, the PSMs for TF A and B are extended by zeros in the same way. To compute $\mathbb{P}_\mu(S_j(A) \geq t_A, S_{j+d}(B) \geq t_B)$The recursion of $Q_i^{(d)}(s, s')$ only slightly changes to

$$Q_i^{(d)}(s, s') := \sum_{a \in \mathfrak{A}} Q_{i-1}^{(d)}(s - \Psi_{i,a}^A, s' - \Psi_{i-d,a}^B) \cdot \mu(a).$$

The scores can be computed by $\mathbb{P}_\mu(S_j(A) = s, S_{j+d}(B) = s') = Q_i^{(d)}(s, s')$ with $i = \max(d + \ell_B, \ell_A)$. Similar speed improvements can be applied. Furthermore, computation of overlap

probabilities with occurrences on the complementary strand are retrieved by substitutung A by its reverse complementary PSM $\Psi^{A'}$.

5.6.3 Time Complexity

The complexity of the algorithm for the computation of Q depends on the length of the PFM ℓ, the size of the set $\mathcal{S}$ of all scores, and the alphabet size $|\mathfrak{A}|$: $O(\ell^2|\mathcal{S}|^2|\mathfrak{A}|)$. The length of the PFM ℓ and the alphabet size $|\Sigma|$ are primitives and, therefore, cannot be reduced any further. In contrast, $|\mathcal{S}|$ is a constructed set, hence, we have to analyze its complexity. It is important to note that the size of $\mathcal{S}$ is independent of the threshold and, therefore, of the number of compatible words. Furthermore, $|\mathcal{S}|$ does not grow exponentially with increasing length of the PFM because the scores of a new column are only added to the overall scores. This only increases the size of $\mathcal{S}$ linearly with increasing PFM length.

Chapter 6

cis-regulatory modules (CRMs)

6.1 Introduction

Interaction of nearby TFs initiate or inhibit transcription of a gene (Fickett, 1996; Arnone and Davidson, 1997; Yuh *et al.*, 1998). The set of TFBS upstream of a gene is called a *cis* regulatory module (CRM, Berman *et al.* (2002)). A CRM is a sequence region with dense clusters of TFBS as demonstrated experimentally (Clyde *et al.*, 2003; Harbison *et al.*, 2004) and computationally (Wagner, 1999; Lifanov *et al.*, 2003). In general, they can be divided into CRMs bound by the same TF - homotypic CRMs - and heterotypic CRMs bound by different TFs (Wagner, 1997; Brown *et al.*, 2002). Homotypic CRMs are often detected using a scoring function (Wagner, 1999; Papatsenko *et al.*, 2002), e.g. FLYENHANCER (Markstein *et al.*, 2002), SCORE (Rebeiz *et al.*, 2002), and CLUSTER (Lifanov *et al.*, 2003). Common programs to find heterotypic CRMs are ClusterDraw (Papatsenko, 2007), ModuleSearcher (Aerts *et al.*, 2003), MCAST (Bailey and Noble, 2003), eCISANALYST (Berman *et al.*, 2004), Cister (Frith *et al.*, 2001), Cluster-Buster (Frith *et al.*, 2003), and TargetExplorer (Sosinsky *et al.*, 2003).

CRMs can be detected using *ab initio* discovery of TF motifs (e.g. (Zhou and Wong, 2004; Gupta and Liu, 2005)) or based on known TF motifs. We assume that the TF motifs are known. Many approaches have been proposed integrating data of different kind for improving CRM prediction (Pilpel *et al.*, 2001; Manke *et al.*, 2005; Yu *et al.*, 2006). Since the main characteristic of CRMs is their high local density of TFBSs, one essential data source is always the DNA sequence annotated with TFBSs. Here, we focus on DNA motifs represented by PFMs. Other approaches compute the co-operative binding energy of multiple sites of TFs (GuhaThakurta and Stormo, 2001; Frith *et al.*, 2004) using thermodynamical models.

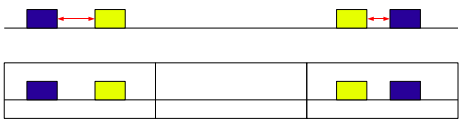

Figure 6.1: Two different approaches to detect CRMs: Upper panel illustrates approaches which are based on short distances between TFBS. Lower panel visualizes detection of CRM considering occurrences in windows.

Based on the PFM representation, GuhaThakurta (2006) classifies the approaches to find CRMs into hidden Markov models (Crowley *et al.*, 1997; Frith *et al.*, 2001) and occurrence-based approaches, which we also focus on. We further divide them into two approaches

- relying on small distances between TFBSs, (Wasserman and Fickett, 1998; Klingenhoff *et al.*, 1999; Wagner, 1999)
- based on co-occurrences of TFBSs in a small window (Berman *et al.*, 2002; Hannenhalli and Levy, 2002; Frith *et al.*, 2002; Rateitschak *et al.*, 2004; Bleser *et al.*, 2007).

The method to compute statistical significance is a difficult problem (Krivan, 2004) and can be split up into

- assuming position independence (Wasserman and Fickett, 1998; Wagner, 1999; Frith *et al.*, 2002),
- employing randomizations (Hannenhalli and Levy, 2002; Bleser *et al.*, 2007) or
- exact calculation (Boeva *et al.*, 2007).

The position independence of binding site occurrences is strongly violated for (self-)similar TFBS (Wagner, 1999; Pape *et al.*, 2008c). The significance calculation based on randomization also encouters problems for similar PFMs, hence, they are removed from the analysis (Hannenhalli and Levy, 2002). In addition, incorporating the complementary strand, introduces further dependencies and worsen the results. The exact calculation (Boeva *et al.*, 2007) based on a Aho-Corasick automaton (Aho and Corasick, 1975) has high computational complexity such that solutions for longer PFMs are hard to compute. To be more precise, the asymptotical complexity is in the simplest case (each PFM occurs at least once, i.i.d. background model) $O(n|\mathfrak{A}|\ell|\mathcal{A}|)$ where n denotes the sequence length, $|\mathfrak{A}|$ is the alphabet size, ℓ the length (of the longest PFM) and $|\mathcal{A}|$ the number of compatible words. As shown in Section 2.4.4, using a standard threshold selection method for the PFMs, the set of compatible words has exponential size. Furthermore, the approach does not use the complementary strand.

We propose a fast and accurate approximation for the significance calculation of CRMs circumventing the position independence assumption, incorporating similarity between PFMs, and including the complementary strand. We define a CRM to be a sequence region (window) of defined length where all TFs of a given set have at least one occurrence. This is called the co-occurrence event. To get statistically significant CRMs, the length of the window has to be small such that the co-occurrence event is unlikely to happen by chance. Firstly, we compute the probability of a CRM which is the probability of the co-occurrence event in a random sequence given a window length. Considering the overlap probabilities between the occurrences of the TFs, we capture the (self-)similarities of the PFMs and most of the dependencies introduced by the complementary strand. We call this a first-order approximation since we ignore dependencies between three or more positions. Based on this calculation, we show how to compute the length of the window for a specific set of TFs by defining the probability of the co-occurrence event as parameter. Intuitively, the results show that for similar PFMs the length of the window is smaller than for dissimilar PFMs given the same probability. Equally, the probability for the co-occurrence event is higher for similar PFMs and smaller for dissimilar events. Hence, we can split sequences into small (non-overlapping) parts and for each part determine whether it is a CRM with a given false positive probability.

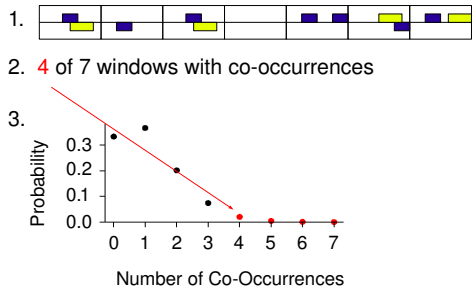

Figure 6.2: Proposed algorithm to compute co-operativity of a pair of TFs: First, divide sequence into windows. Second, count windows containing at least one hit of each TF. Compute corresponding count distribution under random sequence model to retrieve p-values for co-operativity.

Furthermore, one is interested in whether specific TFs are generally involved in the same CRMs. We call this co-operativity of TFs. Here, we also show how to compute the significance of co-operativity of TFs. The sequence is divided into equal-sized windows covering the whole sequence. We compute a p-value for the number of observed CRMs (windows with the co-occurrence event) since we can derive the count distribution of CRMs. We differentiate between non-overlapping and overlapping windows. In case of non-overlapping windows the count distribution is exact besides the approximations in the calculation of the co-occurrence event. In contrast, overlapping windows introduce further dependencies. Hence, we compute error bounds using the Chen-Stein method. The accuracy of the results is shown in Chapter 8.

In the remaining part of this chapter, we derive the formulae for CRMs and co-operativity. In the first part, we restrict our view to pairs of TFs. Hence, we define a CRM to be a window of given length with at least one occurrence for each of the two TFs (the co-occurrence event). After introducing preliminary notation, we compute the probability of the co-occurrence event on a random sequence of the size of the window. Based on this result, we can also compute the size of the window such that the co-occurrence probability is given as a parameter. Instead of considering theoretical occurrence frequencies for the TFs, we incorporate empirical occurrence frequencies in the next step. Then, we extend the statistic from detecting single CRMs to compute the co-operativity of a pair of TFs. Statistically, we can translate this question and ask whether one observes more CRMs of a given pair of TFs than one would do by chance. We assume that the sequence is divided into equal-sized windows covering the whole sequence. Hence, one can employ a Binomial or Poisson distribution for non-overlapping CRMs to compute the distribution of the number of CRMs. If the CRMs are allowed to overlap, dependencies between the windows are introduced. This leads to an approximation error for the count distribution. Therefore, we explicitly state the Chen-Stein error bounds for using a Poisson distribution.

6.2 Number of Hits

We assume that each TF is given by a PFM. For each position j of a sequence, we have an indicator random variable $Y_j(A)$ which is 1 if the summed score at this position reaches

the threshold. We denote the random variables for the complementary strand by a prime, e.g. $Y'_j(A)$. The threshold can be controlled by the type I error

$$\alpha_A := \mathbb{P}_\mu(Y_j(A) = 1),$$

where μ is the null model corresponding to a random sequence. The model for the random sequence is assumed to be an i.i.d. sequence defined by the GC content. We assume this simple background model, since we require the distribution of hits on both strands to be equal.

As stated before, a CRM is a window of given length w with at least one hit for TF A and one hit of TF B. We split up the calculation of this co-occurrence event into three parts: Let $N^*_w(A) = \sum_{j=1}^{w}(Y_j(A) + Y'_j(A))$ denote the random variable for the number of hits of TF A in a random sequence of length w where we allow hits overlapping the boundary of the window. Now, we can state the probability $p(w)$ of a CRM in a given window of length w by

$$p(w) := \mathbb{P}_\mu(N^*_w(A) > 0, N^*_w(B) > 0). \tag{6.1}$$

Direct calculation using the inclusion-exclusion formula yields

$$p(w) = 1 - \mathbb{P}_\mu(N^*_w(A) = 0) - \mathbb{P}_\mu(N^*_w(B) = 0) + \mathbb{P}_\mu(N^*_w(A) = 0, N^*_w(B) = 0). \tag{6.2}$$

In the remaining part of this section, we show how to compute these three probabilities.

6.2.1 Counts for one TF

One strand only

We start with the calculation of $\mathbb{P}_\mu(N^*_w(A) = 0)$. We can rewrite the event $\{N^*_w(A) = 0\}$ using the hit indicator random variables. For the beginning, we ignore the complementary strand. In this case, we obtain:

$$\mathbb{P}_\mu(N^*_w(A) = 0) = \mathbb{P}_\mu\left(\sum_{j=1}^{w} Y_j(A) = 0\right). \tag{6.3}$$

This means we have to compute the joint event that all $Y_j(A) = 0$ for $0 < j \leq w$. In a second step, we can conditionalize the last random variable at the end of the sequence on the remaining random variables

$$\mathbb{P}_\mu\left(\sum_{j=1}^{w} Y_j(A) = 0\right) = \mathbb{P}_\mu\left(Y_w(A) = 0 \,\middle|\, \sum_{j=1}^{w-1} Y_j(A) = 0\right) \cdot \mathbb{P}_\mu\left(\sum_{j=1}^{w-1} Y_j(A) = 0\right). \tag{6.4}$$

Denoting the length of TF A by ℓ_A, we can state the dependencies between $Y_w(A)$ and the remaining random variables. Since $Y_w(A)$ has the highest index w, all remaining variables are to the left. Thus, $Y_w(A)$ only depends on random variables where the corresponding motif overlaps with $Y_w(A)$. These are the random variables within a distance of $\ell_A - 1$: $\{Y_{w-j}(A)\}_{j=1}^{\ell_A-1}$. Hence, we can remove all other random variables from the conditional part. We obtain for the conditional probability of Eq. (6.4) after shifting the indices by $w - \ell_A$ positions and applying the definition of conditional probability

$$\mathbb{P}_\mu\left(Y_w(A) = 0 \,\middle|\, \sum_{j=1}^{w-1} Y_j(A) = 0\right) = \frac{\mathbb{P}_\mu\left(\sum_{j=1}^{\ell_A} Y_j(A) = 0\right)}{\mathbb{P}_\mu\left(\sum_{j=1}^{\ell_A-1} Y_j(A) = 0\right)}. \tag{6.5}$$

The numerator can be computed after some basic transformations: First, we conditionalize $Y_{\ell_A}(A)$ on the remaining random variables, again. Then, we consider the complementary event. Next, we use Bayes theorem to exchange the conditional part with the non-conditional part. Finally, we use a first-order approximation by decomposing the joint event into the product of the single conditional events

$$\begin{aligned}
&\mathbb{P}_\mu\left(\sum_{j=1}^{\ell_A} Y_j(A) = 0\right) \\
&= \mathbb{P}_\mu\left(\sum_{j=1}^{\ell_A-1} Y_j(A) = 0\right)\left[1 - \mathbb{P}_\mu\left(Y_{\ell_A}(A) = 1 \,\middle|\, \sum_{j=1}^{\ell_A-1} Y_j(A) = 0\right)\right] \\
&= \mathbb{P}_\mu\left(\sum_{j=1}^{\ell_A-1} Y_j(A) = 0\right) - \mathbb{P}_\mu\left(\sum_{j=1}^{\ell_A-1} Y_j(A) = 0 \,|Y_{\ell_A}(A) = 1\right) \cdot \mathbb{P}_\mu(Y_{\ell_A}(A) = 1) \\
&= \mathbb{P}_\mu\left(\sum_{j=1}^{\ell_A-1} Y_j(A) = 0\right) - \mathbb{P}_\mu(Y_{\ell_A}(A) = 1) \cdot \prod_{j=1}^{\ell_A-1} \left[1 - \mathbb{P}_\mu(Y_j(A) = 1 | Y_{\ell_A}(A) = 1)\right].
\end{aligned}$$

For convenience, we define variables for the conditional probabilities for any d

$$\bar{\gamma}_{-d}(A) := \mathbb{P}_\mu(Y_{j-d}(A) = 1 | Y_j(A) = 1).$$

In Section 5.6, we show how to compute these conditional probabilities efficiently. For later use, we set $\bar{\gamma}_0(A) := 0$ and correspondingly for B. Since $\mathbb{P}_\mu(Y_{\ell_A}(A) = 1) = \alpha_A$, we can substitute these terms in above equation and obtain

$$\mathbb{P}_\mu\left(\sum_{j=1}^{\ell_A} Y_j(A) = 0\right) = \mathbb{P}_\mu\left(\sum_{j=1}^{\ell_A-1} Y_j(A) = 0\right) - \alpha_A \cdot \prod_{d=0}^{\ell_A-1} \left[1 - \bar{\gamma}_{-d}(A)\right].$$

For extension to the complementary strand, we start the product with $d = 0$ which does not affect the product since $1 - \bar{\gamma}_0(A) = 1$. Obviously, we can apply the same steps

to $\mathbb{P}_\mu(\sum_{j=1}^{\ell_A-1} Y_j(A) = 0)$ which yields following recursive formulae (see figure 6.3 for an illustration)

$$\begin{aligned} f_A(1) &:= \mathbb{P}_\mu(Y_1(A) = 0) = 1 - \alpha_A, \\ f_A(k) &:= \mathbb{P}_\mu\left(\sum_{j=1}^{k} Y_j(A) = 0\right) = f_A(k-1) - \alpha_A \cdot \prod_{d=0}^{k-1}(1 - \bar{\gamma}_{-d}(A)). \end{aligned} \tag{6.6}$$

j	1	...	$k-1$	k
$Y_j(A)$	x	x	x	*

j	1	...	$k-1$	k
$Y_j(A)$	x	x	x	

Figure 6.3: The two figures illustrate the terms of $f_A(k)$ in Eq. (6.6): The left panel contains the right term of Eq. (6.6): $\alpha_A \cdot \prod_{d=0}^{k-1}(1 - \bar{\gamma}_{-d}(A))$. We can write symbolically $P(* = 1) \cdot P(\{x = 0\}|* = 1)$. The right panel contains $f_A(k-1) \hat{=} P(\{x = 0\})$. In this way, f_A computes the joint probability recursively.

The closed formula is given by

$$f_A(k) = (1 - \alpha_A) - \alpha_A \cdot \sum_{j=1}^{k-1} \prod_{d=0}^{j} (1 - \bar{\gamma}_d(A)) = 1 - \alpha_A \cdot \sum_{j=0}^{k-1} \prod_{d=0}^{j} (1 - \bar{\gamma}_{-d}(A)) \,.$$

For convencience, we define

$$\begin{aligned} \mathbb{P}_\mu\left(\sum_{j=w-\ell_A+1}^{w} Y_j(A) = 0\right) &= f_A(\ell_A) =: Z_A^*, \\ \mathbb{P}_\mu\left(\sum_{j=w-\ell_A+1}^{w-1} Y_j(A) = 0\right) &= f_A(\ell_A - 1) =: Z_A. \end{aligned}$$

Hence, we get for Eq. (6.5)

$$\mathbb{P}_\mu\left(Y_w(A) = 0 \,\middle|\, \sum_{j=w-\ell_A+1}^{w-1} Y_j(A) = 0\right) = Z_A^* \cdot Z_A^{-1}.$$

Substituting into Eq. (6.3) yields

$$\mathbb{P}_\mu\left(\sum_{j=1}^{w} Y_j(A) = 0\right) = Z_A^* \cdot Z_A^{-1} \cdot \mathbb{P}_\mu\left(\sum_{j=1}^{w-1} Y_j(A) = 0\right).$$

This recursive formula yields

$$\mathbb{P}_\mu\left(\sum_{j=1}^{w} Y_j(A) = 0\right) = \left(Z_A^* \cdot Z_A^{-1}\right)^{w-\ell_A} \cdot Z_A^*. \tag{6.7}$$

Substituting A by B in all variables and functions, yields the corresponding probability for TF B:

$$\mathbb{P}_\mu\left(\sum_{j=1}^{w} Y_j(B) = 0\right) = \left(Z_B^* \cdot Z_B^{-1}\right)^{w-\ell_B} \cdot Z_B^*. \tag{6.8}$$

Both strands

Incorporation of the complementary strand only doubles the number of random variables under consideration. In the first step, one also has to consider the palindromic hit (denoted by $\bar{\gamma}_0'(A)$). This means that we have to extend the product by this term. Then, an additional step follows for solving with respect to $\bar{\gamma}_0'(A)$. Thus, we have to add the product (without the palindromic hit) one more time to the sum. In fact, we use the symmetrie of the background model which yields for $0 \leq d < \ell_A$ and any j:

$$\begin{aligned}
\bar{\gamma}_{-d}(A) &= \mathbb{P}_\mu(Y_{j-d}(A) = 1 | Y_j(A) = 1) = \mathbb{P}_\mu(Y'_{j-d}(A) = 1 | Y'_j(A) = 1), \\
\bar{\gamma}'_{-d}(A) &= \mathbb{P}_\mu(Y'_{j-d}(A) = 1 | Y_j(A) = 1) = \mathbb{P}_\mu(Y_{j-d}(A) = 1 | Y'_j(A) = 1).
\end{aligned}$$

Now, we have to extend the definition of f since in each step we consider A and A':

$$\begin{aligned}
f_A(0) &:= \mathbb{P}_\mu(Y_1(A) + Y'_1(A) = 0) = (1 - \alpha_A) - \alpha_A \left[1 - \bar{\gamma}_0'(A)\right], \\
f_A(k) &:= \mathbb{P}_\mu\left(\sum_{j=1}^{k} (Y_j(A) + Y'_j(A) = 0\right) \qquad (6.9) \\
&= f_A(k-1) - \alpha_A \cdot \prod_{j=0}^{k-1} [1 - \bar{\gamma}_{-j}(A)][1 - \bar{\gamma}'_{-j}(A)] \\
&\quad - \frac{\alpha_A}{1 - \bar{\gamma}_0'(A)} \cdot \prod_{j=0}^{k-1} [1 - \bar{\gamma}_{-j}(A)][1 - \bar{\gamma}'_{-j}(A)]. \qquad (6.10)
\end{aligned}$$

In figure 6.4, $f_A(k)$ is graphically illustrated starting with the last term. The first equation is straight-forward to derive: In this step, we only consider the two random variables $Y_1(A)$ and $Y_0'(A)$. We shift the event $Y_1'(A)$ into the conditional part of the joint probability and perform the same transformation as above. The formula for $f(k)$ is derived by the same strategy. The first term following $f(k-1)$ contains the first-order approximated probability for $Y_k(A) + \sum_{j=1}^{k-1}(Y_j(A) + Y_j'(A)) = 0 | Y_k'(A) = 1$. Since we defined $\bar{\gamma}_0(A) = 0$, we start the product with index $d = 0$. The next term considers $\sum_{j=1}^{k-1}(Y_j(A) + Y_j'(A)) = 0 | Y_k(A) = 1$.

j	1	...	$k-1$	k
$Y_j(A)$	x	x	x	x
$Y'_j(A)$	x	x	x	*

j	1	...	$k-1$	k
$Y_j(A)$	x	x	x	*
$Y'_j(A)$	x	x	x	

j	1	...	$k-1$	k
$Y_j(A)$	x	x	x	
$Y'_j(A)$	x	x	x	

Figure 6.4: The three figures illustrate the terms of $f_A(k)$ in Eq. (6.9): The left panel contains the right term of Eq. (6.9): $\frac{\alpha_A}{1-\bar{\gamma}'_0(A)} \cdot \prod_{d=0}^{k-1}[1-\bar{\gamma}_{-d}(A)][1-\bar{\gamma}'_{-d}(A)]$. More symbolically written, this is equal to $P(* = 1) \cdot P(\{x = 0\}|* = 1)$. The middle panel contains $\alpha_A \cdot \prod_{d=0}^{k-1}[1 - \bar{\gamma}_{-d}(A)][1 - \bar{\gamma}'_{-d}(A)] \hat{=} P(* = 1) \cdot P(\{x = 0\}|* = 1)$ and the right panel is $f_A(k-1) \hat{=} P(\{x = 0\})$. In this way, f_A computes the joint probability recursively.

As we still start the product with index $d = 0$, we have to cancel the probability $1 - \bar{\gamma}'_0(A)$ of the product. In fact, the event $Y'_k(A) = 0$ has already been removed from the joint event in the term before. The closed formula is given by

$$f_A(k) = 1 - \alpha_A \left(1 + \frac{1}{1 - \bar{\gamma}'_0(A)}\right) \cdot \sum_{j=0}^{k-1} \prod_{d=0}^{j} \left[1 - \bar{\gamma}_{-d}(A)\right] \left[1 - \bar{\gamma}'_{-d}(A)\right].$$

Using the corresponding definitions for Z_A and Z_A^*, we obtain Eq. (6.7) for both strands. Again, computation for TF B is along the same line.

6.2.2 Joint Counts for two TFs

One strand only

The next task is to combine the statistic for both TFs which means to compute the probability to observe neither a hit of A nor a hit of B. For illustration purposes, we ignore the complementary strand for the beginning. The extension to include the random indicator variables of B is not as straight forward as to include the complementary strand because there are no corresponding symmetries between A and B. Furthermore, A and B might have different lengths. Therefore, we set $\ell_{AB} = \max(\ell_A, \ell_B)$. We define the conditional overlap probabilities between A and B for $0 \leq k < \ell_{AB}$ by

$$\begin{aligned} \bar{\gamma}_{-d}(A|B) &= \mathbb{P}_\mu(Y_{j-d}(A) = 1 | Y_j(B) = 1), \\ \bar{\gamma}_{-d}(B|A) &= \mathbb{P}_\mu(Y_{j-d}(B) = 1 | Y_j(A) = 1). \end{aligned}$$

To simplify notation, we introduce two auxiliary functions. g_A corresponds to the probability for the joint event of neither a hit for A nor for B given the hit of A at the most right position. Correspondingly we define g_B where the given hit is a hit of B:

$$
\begin{aligned}
g_A(k) &:= \mathbb{P}_\mu\left(\sum_{j=1}^{k-1}(Y_j(A)+Y_j(B))+Y_k(A)=0\right)\\
&= \frac{\alpha_A}{1-\bar{\gamma}_0(B|A)}\cdot\sum_{j=0}^{k-1}\prod_{d=0}^{j}\left[1-\bar{\gamma}_{-d}(A)\right]\cdot\left[1-\bar{\gamma}_{-d}(B|A)\right],\\
g_B(k) &:= \mathbb{P}_\mu\left(\sum_{j=1}^{k}(Y_j(A)+Y_j(B))=0\right)=\alpha_B\cdot\sum_{j=0}^{k-1}\prod_{d=0}^{j}\left[1-\bar{\gamma}_{-d}(B)\right]\cdot\left[1-\bar{\gamma}_{-d}(A|B)\right].
\end{aligned}
$$

Here, we arbitrarily defined to remove $Y_j(B)$ from the joint event before removing $Y_j(A)$. Due to the symmetry of $\alpha_A[1-\bar{\gamma}_0(B|A)]=\alpha_B[1-\bar{\gamma}_0(A|B)]$ this choice has no impact on the result. We obtain for $f_{AB}(k)$ for A and B

$$f_{AB}(k)=1-g_A(k)-g_B(k).$$

Again, using the corresponding definitions for $Z^*_{AB}:=f_{AB}(l_{AB}-1)$ and $Z^*_{AB}:=f_{AB}(l_{AB})$, we can compute the probability to observe neither a hit of TF A nor of TF B:

$$\mathbb{P}_\mu\left(\sum_{j=1}^{w}[Y_j(A)+Y_j(B)]=0\right)=\left(Z^*_{AB}\cdot Z^{-1}_{AB}\right)^{1-l_{AB}}\cdot Z^*_{AB}. \tag{6.11}$$

Both strands

Now, we can incorporate the complementary strand, again. For this purpose, we introduce more functions h which are similar to g. The formulae are derived in the same way as the functions g except that each step k corresponds to four random variables (both TFs and both strands). Again, we have to choose an order of removal from the joint. We proceed by starting with B', then B, next A', and finally A.

$$
\begin{aligned}
h_A(k) &:= \frac{\alpha_A\cdot\sum_{j=0}^{k-1}\prod_{d=0}^{j}\left[1-\bar{\gamma}_{-d}(A)\right]\cdot\left[1-\bar{\gamma}'_{-d}(A)\right]\cdot\left[1-\bar{\gamma}_d(B|A)\right]\cdot\left[1-\bar{\gamma}_d(B'|A)\right]}{\left[1-\bar{\gamma}'_0(A)\right]\cdot\left[1-\bar{\gamma}_0(B|A)\right]\cdot\left[1-\bar{\gamma}_0(B'|A)\right]},\\
h_{A'}(k) &:= \frac{\alpha_A\sum_{j=0}^{k-1}\prod_{d=0}^{j}\left[1-\bar{\gamma}_{-d}(A)\right]\cdot\left[1-\bar{\gamma}'_{-d}(A)\right]\cdot\left[1-\bar{\gamma}_d(B|A')\right]\cdot\left[1-\bar{\gamma}_d(B'|A')\right]}{\left[1-\bar{\gamma}_0(B|A')\right]\cdot\left[1-\bar{\gamma}_0(B'|A')\right]},\\
h_B(k) &:= \frac{\alpha_B\cdot\sum_{j=0}^{k-1}\prod_{d=0}^{j}\left[1-\bar{\gamma}_{-d}(B)\right]\cdot\left[1-\bar{\gamma}'_{-d}(B)\right]\cdot\left[1-\bar{\gamma}_d(A|B)\right]\cdot\left[1-\bar{\gamma}_d(A'|B)\right]}{1-\bar{\gamma}'_0(B)},\\
h_{B'}(k) &:= \alpha_B\cdot\sum_{j=0}^{k-1}\prod_{d=0}^{j}\left[1-\bar{\gamma}_{-d}(B)\right]\cdot\left[1-\bar{\gamma}'_{-d}(B)\right]\cdot\left[1-\bar{\gamma}_d(A|B')\right]\cdot\left[1-\bar{\gamma}_d(A'|B')\right].
\end{aligned}
$$

Hence, we get following definition of f_{AB}:

$$f_{AB}(k) = 1 - h_A(k) - h_{A'}(k) - h_B(k) - h_{B'}(k).$$

Finally, substituting the corresponding variables Z_{AB} and Z^*_{AB} into Eq. (6.11), we obtain the first order approximation for $p(w)$.

6.3 Probability for Co-Occurrence

In this subsection, we explicitly state the formula for the probability $p(w)$ of a co-occurrence event and subsequently simplify it by using certain approximations. Afterwards, we invert the formula to be able to compute the window size $w(p)$ given the co-occurrence probability p.

Using the derived expressions (Eq. (6.7), (6.8), and (6.11)), we can state the co-occurrence probability in a first order approximation regarding the dependencies:

$$p(w) = 1 - \left(Z^*_A \cdot Z_A^{-1}\right)^{w-\ell_A} \cdot Z^*_A - \left(Z^*_B \cdot Z_B^{-1}\right)^{w-\ell_B} \cdot Z^*_B + \left(Z^*_{AB} \cdot Z_{AB}^{-1}\right)^{w-l_{AB}} \cdot Z^*_{AB}.$$

We can simplify this expression by using some approximations. In fact, one chooses the thresholds for the PSMs such that $\alpha_A \ll 1$ and $\alpha_B \ll 1$ holds. Hence, $Z^*_A, Z^*_B, Z^*_{AB} \approx 1$ since the probability to find no hit on a sequence of the length of the PFM is very high. The same is true for Z_A, Z_B, Z_{AB} which are even greater than the corresponding Z^*s. Furthermore, the window size is much larger than the PFM lengths: $w \gg \ell_A, \ell_B, l_{AB}$. Hence, we can remove ℓ_A, ℓ_B, l_{AB} from the exponents. Using these approximations, and applying the limit of the exponential series three times, we obtain:

$$\begin{aligned} p(w) &\approx 1 - \left(Z^*_A \cdot Z_A^{-1}\right)^{w} - \left(Z^*_B \cdot Z_B^{-1}\right)^{w} + \left(Z^*_{AB} \cdot Z_{AB}^{-1}\right)^{w} \\ &\approx 1 - e^{(1-Z^*_A \cdot Z_A^{-1})\cdot w} - e^{(1-Z^*_B \cdot Z_B^{-1})\cdot w} + e^{(1-Z^*_{AB} \cdot Z_{AB}^{-1})\cdot w}. \end{aligned}$$

In a final step, we substitute the exponents without w by new variables and obtain

$$\begin{aligned} r_A &= 1 - Z^*_A \cdot Z_A^{-1}, \qquad r_B = 1 - Z^*_B \cdot Z_B^{-1}, \qquad r_{AB} = 1 - Z^*_{AB} \cdot Z_{AB}^{-1}, \\ p(w) &\approx 1 - e^{-r_A \cdot w} - e^{-r_B \cdot w} + e^{-r_{AB} \cdot w}. \end{aligned} \tag{6.12}$$

In fact, we have approximated the three random variables $N^*(A), N^*(B), N^*(A)+N^*(B)$ by three Poisson distributions $\mathcal{P}(r_A), \mathcal{P}(r_B), \mathcal{P}(r_{AB})$ and substituted into Eq. (6.2). Note that $N_a + N_B$ is computed by considering first-order dependencies between both TFs. Since we only evaluate this random varibale at $N_A + N_B = 0$, the Poisson approximation is valid.

6.3.1 Calculate Window Size

In Eq. (6.12), we present the formula to compute the co-occurrence probability in a window of size w. In practise, the probability for the co-occurrence event is given as parameter and the window size has to be computed. In this case, we have to find the roots of

$$1 - \exp(-r_A \cdot w) - \exp(-r_B \cdot w) + \exp(-r_{AB} \cdot w) - p.$$

Using the Newton approach, we obtain following recursion starting from a chosen initial value w_0:

$$w_{i+1} = w_i - \frac{1 - \exp(-r_A \cdot w_i) - \exp(-r_B \cdot w_i) + \exp(-r_{AB} \cdot w_i) - p}{r_A \exp(-r_A \cdot w_i) + r_B \exp(-r_B \cdot w_i) - r_{AB} \exp(-r_{AB} \cdot w_i)}.$$

In case one requires a closed formula, one can apply a Taylor expansion to Eq. (6.12):

$$p(w) = 1 + \sum_{k=0}^{\infty} \frac{\left(r_{AB}^k - r_A^k - r_B^k\right) \cdot (-w)^k}{k!}$$

E.g., the formula for a 2nd order expansion which already gives accurate results for small p is given by

$$w(p) = \frac{r_{AB} - r_A - r_A}{r_{AB}^2 - r_A^2 - r_A^2} + \sqrt{\left(\frac{r_{AB} - r_A - r_A}{r_{AB}^2 - r_A^2 - r_A^2}\right)^2 + \frac{2p}{r_{AB}^2 - r_A^2 - r_A^2}}.$$

6.3.2 Empirical Frequencies for Hits

The probability for a false positive (α) can be derived from the PSMs and the background model as shown at the beginning of this section. For simulated sequences, this probability fits well to the observed frequency of hits. In contrast, biological sequences might contain an enriched number of hits. In this case, the higher number of hits would bias the number of windows with co-occuring binding sites. Thus, the enrichment should be incorporated into the background model.

We suggest to estimate α_A resp. α_B by counting the number of hits and dividing by twice the sequence length because of the complementary strand. The estimates $\hat{\alpha}_A$ and $\hat{\alpha}_B$ should then be substituted in all formulae of this section.

6.4 CRMs for a Set of TFs

So far, we derived formulae to compute the co-occurrence probability for pairs of TFs. Here, we briefly extend the approach to deal with a set $\mathcal{T}$ of TFs with size $|\mathcal{T}|$. Eq. (6.2) reduces the calculation of the co-occurrence probability to compute the (joint) events of zero counts of the TFs. For a set of TFs, we simply have to apply the inclusion-exclusion formula on the count variables of all TFs:

$$\mathbb{P}_\mu(\min_{T\in\mathcal{T}} N_w^*(T) > 0) = 1 - \sum_{T\in\mathcal{T}} \mathbb{P}_\mu(N_w^*(T)=0) + \sum_{T\in\mathcal{T}}\sum_{U\in\mathcal{T}\setminus T} \mathbb{P}_\mu(N_w^*(T)=0, N_w^*(U)=0) - \dots$$

Hence, we only have to show how to compute the probability for no hits for all subsets $\mathcal{U} \in \mathscr{P}(\mathcal{T})$ of the power set of $\mathcal{T}$. This means, we have to compute Z and Z^* for all $\mathcal{U}$. Let $l_\mathcal{T}$ denote the maximum length of the TFs $\ell = \max_{T\in\mathcal{T}} \ell_T$. Then, we obtain

$$Z^*_\mathcal{U} = f_\mathcal{U}(\ell_\mathcal{T}), \qquad Z_\mathcal{U} = f_\mathcal{U}(\ell_\mathcal{T} - 1)$$

where f is defined by

$$f_\mathcal{U} = 1 - \sum_{t=1}^{|\mathcal{U}|} \left(h_{t,\mathcal{U}} + h'_{t,\mathcal{U}}\right).$$

Note that the functions h also depend on the subset of TFs under consideration. Denoting the elements of $\mathcal{U}$ by $\{U_1, \dots, U_{|\mathcal{U}|}\}$, we obtain for h:

$$h_{t,\mathcal{U}}(k) = \frac{\alpha_{U_t} \cdot \sum_{j=0}^{k} \prod_{d=0}^{j} \left[1-\bar\gamma_{-d}(U_t)\right] \left[1-\bar\gamma'_{-d}(U_t)\right] \prod_{\tau=1,\tau\neq t}^{|\mathcal{U}|} \left[1-\bar\gamma_{-d}(U_\tau|U_t)\right] \left[1-\bar\gamma_{-d}(U'_\tau|U_t)\right]}{\left[1-\bar\gamma'_0(U_t)\right] \prod_{\tau=t+1}^{|\mathcal{U}|} \left[1-\bar\gamma_0(U_\tau|U_t)\right] \left[1-\bar\gamma_0(U'_\tau|U_t)\right]},$$

$$h'_{t,\mathcal{U}}(k) = \frac{\alpha_{U_t} \cdot \sum_{j=0}^{k} \prod_{d=0}^{j} \left[1-\bar\gamma_{-d}(U_t)\right] \left[1-\bar\gamma'_{-d}(U_t)\right] \prod_{\tau=1,\tau\neq t}^{|\mathcal{U}|} \left[1-\bar\gamma_{-d}(U_\tau|U'_t)\right] \left[1-\bar\gamma_{-d}(U'_\tau|U'_t)\right]}{\prod_{\tau=t+1}^{|\mathcal{U}|} \left[1-\bar\gamma_0(U_\tau|U'_t)\right] \left[1-\bar\gamma_0(U'_\tau|U'_t)\right]}.$$

6.5 p-value for Co-operativity

Previously, we showed how to compute the co-occurrence probability $p(w)$ in a given window, see Eq. (6.12). To compute co-operativity, we suggest to decompose the sequence into non-overlapping windows of equal size and count the number x of CRMs (windows with the co-occurrence event). Next, we can compute a p-value for the observed number of CRMs by using the Binomial distribution $\mathcal{B}(x; p(w), n/w)$ where n is the sequence length and w

the window size. Denoting the Bernoulli random variable of the ith window by W_i, and the number of windows by $m = n/w$, we directly obtain

$$W := \sum_{i=1}^{m} W_i \sim \mathcal{B}(x; p(w), m) \tag{6.13}$$

since we assumed that non-overlapping windows are independent. Since the probability for co-occurrence event is rather small while the number of windows is high, this distribution is asymptotically a Poisson distribution $\mathcal{P}(\vartheta)$ with $\vartheta = p(w) \cdot m$ if $p(w) \to 0$ and $m \to \infty$. In this thesis, we focus on the Poisson distribution.

6.5.1 Bounds for Overlapping Windows

Considering overlapping windows necessiates the step size s as an additional parameter. The number m of windows becomes $m = n/s - w + 1$. We assume that n, s, w are chosen such that $m, n, s, w \in \mathbb{N}^+$ and $s < w < \frac{1}{2}n$. Obviously, overlapping windows are dependent on each other. In this case, we can still use a Binomial or Poisson distribution but the dependencies lead to an error in the approximation. Using the Chen-Stein method (Chen, 1975), the error can be quantified. The quantification is done in terms of the total variation distance. Let U and V be any two random processes with values in the same space E, then the total variation distance between their distributions (denoted by $\mathcal{L}(\cdot)$) is

$$d_{\mathrm{TV}}\left(\mathcal{L}(U), \mathcal{L}(V)\right) = \sup_{D \subset E} |\mathbb{P}(U \in D) - \mathbb{P}(V \in D)|$$

where D is assumed to be measurable. Here, we focus on the Poisson Approximation since it yields slightly better error bounds. Thus, we calculate the bound for $d_{\mathrm{TV}}(\mathcal{L}(W), \mathcal{P}(\vartheta))$. Let denote $I := \{i : 0 < i \leq m\}$ the index set of the Bernoulli variables. The main idea is to define for each Bernoulli variable W_i a set $B_i \subseteq I$ of variables which have strong dependencies with W_i. We also require $i \in B_i$. In our case, there are only local dependencies since only overlapping windows are dependent on each other. Therefore, we capture all dependencies in the sets B_i which means that for each window i the set B_i contains the index i and the indices of overlapping windows to the left and to the right. Hence, we obtain the bound derived from Theorem 1 in Arratia *et al.* (1990) using an improved bound (Barbour *et al.*, 1992):

$$d_{\mathrm{TV}}\left(\mathcal{L}(W), \mathcal{P}(\vartheta)\right) \leq \frac{1 - e^{-\vartheta}}{\vartheta} \cdot (b_1 + b_2)$$

with

$$b_1 := \sum_{i \in I} \sum_{j \in B_i} \mathbb{E}[W_i] \cdot \mathbb{E}[W_j], \qquad b_2 := \sum_{i \in I} \sum_{j \in B_i, j \neq i} \mathbb{E}[W_i \cdot W_j].$$

The bound b_1 is straight forward to compute as it only contains the first moments. We have to consider the fact that the B_is for the first and last few windows contain less dependent

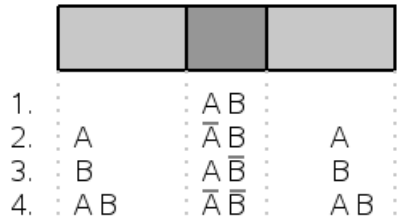

Figure 6.5: The four disjoint events for two windows where the dark grey area indicates the overlap. Regions containing an A or B must necessarily contain at least one hit of the corresponding TF while $\bar{A}$ and $\bar{B}$ label regions where the respective TF must not occur. In blank regions, any TF and combinations of TFs might be present.

variables than windows in the middle of the sequence. Let $r = w/s$, then for example, the first window has $r - 1$ overlapping windows, thus, $|B_1| = r$ since we also include index 1 in the set. The second window additionally overlaps with the first window, thus, $|B_2| = |B_1| + 1$. The set size is incremented by 1 until the $r + 1$th window as this window has equal number of overlaps to the left and to the right. At the end of the sequence, the set size is decremented in the same way. Hence, we obtain

$$b_1 = 2 \cdot \sum_{i=1}^{r} (r + i - 1)\, p(w)^2 + \sum_{i=1}^{m-2r} (2r - 1)\, p(w)^2 = p(w)^2 \left(r\,(1 - r + 2m) - m\right).$$

The second bound b_2 is more complicated to calculate because it contains the second moments. Since we consider Bernoulli variables, the second moment is the probability that both variables are equal to one: $\mathbb{E}[W_i W_{i+k}] = \mathbb{P}_\mu(W_i = 1, W_{i+k} = 1)$. Considering only two TFs A and B, we can write this probability in terms of the count random variables by decomposing it into four disjoint events (see Fig. 6.5):

1. There is at least one hit of A and one hit of B in the overlapping part of the two windows.

2. We observe at least one hit for A in each of the non-overlapping parts, and at least one hit for B and no hit for A in the overlapping part.

3. We observe at least one hit for B in each of the non-overlapping parts, and at least one hit for A and no hit for B in the overlapping part.

4. The overlapping part contains neither a hit of A nor of B while both non-overlapping parts contain at least one hit of A and at least one hit of B.

The decomposition can be extended for three TFs A, B, and C by splitting each event into two: First, where C has to occur in the overlapping part. Second, where C has to occur in both non-overlapping parts but not in the overlapping part. Adding more TFs splits each event further. Thus, we get for a set of TFs $\mathcal{T}$

$$\begin{aligned}\mathbb{E}[W_i W_{i+k}] &= \mathbb{P}_\mu(W_i = 1, W_{i+k} = 1) \\ &= \sum_{\mathcal{U} \in \mathscr{P}(\mathcal{T})} \mathbb{P}_\mu\left(\min_{U \in \mathcal{U}} N_d^*(U) > 0\right)^2 \mathbb{P}_\mu\left(\max_{U \in \mathcal{U}} N_{w-d}^*(U) = 0, \min_{U \in \mathcal{T} \setminus \mathcal{U}} N_{w-d}^*(U) > 0\right)\end{aligned}$$

where the size of each non-overlapping part is $d = k \cdot s$ while the overlapping part has a length of $w - d$. This equation can easily be computed since the joint events can be transformed into events we have computed before. E.g. considering only a pair of TFs, we get the following second moment:

$$\begin{aligned}\mathbb{E}[W_i W_{i+k}] &= \mathbb{P}_\mu(W_i = 1, W_{i+k} = 1) \\ &= \mathbb{P}_\mu(N_{w-d}^*(A) > 0, N_{w-d}^*(B) > 0) \\ &\quad + \mathbb{P}_\mu(N_d^*(A) > 0)^2 \cdot \mathbb{P}_\mu(N_{w-d}^*(A) = 0, N_{w-d}^*(B) > 0) \\ &\quad + \mathbb{P}_\mu(N_d^*(B) > 0)^2 \cdot \mathbb{P}_\mu(N_{w-d}^*(A) > 0, N_{w-d}^*(B) = 0) \\ &\quad + \mathbb{P}_\mu(N_d^*(A) > 0, N_d^*(B) > 0)^2 \cdot \mathbb{P}_\mu(N_{w-d}^*(A) = 0, N_{w-d}^*(B) = 0) \\ &= p(w-d) + \left(1 - e^{-dr_A}\right)^2 \cdot \left[1 - e^{-(w-d)r_B} - p(w-d)\right] \\ &\quad + \left(1 - e^{-dr_B}\right)^2 \cdot \left[1 - e^{-(w-d)r_A} - p(w-d)\right] + p(d)^2 \cdot e^{-(w-d)r_{AB}}.\end{aligned}$$

To compute the bound, we observe that $\mathbb{E}[W_i W_{i+k}]$ is independent of i since all W_i are identically distributed and have the same pairwise dependencies. Therefore, we clarify notation by defining $\zeta_k := \mathbb{E}[W_i W_{i+k}]$. For the same reason, we also obtain $\zeta_k = \mathbb{E}[W_i W_{i-k}]$. Using the further definition of $\zeta = \sum_{k=1}^{r-1} \zeta_k$, we obtain for bound b_2 applying the same logic as above:

$$b_2 = 2 \cdot \sum_{i=1}^{r} \left[\zeta + \sum_{k=1}^{i-1} \zeta_k\right] + 2(m - 2r)\zeta = 2\left(m\zeta - r\zeta + \sum_{i=1}^{r} \sum_{k=1}^{i-1} \zeta_k\right).$$

Here, we assume that the empty sum ($\sum_{k=1}^{i-1} \zeta_k$ for $i = 1$) is equal to 0.

Part II

Applications

Chapter 7

Count Statistics

In Chapter 5, a new approximation for the count statistic of TFBS is derived. Here, we compare the results of the approximation to competing approaches, as well as to results based on a simulation. Therefore, we create a small set of artificial PFMs incorporating typical difficulties for count statistics like self-overlap and palindromicity. After describing the simulation, we present the results. Finally, we compare the running time of our approximation to the running time of the exact approach (Zhang *et al.*, 2007).

7.1 Simulation

7.1.1 Sequences

To compare the new statistic with previous approaches, we use a simulation study. We simulate 100,000 sequences of length 10,000 with an arbitrarily selected GC content of 40% using an i.i.d. model. These sequences are annotated by binding sites of artificially constructed and real PFMs (see next paragraph). Counting the number of hits/clumps per sequence and computing the frequency for each count, one retrieves a simulated count distribution for each PFM. Comparing the simulated count distribution with the theoretically approximated distributions, we can easily assess the accuracy of the approximations, as well, as comparing the approximations between themselves.

7.1.2 PFMs

We artificially construct four PFMs, each of them carrying a certain characteristic regarding self-overlap, see Figure 7.1 for sequence logos (Crooks *et al.*, 2004):

- 'nothing': a PFM without any self-overlaps
- 'palindrome': a PFM with a likely hit on the complementary strand
- 'repeat': a PFM where the suffix matches the prefix such that one expects overlapping hits in a chain.
- 'repeatpalindrome': a combination of the 'palindrome' and the 'repeat'.

Furthermore, we arbitrarily pick one real PFM from TransFac (Matys *et al.*, 2003) with a self-overlapping structure to show that the gain in accuracy is relevant in practise. We select the palindromic PFM M00950 corresponding to the binding site of the MADS domain protein AGAMOUS-like 15 (AGL15) (Tang and Perry, 2003).

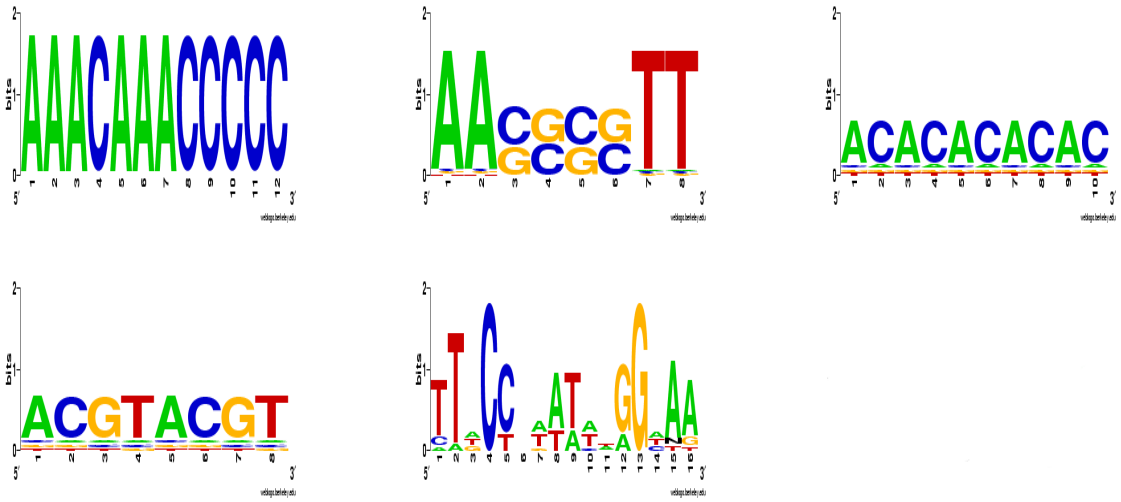

Figure 7.1: Sequence Logos from left to right and top to bottom: 'nothing', 'palindrome', 'repeat', 'repeatpalindrome', and 'M00950'.

In a pre-processing step, we regularize the PFMs to ensure strictly positive frequencies. Thus, we add pseudocounts to the position specific distributions according to the information content of the position (Rahmann, 2003). In fact, positions with low information content are shifted towards the background distribution. For positions with high information content, the difference to the background distributions is enforced. Then, we compute PSSMs from the regularized PFMs by taking the log-likelihood ratio of the nucleotide frequencies of the binding site and the background model.

We set the threshold for each PFM according to Pape *et al.* (2006) ensuring that the probability α_{500} for at least one false positive in a sequence of length 500 for any higher threshold is 10% at maximum. Thus, in the case that one cannot balance α_{500} and β, we obtain $\alpha_{500} \approx 0.1$. Furthermore, in case of a balanced threshold, α_{500} and β will not be exactly equal due to the discrete nature of the score and, thus, of α_{500} and β. Applying this procedure, the number of compatible words for PFM 'nothing' using a threshold $t = 136$ with $\alpha_{500} = 0.013$ and $\beta = 0.01$ is 1,142 only containing unique words. PFM 'palindrome' ($t = 99$, $\alpha_{500} = 0.129$ and $\beta = 0.325$) encodes 48 words based on 24 unique words while PFM 'repeat' ($t = 128$, $\alpha_{500} = 0.118$ and $\beta = 0.553$) has 702 words without any non-unique words and, finally, 'repeatpalindrome' ($t = 125$, $\alpha_{500} = 0.157$ and $\beta = 0.649$) yields 50 words from which 25 are unique. Thus, the compatible set of both PFMs with a palindromic structure contain each word twice (for each strand once). The Transfac PFM 'M00950' ($t = 118$, $\alpha_{500} = 0.0785$ and $\beta = 0.0747$) has 846,976 words with 429,812 unique words.

7.2 Standard Count Statistics

In this section, we introduce the competing count statistics that we compare our approach with. They are applied on the set of compatible words after removing redundant words such that the assumptions are met.

- Binomial and Poisson Approximation without considering the self-overlap (see Section 3.4.2 for counts and Section 3.5.1 for clumps),
- Compound Poisson approach (Reinert and Schbath, 1999) considering homogeneous clumps (see Section 3.3.3),
- Compound Poisson approach (Roquain and Schbath, 2007) modelling heterogeneous clumps (see Section 3.5.1),
- and normal approximation (Waterman, 2000) for occurrences (see Section 3.5.1).

Note that enumeration of compatible words requires exponential time assuming a threshold selection method based on the type-I or type-II error probabilities (see Section 2.4.3). Furthermore, the self-overlaps introduce approximation errors. In more detail, the binomial/Poisson approximation cannot deal with self-overlaps.

The classical compound Poisson approach (Reinert and Schbath, 1999) has two big problems: First, the number of compatible words might be fairly high which leads to large bounds on the total variation distance. Second, incorporation of the complementary strand can only be achieved by extending the set of compatible words by all reverse complementary words. This almost always breaks the necessary assumption for the approximation that no word is a substring of any overlap of any two other words. Therefore, we only use the approximation for the clump statistics because there only the weaker assumption that no word is a substring of any other word has to be fulfilled. This is indeed the case except for a palindrome.

The improved compound Poisson approximation (Roquain and Schbath, 2007) only requires that no word is a substring of any other word for both the clump and the hit distribution. Hence, one would only expect problems with palindromes which lead to multi-sets of compatible words. So far, they cannot be handled by word counting approaches. Still, the main drawback to enumerate all the compatible words remains. Furthermore, the approach involves multiple matrix multiplications where the two dimensions of the matrices are equal to the number of compatible words. This leads to numerical instabilities for large sets of compatible words.

The main drawback of the normal approximation (Waterman, 2000) besides enumeration of all compatible words is its assumption that occurrences are required to have high probability. Obviously, this is not true for TFBS.

7.3 Comparison of the Different Approaches Using Simulated Data

We compare the approaches based on the p-values since the statistic will mainly be used to retrieve p-values for observed number of hits/clumps. We present them after taking the logarithm to base ten. Therefore, the p-p plots show log-p-values. The x-axis always refers to the simulated distribution while the y-axis corresponds to the approximated distribution. The more points are located on the diagonal, the better the approximation. Furthermore, points below the diagonal correspond to underestimation of the p-values while points above the diagonal are conservative approximations.

7.3.1 Artificial PFMs

Figure 7.2A shows the p-p plots of the 'nothing' PFM. Most of the points lie on the diagonal. Furthermore, there is no big difference between the binomial and Poisson approximations, as well, as the the approach from Roquain & Schbath and the new approach. Only for very small p-values, there is a subtle difference between the approximations: The binomial and Poisson approaches seem to slightly outperform the others. However, the very small p-values are based on very few sequences because such high numbers of hits/clumps do not occur very often. Therefore, we ignore these points for interpretation. Only the normal approximation underestimates the p-values systematically. As the Poisson approximation works better, obviously, the rare word assumption is fullfilled instead of the normal approximation assuming often-occurring words. The results for clumps do not differ. As there are no overlaps, both the hit and the clump statistics are similar. Obviously, the new approach captures this non-self-overlap.

Figure 7.2B contains the results for the PFM 'palindrome'. The single distribution lying on the diagonal corresponds to our new approach. The binomial, Poisson, the normal and the Roquain & Schbath approach substantially underestimate the p-values. For the binomial and Poisson approximation, this is due to the fact that the number of hits is higher since the PFM tends to hit on both strands the same time. Furthermore, there are always pairs of points very close to each other. This is due to the hit on the complementary sequence which always occurs with a hit on the $5' - 3'$ strand: Having one hit on one strand implies a second hit on the other strand. Obviously, only the new approach can deal with this. In contrast, the Roquain & Schbath approach does not lead to a reasonable approximation. Since the set of compatible words contains each word twice but the approach can only deal with a set of unique words, the weak approximation is not surprising. For statistics of clumps, the Roquain & Schbath and the new approach lie fairly on the diagonal, as well, as the Chen-Stein approximation. Here, the approximations for the Roquain & Schbath and the Chen-Stein approach work because for clumps redundant words in the set of compatible words have no influence.

Figure 7.2C compares the approximations for the 'repeat' PFM. Binomial, Poisson, and normal approximations look very similar. Neither these nor the other two approaches lie on the diagonal. Though, the Roquain & Schbath and the new approach are more similar to the diagonal. In general, the Roquain & Schbath estimates are lower than the ones from the new approach and fit better to the diagonal. In addition, both approaches are

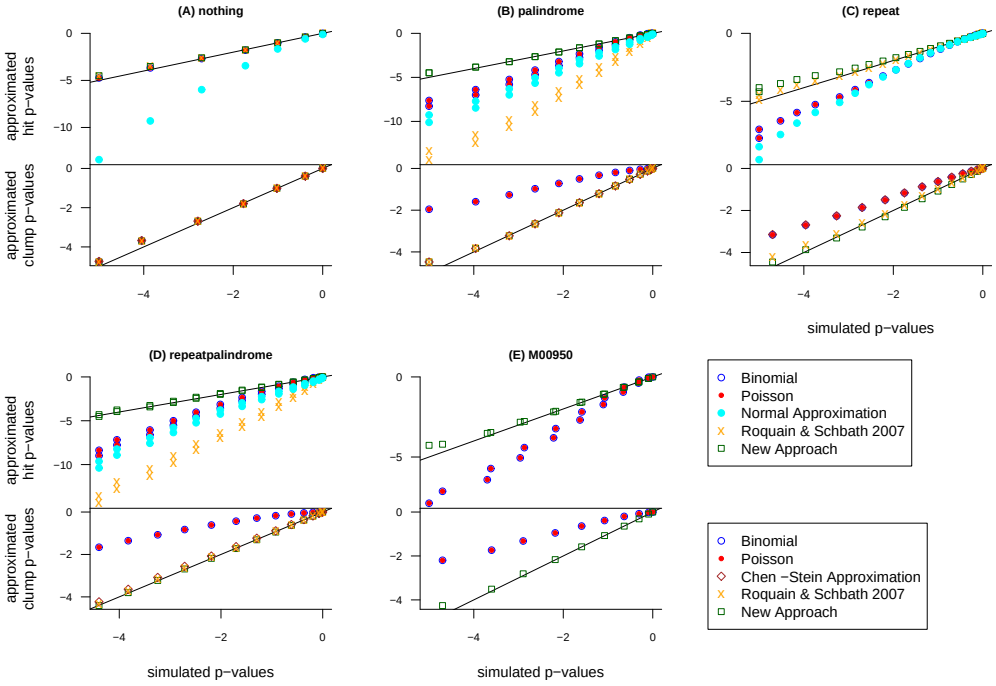

Figure 7.2: Comparison of the simulated p-values (x-axis) with the approximated p-values (y-axis) in a log-scale p-p plot.

conservative in contrast to the others which significantly underestimate the p-values. The approximations for the clump statistic are similar except that the new approach slightly underestimates the p-values for small number of clumps and overestimates them for higher number of clumps. In general, while the Roquain & Schbath approach yields higher estimates leading to a more accurate approximation for small number of clumps but a slightly weaker approximation for higher number of clumps. The other approaches overestimate the p-values significantly. Here, the Chen-Stein approximation performs as bad as the binomial/Poisson approximation since the assumption that no word is a substring of the concatenation of any other two words is extensively violated.

In Figure 7.2D the comparisons for the 'repeatpalindrome' PFM is shown. In general, the approximations are similar to the ones of the 'palindrome' PFM with some influence of the 'repeat' PFM. This shows that the new approach can deal with both types of similarity at the same time in contrast to all other approaches. Only the new approach leads to a reasonable approximation of p-values for the number of hits.

7.3.2 PFM M00950

The results for the Transfac PFM M00950 are shown in Figure 7.2E. Using the balanced threshold to obtain a probability of a false positive in a region of 500bp equal to 7.6%, the number of unique compatible words is equal to $429,812$. Therefore, comparison with the Chen-Stein approximation, the normal approximation, and the Roquain & Schbath approach is not possible because these statistics could not be computed in a feasible amount

	λ_1	λ_2	Comment
nothing	0.0123	0.0000	λ_1 small, $\lambda_2 = 0$
palindrome	0.0016	-0.0016	λ_1 small, $\lambda_2 = -\lambda_1$
repeat	0.2599	0.0000	λ_1 large, $\lambda_2 = 0$
repeatpalindrome	0.1792	-0.1526	λ_1 large, $\lambda_2 \approx -\lambda_1$
M00950	0.0455	-0.0435	$\lambda_2 \approx -\lambda_1$

Table 7.1: The two characteristic values given by the Eigenvalues of matrix A for each PFM.

of time. The comparison with the simulated p-values shows that the new approach fits very well. In contrast, the binomial/Poisson approximations show the typical significant deviation we have already seen for the artificial PFMs. Hence, in such a realistic framework, the new approach is the only possiblity to compute the count statistic without simulations.

7.4 Characteristic Values

Table 7.1 shows the characteristic values for each PFM. The 'nothing' PFM has a low first eigenvalue while the second eigenvalue is equal to zero. Since the PFM has no self-overlap, these two characteristic values confirm the analysis given in the method section. For the 'palindrome' PFM the equation $\lambda_1 \approx -\lambda_2$ holds because the only self-overlap is given by the the palindromic property of the PFM. The 'repeat' PFM has a much higher first eigenvalue than the 'nothing' PFM since it has a strong repeat-structure. Since there is no palindromic feature within the PFM, we obtain $\lambda_2 = 0$. In constrast, the 'repeatpalindrome' PFM contains both self-overlaps, thus, $\lambda_2 < 0$ but $\lambda_1 \neq -\lambda_2$. Finally, the PFM 'M00950' has also a clear palindromic self-overlap. All these observations are confirmed by the sequence logos (see Figure 7.1) and the resulting count statistics (see Figure 7.2). Thus, the characteristic values describe the self-overlapping features well. In the given cases, the self-overlap is clear for illustration purposes but in more difficult cases they shed light on the self-overlapping structure.

7.5 Running Time Comparison

Here, we compare the running time of the exact approach (Zhang *et al.*, 2007) described in Section 3.2.1 and Section 3.4.1 with the running time of the approximation of our approach. The implementation of the exact approach `MotifRankMatrix` (`http://bio.dlg.cn/MotifRankIntro.html`) offered by Zhang *et al.* (2007) is used for the exact calculation while the code for the approximation is based on our own code (http://mosta.molgen.mpg.de). We only slightly changed the `MotifRankMatrix` to implement time measure ability and to pass parameters from the command line (threshold, number of hits).

7.5.1 Simulation

We only measure the time of the calculations for each algorithm. Reading of the input and motif detection (for `MotifRankMatrix`) are discarded. Both programs use the same threshold which is controlled by the type I error rate α for a 500 bp region. The count distribution is computed until the approximated probabilities are smaller than 10^{-9}.

The simulation is performed for different sequence lengths (500bp to 10,000bp in 500bp steps), for different PFM lenghts (5 to 15 positions), and two tresholds ($\alpha = .01$ and $\alpha = .1$). In each simulation, we generate 10 PFMs: For each PFM, we randomly draw from a Dirichlet distribution with parameters $(1, 1, 1, 1)$ for each column a probability vector. This probability vector is multiplied by 30 and rounded to be an integer. This means, that the PFMs are based on 30 sequences. We ensure that the information content for each PFM is at least 2 bits.

7.5.2 Results

Figure 7.3A shows the comparison of the algorithms for different PFM lenghts on a sequence of length 5000bp. The exact approach has a exponential increase of the running time for increasing PFM length. E.g., a PFM of length 12 takes around 20 minutes for $\alpha = .1$ and 1 minute for $\alpha = .01$. Obviously, the threshold has a strong influence on the running time. The generated PFMs longer than 13bp cannot be computed for $\alpha = .1$ at all. PFM of length 15 with $\alpha = .01$ takes around 1.5h. In contrast, the running time of the approximation is always less than or equal to 1s. For smaller PFMs, the running time differs for $\alpha = .01$ and $\alpha = .1$ but for longer PFMs, the difference diminishes.

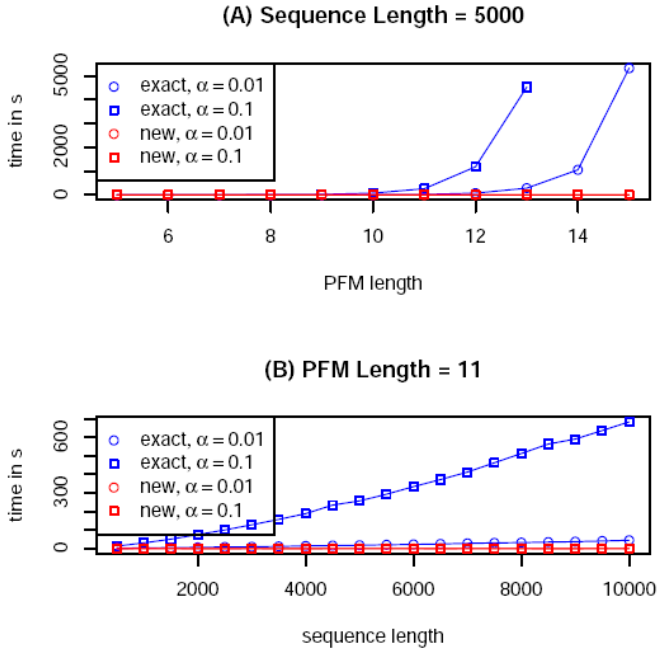

Figure 7.3: Running time comparison in s for randomly generated PFMs. Upper panels contains different PFMs length while the lower panel varies sequence length.

The influence of the sequence length for a PFM of length 11 is shown in figure 7.3B. The running time for the exact approach increases with sequence length. Again, we observe a strong difference for the choice of α for the exact approach. For a sequence length of 10,000 the running time is around 40s for $\alpha = .01$ and around 10 minutes for $\alpha = .1$. In contrast, the approaximations takes less than 1/2 s. Furthermore, the running time is independent of the sequence length for the approximation.

The results show that the running time of the exact approach is much higher than for the approximation. There are three parameters influencing the running time: threshold, PFM length, and sequence length.

Threshold The threshold defines the number of compatible word (words with a score higher than the threshold). The exact approach enumerates all compatible words. Since the number of compatible words grows exponentially with increasing PFM length, the influence on the running time is strong. In contrast, the approximation only enumerates the compatible scores. As mentioned in the corresponding manuscript, the bounds for the scores which have to be considered take advantage of a high and low thresholds. Therefore, the influence is very small.

PFM Length Here, we can draw a similar conclusion: The longer the PFM, the more compatible words. Thus, the influence for the exact approach is very strong and for the approximation very small.

Sequence Length The exact approach runs over the sequence similar to an automaton. Therefore, the running time depends on the sequence length. Furthermore, the number of hits depend on the sequence length. Since also the p-values for the number of hits are computed iteratively by running over the sequence, the running time strongly depends on the sequence length. In contrast, the approximation uses the compound Poisson distribution without a dependence on the sequence length.

7.6 Discussion

The results clearly indicate that the new approximation for the count statistic of PFMs retrieves very accurate results. In contrast to all previous works, we incorporate the complementary strand, which introduces further dependencies of overlapping hits. Due to explicit modelling of these dependencies, as well as dependencies between overlapping hits on the same strand, we are able to compute precise p-values for any PFM. Furthermore, we have shown how to compute two characteristic values describing the tendency of overlaps and palindromic hits of a given PFM with the same algorithm. The time complexity neither depends on the sequence length nor on the number of compatible words. Therefore, the algorithm is very efficient as the comparison with the running time of the exact approach shows. It might be further improved using the Fourier transform with the convolution theorem (Press *et al.*, 1992; Keich, 2005).

Comparison with other approaches shows that our approach has highest accuracy. Furthermore, most of the competing approaches enumerate all compatible words $\mathcal{W}$. Since $|\mathcal{W}|$ grows exponentially with the length of the PFM the overall-running time is exponential. Hence, the normal approximation (Waterman, 2000), the Chen-Stein approach (Reinert and Schbath, 1999), as well as the Roquain & Schbath approach (Roquain and Schbath, 2007), and the exact approach (Zhang *et al.*, 2007) cannot generally be applied in practice.

The exact approach seems to work in practical time for small PFMs on small sequences with a high threshold. If any of these parameters change, the running time grows significantly such that results cannot be obtained in reasonable time. Then, it is more appropriate to use the approximation which also has been shown to be very accurate.

We also want to mention that the memory requirements differ significantly: While the new approach has to keep all compatible words in memory, space requirement quickly exceed 1GB. In contrast, the two dimensional score distribution only needs a few MBs because only the quantiles are kept which reach the threshold.

A major drawback of the new approach is the restricted background model. So far, we only use a symmetric i.i.d. model defined by the GC content. Therefore, the statistics are symmetric between both strands. However, extension to symmetric Markov models is possible but increases computional complexity of the score convolution.

Chapter 8

Co-Occurrences and Co-Operativity

In Chapter 6, a new statistics for computing the significance of CRMs was developed. Here, a comparison for CRM probabilities for TF pairs is performed and analyzed based on simulations. In a second step, empirical frequencies are considered in the simulation by artificially implanting motifs. Lastly, co-operativity with non-overlapping and overlapping windows including error bounds are shown. Eventually, we discuss the impact of the new approach and its results.

8.1 Simulation

8.1.1 Alternative Approach ignoring Dependencies

For comparison, we compute the statistics for the probability of at least one occurrence of each TF by a simple approach ignoring dependencies between the positions. We obtain with similar notation as in Chapter 6 and incorporating the complementary strand

$$\begin{aligned} \mathbb{P}_\mu(N_w^*(A)=0) &\approx (1-\alpha_A)^{2w}, \\ \mathbb{P}_\mu(N_w^*(B)=0) &\approx (1-\alpha_B)^{2w}, \\ \mathbb{P}_\mu(N_w^*(A)=0, N_w(B)=0) &\approx [(1-\alpha_A)\cdot(1-\alpha_B)]^{2w}, \end{aligned}$$

where $N_w^*(A)$ denotes the number of occurrences of TF A in a sequence of length w including hits overlapping the boundary of the sequence. Furthermore, μ describes the symmetric i.i.d. sequence model. α_A is the probability of an occurrence of A at one position. For the rates, we obtain

$$r_A^* \approx 2\alpha_A - \alpha_A^2, \qquad r_B^* \approx 2\alpha_B - \alpha_B^2, \qquad r_{AB}^* \approx 2[1-(1-\alpha_A)\cdot(1-\alpha_B)] - [1-(1-\alpha_A)\cdot(1-\alpha_B)]^2.$$

Hence, we obtain for the probability $p(w)$ of at least one occurrence of each TF A and B in a sequence of length w

$$p^*(w) = 1 - e^{-r_A^*\cdot w} - e^{-r_B^*\cdot w} + e^{-r_{AB}^*\cdot w}.$$

Obviously, the approach does not incorporate similarities between the TFs A and B.

8.1.2 Sequences

The approximations of the new and the alternative approach are compared with results from a simulation. The co-occurence probability for a pair of TFs is the probability that both TFs have at least one hit in a random sequence of given length. Thus, we generate random sequences and count the number of co-occurence events. The frequency of these events should match with the approximations. We generated 100 sequences each of length 1,000,000 using an i.i.d. background model with a GC content of 50%. After dividing each sequence into the given window length, we use the total observed frequency of co-occurence events for comparison. We similarily proceed for the p-values for co-operativity using the number of counts of co-occurence events for each sequence.

8.1.3 PFMs

We create five PFMs manually, see Figure 8.1 for sequence logos (Crooks *et al.*, 2004). Since one can anticipate that overlapping structure between PFMs has an influence on the result of the approximation, we include PFMs with/without self- and inter-repetive elements:

- 'nothing': A PFM with consensus 'AAACAAACCCCC'.
- 'repeat': A PFM with strong self-repetitive elements with consensus 'ACACACACAC'.
- 'palindrome': A PFM which always yields a hit on the complementary strand if there is a hit on the leading strand with consensus 'AA[CG][CG][CG][CG]TT'.
- 'repeatpalindrome': A PFM which always yields a hit on both strands and has a repetitive structure with consensus 'ACGTACGT'.
- 'overlap': A PFM which overlaps with PFM 'repeat' and with PFM 'repeatpalindrome' with consensus 'ACACGT'.

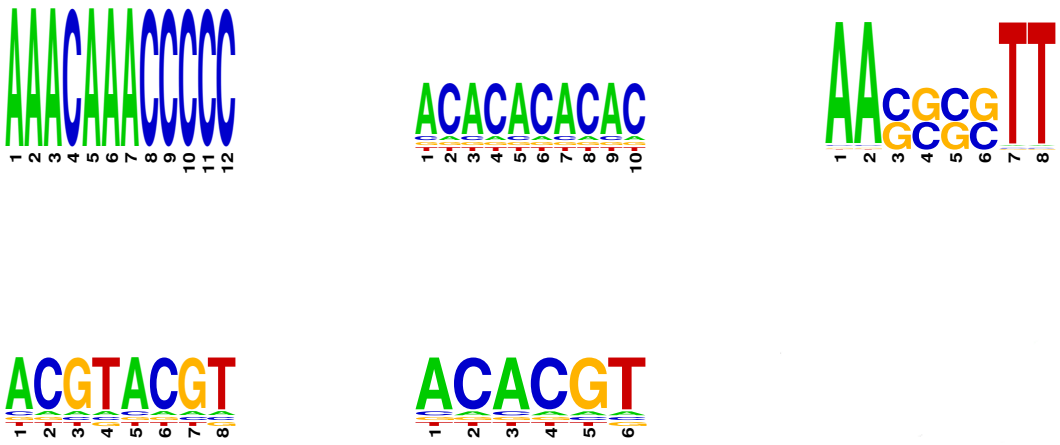

Figure 8.1: Sequence Logos from left to right and top to bottom: 'nothing', 'palindrome', 'repeat', 'repeatpalindrome', and 'overlap'.

Before we can annotate the sequences, one has to compute PSSMs from the PFMs by taking the log-likelihood ratio of the nucleotide frequencies of the binding site and the background model. We regularize the PSSMs to avoid zero entries in the PFMs by adding pseudocounts to the position specific distributions according to the information content of the position (Rahmann, 2003). Next, one has to determine a threshold for each PFM. The threshold controls the probabilities of the type I error α and the type II error β. Probabilities α and β can be computed by the convolution of the position-specific scores and the respective nucleotide probabilities as weights (Rahmann, 2003). We set the threshold using the type-I extended error with parameter 10% (see Section 2.4.3 for details).

8.1.4 Implanted Motifs

For the analysis of the impact of higher motif frequencies, we modified the random sequences by implanting additional motifs for some PFMs:

- 'palindrome': At each position of the sequence, we implanted the consensus of the palindrome with a probability of .0004. We choose this number since it is not a multiple of α. Otherwise, artifacts are generated especially for the palindrome motif since one always detects two hits at the same time.
- 'overlap': Using the same procedure we implanted the corresponding consensus with a probability of .0008.

8.2 Results

In this section, we compare the co-occurence probability p of the approximation with the results from the simulation. First, we show results for simulated sequences with the theoretically obtained rate α. Subsequently, we consider simulated sequences with implanted motifs. Thus, the theoretically computed α is too small. To capture the influence, we compare the approaches based on the theoretically computed and the empirically computed α.

8.2.1 Co-Occurence Probability

Figure 8.2 shows the comparison for all pairs of the five PFMs and for different window sizes: The size of the window increases from 100bp up to 5000bp in 100bp steps. On the one hand, one can clearly state that the result of the simulation fits well with the new approach which considers dependencies between positions. On the other hand, the alternative approach which assumes position independence performs well for the pairs 'nothing'-'repeat', 'nothing'-overlap', and little worse for 'repeat-overlap'. In cases including at least one PFM containing a palindrome, the alternative approach over-estimates the co-occurence probabilty. The worst result for the alternative approach is obtained by the pair 'palindrome'-'repeatpalindrome' where both PFMs contain the palindrome.

The alternative approach over-estimates the value for p because the probability of hit is higher than the probability of a hit given no hits at the surrounding positions (under the

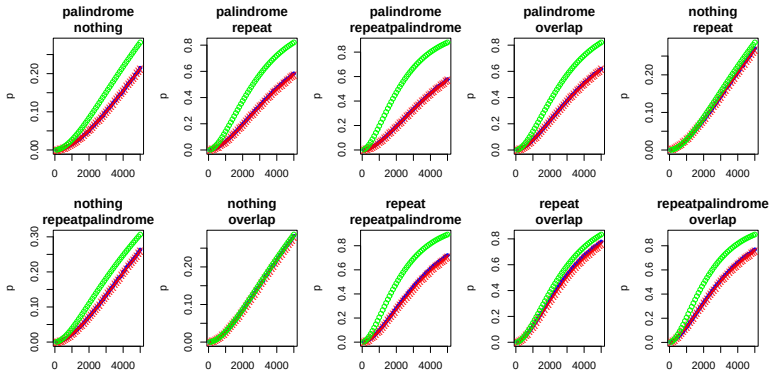

Figure 8.2: Pairs of TFBS and their probability (y-axis) for co-occurences in a window for different window lengths (x-axis). Blue circles are the probabilities yielded by simulation, red 'x' marks the new approach results and green '+' corresponds to the approach with assumed independencies.

assumption that the probability of a hit is very small). Since the palindromes and (less) the repetitive elements in the PFMs increase the strength of the position dependencies, it is not surprising that the alternative approach performs worse in these situations. Furthermore, the impact of palindromes is higher than for repetitive structures which is due to the fact that in case of our palindromic PFMs all positions are covered by both PFMs while the overlap probabilities for repeats are smaller as non-overlapping positions contain further randomness.

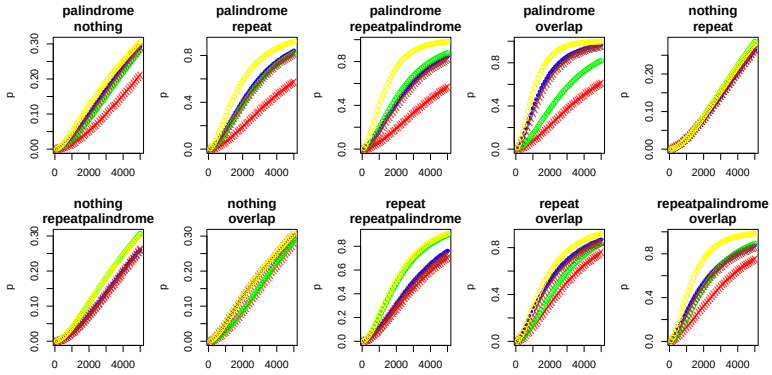

Figure 8.3: Pairs of TFBS and their co-occurence probability (y-axis) in a window for different window lengths (x-axis) where motifs for palindrome and overlap are implanted in the sequences. Blue circles are the co-occurence probabiities retrieved by simulation, red 'x' marks the new approach results based on theoretically computed αs, brown 'x' is for the new approach based on empirically retrieved αs, green '+' corresponds to the approach with assumed independencies with theoretically computed αs, and yellow '+' based on empirically retrieved αs.

Next, we present the results regarding random sequences with implanted motifs. Figure 8.3 contains all pair-wise co-occurence probabilities. In contrast to Fig. 8.2, the new approach based on theoretically computed α does not fit for the simulated probabilities if at least one of the two TFs is either palindrome or overlap. Hence, adjustment of α is necessary. The adjusted new approach accurately approximates the simulated probabilities in all cases.

Interestingly, the approach with independence assumption based on the theoretically computed α performs better than before if the palindrome is involved except for the pair with the overlap. Without implanted palindromes, this approach overestimated the probabilities. Here, we increase the frequency and, thus, obtain higher probability for co-occurences which match to the probability of the independence approach. Therefore, it seems that the error based on the self-overlap in the approximation for sequences without implanted motifs is eliminated by the error of the wrong frequency of the motif. Hence, this is likely to be an artifact. This is confirmed by the fact that the approximation for the pair 'palindrome' and 'overlap' yields a high error. Furthermore, the pair 'nothing' and 'overlap' which do not have and do not share similarities achieves the same error for the new approach and the independence approach with theoretically computed α in contrast to the two approaches based on empirically retrieved α. In general, also the independence approach with empirically retrieved α does not perform well in cases of similarities.

8.2.2 p-value for Co-operativity

In this subsection, we consider the p-value for co-operativity. In contrast to the last subsection where we considered the co-occurence probability for a window, here, we use the Poisson approximation to compute a p-value for the number of CRMs. First, we describe the results using a non-overlapping window of size 100bp. Subsequently, we show the approximation using an overlapping window of size 100bp slided by 50bp steps and the derived error bounds.

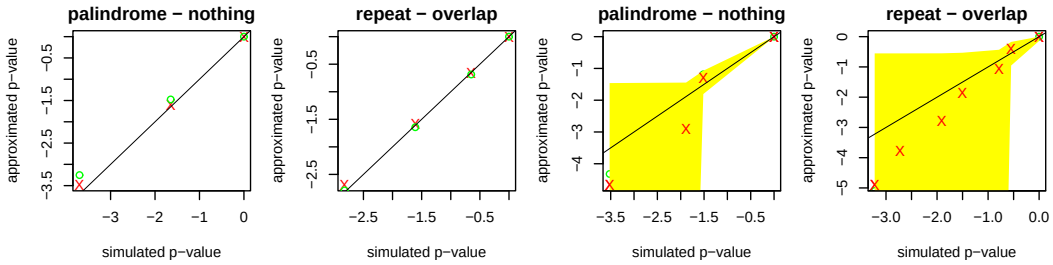

Figure 8.4: The figure shows the comparison of the log-transformed simulated p-values (x-axis) and the log-transformed approximated p-values (y-axis) for co-operativity of pairs of TFs. The left plot considers the two TFs 'palindrome' and 'nothing', the right part contains 'repeat' and 'overlap'. The upper row considers non-overlapping windows of size 100bp. The lower row shows overlapping windows of size 100bp slided by 50bp. The yellow area indicates the Chen-Stein bounds.

Figure 8.4 shows in the upper row a comparison of the log-transformed co-operativity p-values retrieved by the simulation and the new approach. The window size of 100bp yields a window probability of $p = 0.0002$. Although, the comparison for 'palindrome' and 'nothing' only contains co-operativity probabilities for observing 0,1, and 2 windows with at least one hit for A and B, the approximation works well since all points are almost on the diagonal. Using the same window size for the two TFs 'repeat' and 'overlap', the co-occurence probability becomes $p = 0.0025$. This is due to the similarity of 'repeat' and 'overlap'. It is not surprising that the simulation yields higher numbers of observed windows. Still, the approximation fits well to the simulated p-values. Furthermore, the p-value for at least two observed windows with co-occurences for TFs 'palindrome' and

'nothing' is about $2.5 \cdot 10^{-4}$ while for TFs 'repeat' and 'overlap' one needs to observe at least four CRMs for an equally low p-value. The results for the other pairs of TFs are shown in the Supplementary Material.

Overlapping Windows

Figure 8.4 shows in the lower row the log-transformed co-operativity p-values for two pairs of PFMs by using a sliding window approach with window size 100bp and step size 50bp. The figure also shows the error bounds from the Chen-Stein approximation as yellow background. If the computed p-value becomes orders of magnitude smaller than the bound, the upper bounds looks constant. If the p-values become smaller than the bound, the lower bound is equal to 0 which corresponds to $-\infty$ in log scale.

In the left part of the lower row of Fig. 8.4, the considered pair of TF contains the 'palindrome' and the 'nothing' motif. The approximated co-operativity p-values are smaller than the simulated p-values. Still, all of them are within the bounds given by the Chen-Stein approximation. Since we ignore the window dependencies in the approximation, the error is not surprising. The upper bound for high number of observed windows is 0.027. The approximated co-operativity p-values for the pair 'repeat' and 'overlap' in the right part of the lower row of Fig. 8.4 also underestimate the real p-value. Here, the upper bound is equal to 0.27. The reason for the higher bounds is the similarity between both TFs. Due to the similarity, the dependencies between the windows are stronger. Obviously, the Chen-Stein error bounds capture these dependencies. The results for the other pairs of TFs are shown in the Supplementary Material.

8.3 Discussion

We show that the co-occurence probability (at least one binding site for each of two TFs in a given window) is influenced by position dependencies. We derive a first-order approximation to consider the strongest dependencies. We can conclude from the comparison with simulations that the approximation is sufficient to obtain accurate results. Furthermore, in case of motifs with a higher background frequency of occurences, we show that it is necessary to incorporate this frequency into the background model. Results show that simple substitution of the theoretically computed frequency by the empirically retrieved frequency adjusts the approximation well. We also give results for the co-operativity p-value calculation for a pair of TFs. They show that we can accurately compute this p-value for non-overlapping windows. In the case of overlapping windows, the similarity between the TFs strongly influences the precision of the approximation. Using Chen-Stein error bounds, we can compute the precision of the approximation.

Using these results, one can conduct an analysis for co-occurences of TFs. In a first step, one would adjust the window size and the co-occurence probability for each pair of TFs. Using the described statistics, one can set the window size such that the probability for co-occurence within this window is reasonable small. If one decides to use overlapping windows, one can also compute the error bounds for the p-value approximation of co-occurences before running the analysis. After setting the window size such that the probability and the error are reasonable small, one runs the analysis and retrieves accurate p-value approximations

for the co-operativity of TFs. In fact, the choice of the co-occurence probability or the window size is rather robust since the Poisson approximation for the number of windows considers both as parameters to compute the p-value.

Chapter 9

Similarity of DNA Motifs

9.1 Introduction

Many computational tools deal with *ab initio* discovery of new PFMs on a set of related sequences (Tompa *et al.*, 2005). Since there is no best method, several programs are usually applied resulting in a redundant set of PFMs. Furthermore, the methods might discover PFMs similar to known PFMs. Therefore, either similar PFMs should be removed or merged in to a new PFM. Thus, an appropriate similarity measure for PFMs is required.

Most similarity measures consider PFMs as probability distibutions. Hence, the distance between the distributions is used as dissimilarity measure. Due to the position independence of PFMs, the comparison is done column-by-column which has been shown to work well (Liu *et al.*, 1990). The Pearson correlation coefficient which has been shown to be more effective than other methods (Pietrokovski, 1996) is widely used. Wang and Stormo (2003) describe the average log-likelihood ratio method. Schones *et al.* (2005) and Kielbasa *et al.* (2005) calculate the independence of the columns of two PFMs using the χ^2 statistic (Fleiss *et al.*, 2003). The Kullback-Leibler distance is also often used (Roepcke *et al.*, 2005; Aerts *et al.*, 2003). The Tomtom algorithm (Gupta *et al.*, 2007) can use any of these measures to compute a null distribution of similarity scores to obtain p-values. An additional measure described by Kielbasa *et al.* (2005) does not compute the distance between the PFM distributions but the correlation between the scores of the PFMs on a given sequence. The idea to correlate scores on random sequences is proposed in Liefooghe *et al.* (2006). However, they do not propose a method to summarize the correlations for different overlaps.

In spite of the wealth of literature on this topic, to date there is neither a 'natural' nor a general (model independent) definition of the similarity of two DNA motifs. Here we propose what we think is a natural similarity measure: Two PFMs should be regarded as similar when they describe similar binding sites. In this case, they obtain a high number of overlapping hits on a random sequence. Hence, the number of hits on the sequence is correlated between both PFMs. Considering the number of hits for a PFM on a random sequence as a random variable, the correlation is captured in the covariance between the random variables of two PFMs. We normalize the covariance by the sequence length and compute the asymptotic covariance for the sequence length approaching infinity. This measure does not depend on the PFM model. As long as a DNA motif model allows the calculation of the count distribution, one can also compute the asymptotic covariance - also with an instance of another DNA motif model. And, thus, compute the similarity. Furthermore, we introduce a related measure based on log-odd scores for the maximum overlap probability for the clustering, which is presented in Chapter 10.

The covariance approach is related to the score correlation methods (Kielbasa *et al.*, 2005; Liefooghe *et al.*, 2006): The covariance capturing the tendency of overlapping hits is derived using the two-dimensional joint score distributions for each possible overlap between both PFMs. The two dimensions of the joint distributions correspond to the score distributions of the two PFMs. On the one hand, the probability of an overlapping hit is the quantile of the joint score distribution with both scores greater or equal than the corresponding thresholds. On the other hand, the empirical score correlation method (Kielbasa *et al.*, 2005) approximates the correlation between both score distributions. A higher correlation of the scores is related to a higher joint probability of scores greater or equal than the thresholds. Therefore, all three approaches are based on similar ideas. However, the new approach presented here does not use an approximation for the score correlation as and it naturally summarizes the possible overlap positions by computing the covariance.

As mentioned above, Gupta *et al.* (2007) developed the Tomtom algorithm to compute the null distribution of similarity scores. Although we use a similar algorithm, we do not compute the null distribution of similarity scores but of motif scores based on the PFMs. Hence, we circumvent the arbitrary choice of a column-by-column similarity measure and, instead, we can use the covariance for summarization instead of a minimum p-value statistic.

We show the performance of the new approach by a simulation. We use the ratio of overlapping and non-overlapping hits in simulated sequences to obtain a similarity between pairs of PFMs. A generated PFM family is used for comparison with the χ^2 test which performed best in Schones *et al.* (2005), the Kullback-Leibler distance (Roepcke *et al.*, 2005; Aerts *et al.*, 2003), and the best Tomtom approach (Gupta *et al.*, 2007) using the euclidean distance. Since the exact Fisher-Irwin test (Bailey, 1977) is as good as the χ^2 test, we focussed on the latter one. Furthermore, we omit the score correlation (Kielbasa *et al.*, 2005) because its performance is similar to the χ^2 test (Kielbasa *et al.*, 2005). We also use a subset of Transfac (Matys *et al.*, 2003) PFMs and correlate the simulated similarities with our approaches.

The computation of the asymptotic covariance is described in Section 5.3.2. Hence, we skip its detailled presentation here, and shortly review alternative methods. We also describe the generation of the PFM family for the simulation. Finally, we present a comparison of the approaches and the performance on Transfac PFMs and discuss the impact of the results.

9.2 Methods

We define similarity between two PFMs by its asymptotic covariance. Hence, we obtain

$$S(A, B) := \lim_{n\to\infty} n^{-1}\, \mathbb{C}\mathrm{ov}\left[N_n(A) + N_n(A'), N_n(B) + N_n(B')\right]. \tag{9.1}$$

Its computation is described in detail in Section 5.3.2.

We also derive a second similarity measure based on the maximum overlap probability to perform clustering (see Chapter 10). For each possible shift d, we consider the ratio of the

overlap probability and the probability of two independent hits of A and B. Obviously, the denominator corresponds to the probability of two hits under a null model where A and B are independent. In contrast, the numerator contains the probability of two hits considering the dependencies between A and B. Applying the logarithm to the ratio yields log-odds scores:

$$S'_d(A,B) := \log\left(\frac{\gamma_d(A,B)}{\alpha_A \cdot \alpha_B}\right),$$

where $\gamma_d(A,B)$ denotes the joint probability of an occurrence of A and d positions later an occurrence of B. Taking the maximum over all shifts d and all pairs of A, A' and B, B', we can define the similarity measure $S^{\max}(A,B)$:

$$S^{\max}(A,B) := \max\left(\max_d S'_d(A,B), \max_d S'_d(A',B),\ \max_d S'_d(B,A), \max_d S'_d(B',A)\right).$$

Again, we are using certain equalities derived from the symmetric background model, in detail: $S'_d(A,B) = S'_d(A',B'), S'_d(A',B) = S'_d(A,B')$ and correspondingly for B followed by A.

Since we compare the new approaches with existing alternative approaches, we give a brief review of those in this subsection. The alternative approaches presented here are based on a column-by-column comparison. The course of action is the same for all approaches: In the first step, a score or p-value is obtained for each position for each possible shift/gapless alignment. In a second step, the scores/p-values for each position are summarized yielding a score/p-value for each shift. Finally, the score/p-value for all shifts are summarized to one final value.

As introduced in Schones *et al.* (2005), the χ^2 statistic is used to compute the probability whether two columns are drawn from the same multinomial distribution. Let m_{Xa} be the number of bases $a \in \mathfrak{A}$ for PFM X where $\mathfrak{A}$ denotes the alphabet. The marginal for PFM X is $m_X = \sum_{a\in\mathfrak{A}} m_{Xa}$. The nucleotide marginals for two PFMs X and Y are denoted by $m_a = m_{Xa} + m_{Ya}$. The overall number of counts is m^*. Denoting the observed number of counts with an upper index o and the expected number of counts by $m^e_{Xa} = m_X \cdot m_a/m^*$, we obtain a p-value using a χ^2 statistic with three degrees of freedom:

$$\sum_{a\in\mathfrak{A}}\left(\frac{(m^o_{Xa} - m^e_{Xa})^2}{m^e_{Xa}} + \frac{(m^o_{Ya} - m^e_{Ya})^2}{m^e_{Ya}}\right) \sim \chi^2_3$$

The p-values for all columns are summarized using the geometric mean. The final p-value is the minimum of the p-values for each shift.

The Kullback-Leibler distance (Kullback, 1959) is often used as a similarity measure in this context (Roepcke *et al.*, 2005; Aerts *et al.*, 2003). Using above notation the symmetric form is defined by:

$$\frac{1}{2}\left[\sum_{a\in\mathfrak{A}}\left(\frac{m^o_{Xa}}{m^o_X}\log\frac{m^o_{Xa}\cdot m^o_Y}{m^o_X\cdot m^o_{Ya}}+\frac{m^o_{Ya}}{m^o_Y}\log\frac{m^o_{Ya}\cdot m^o_X}{m^o_Y\cdot m^o_{Xa}}\right)\right]$$

The distances for the positions are summarized using the mean. The overall distance is computed by taking the maximum over all shifts.

The Tomtom algorithm (Gupta *et al.*, 2007) can use any column-by-column similarity measure. The authors show that the euclidean distance introduced in this area by Choi *et al.* (2004) performs best. The distance is defined by

$$-\sqrt{\sum_{a\in\mathfrak{A}}\left(\frac{m^o_{Xa}}{m^o_X}-\frac{m^o_{Ya}}{m^o_Y}\right)^2}.$$

The sum of the distances for all position are the so-called raw scores. The Tomtom algorithm approximates a null distribution of these raw scores to obtain a p-value. The p-values for all d shifts are summarized by computing the p-value for the smallest observed p-value p^* by $1-(1-p^*)^d$.

9.2.1 Data

In this section, we describe the simulation, the Transfac and Jaspar set of PFMs and their preprocessing. In a first step, we compute PSSMs from the PFMs by taking the log-likelihood ratio of the nucleotide frequencies of the binding site and the background model. To ensure strictly positive ratios one adds pseudocounts in a step called regularization. We add pseudocounts to the position specific distributions according to the information content of the position (Rahmann, 2003), for details see Section 2.4.1. We set the threshold by the type-I extended method (see Section 2.4.3).

Simulation

We compare the new similarity measures to existing approaches by using a simulation as reference: 10,000 sequences of length 10,000 are generated with an arbitrarily selected GC-content of 50%. After detecting all binding sites for a set of PFMs each with a threshold as defined earlier, we compute the number of overlapping hits N_{AB} between all pairs of TFs A and B. Based on these counts, we compute the simulated similarity as $\hat{S}(A,B)=N_{AB}/N_A$ where N_A denotes the number of hits of TF A. We get a symmetrical measure by using the average: $\hat{S}^{sym}(A,B)=(\hat{S}(A,B)+\hat{S}(B,A))/2$. In addition, we compare $S^{\max}$ with $\hat{S}^{\max}(A,B)=\max\{\hat{S}(A,B),\hat{S}(B,A)\}/2$.

The comparison is visualized by scatter plots for all pair-wise similarities. One dimension corresponds to the simulated similarity while the other dimension shows the computed similarity. We quantify the agreement between both measures using the Pearson correlation coefficient.

nr.	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	factor
1					A	C	G	T	A	C	G	T					1
2					A	C	G	T	A	C	G	T					10
3					A	C	G	T	A	C	G	T					.1
4					A	C	G	T	A	C	G	T					.01
5	*	*	*	*	A	C	G	T									1
6									A	C	G	T	*	*	*	*	1
7			*	*	A	C	G	T	A	C							1
8							G	T	A	C	G	T	*	*			1
9					A	C	G	T	A	C							1
10							G	T	A	C	G	T					1

Table 9.1: Dirichlet parameters for the generated PFMs. 'A' denotes $(30, 10, 10, 10)$, $(10, 30, 10, 10)$ for 'C', and correspondingly for 'G' and 'T'. $(1, 1, 1, 1)$ is labelled by '*'.

Sampling PFMs

We generate a family of PFMs where the members are gradually more similar to each other. We sample the PFM column by column using a Dirichlet distribution with different parameter sets (Schones *et al.*, 2005). The blueprint is the consensus sequence 'ACGTACGT'. We choose this sequence because it contains palindromic as well as repeat features. Such features are crucial for a realistic test setting since they determine the overlap probabilities. The count matrix is based on 60 sequences where 30 sequences have the consensus letter at each position and 10 sequences for each of the other nucleotides. The counts for each position serve as parameters for the Dirichlet distribution to sample the multinomial frequency distribution per position. Thus, we have one parameter set for consensus letter 'A': $(30, 10, 10, 10)$, one for 'C' $(10, 30, 10, 10)$, and so on for 'G' and 'T'. To modify the sharpness of the Dirichlet distribution (see Table 9.1). Furthermore, we shift the PFM relative to the consensus. In combination, we also reduced the length or added positions with samples from a Dirichlet distribution with non-informative parameters $(1, 1, 1, 1)$ denoted as '*' in the Table. In this manner, we sampled 10 PFMs. Table 9.1 shows the Dirichlet parameters for the 10 PFMs. Due to the palindromic structures, there are some matrices with the same Dirichlet parameter set. Still, the resulting PFMs are different because of the randomness in sampling. The resulting 10 PFMs are shown in Fig. 9.1 for sequence logos (Crooks *et al.*, 2004).

Transfac PFMs

As a further test set, we used a vertebrate subset of Transfac (Matys *et al.*, 2003) PFMs of version 11.1. We selected 279 of the 588 vertebrate PFMs due to the following filtering: The position specific nucleotide distributions for some PFMs are similar to the background distribution. In these cases, they cannot be used for binding site detection since the score for a binding site is not significantly higher than a score for a random sequence. Such PFMs can be selected by assessing the average information content per position and the power of a PFM (Rahmann *et al.*, 2003). Thus, PFMs are discarded if either they have an average information content less than 50% or a type II error β based on the balanced threshold greater than 15%. Instead of using the balanced threshold for sequence annotation, we always set α to 10%. Otherwise, very powerful PFMs have such a small α that hits occur rarely and, therefore, the simulation yields too few overlapping hits leading to bad estimates for the simulated similarity values.

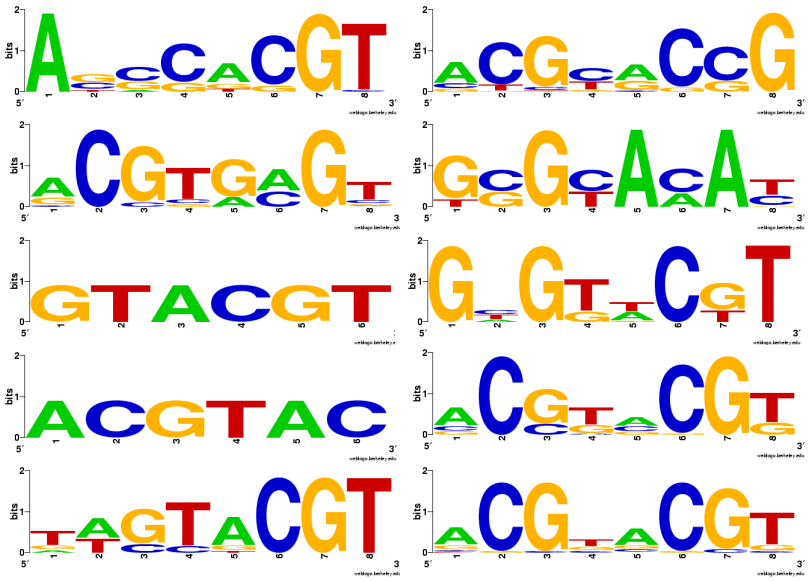

Figure 9.1: Sequence logos of generated PFMs used to compute similarities.

Sequences

The similarity for the Transfac PFMs is computed for two sets of sequences: random sequences and human promoter sequences. The random sequences are generated as above but with an average GC content equal to the human promoter sequences (44.86%). The human promoter sequences are based on Ensembl v46 (Hubbard *et al.*, 2005). For each Ensembl ID, we take the sequence region -10,000 to +200 relative to the transcription start site. If this sequence overlaps with another Ensembl gene entry, we cut the sequence at that position.

9.3 Results

9.3.1 Comparison with alternative approaches

In this article, we propose two new measures for similarity between PFMs. The first measure S is the asymptotic covariance between the number of hits of two TFs. For the purpose of clustering, we introduced the related measure $S^{\max}$ which computes the maximum log-odds score for the overlap probabilities. Figure 9.2 compares the new and three alternative measures with the measure computed by simulations. In Fig. 9.2A the χ^2 test is compared with simulation. One observes a rough correlation although the highest simulated pair-wise similarity has a χ^2 similarity of 0. The Kullback-Leibler distance is shown in Fig. 9.2B. Of course, the pairs with high Kullback-Leibler distance correspond to low simulated similarities. The measure can be used to separate similar and dissimilar pairs of TFs without too many false positives, e.g. with a low cut-off of 0.5. In addition, visualization of the rank transformed values shows that a rough ordering of similar pairs is possible (see Fig. 9.3). The Tomtom approach based on the euclidean distance (see Fig. 9.2C) shows a similar

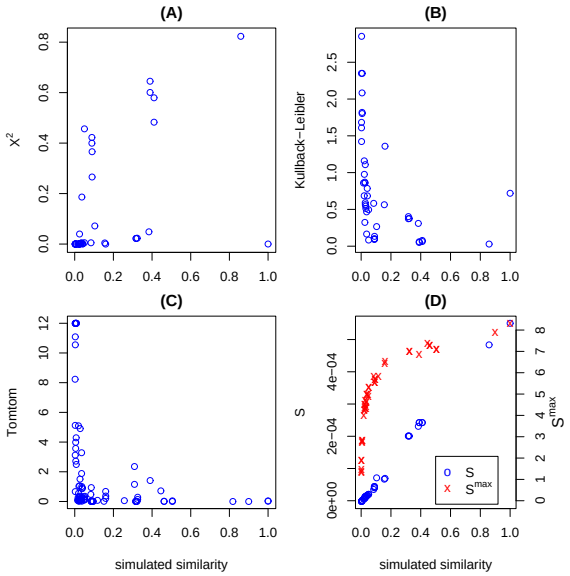

Figure 9.2: Scatter Plot of all pair-wise similarities for the simulation (x-axis) and the calculated similarity (y-axis). (A) contains χ^2, (B) the Kullback-Leibler distance, (C) the Tomtom result using the euclidean distance, and (D) consists of the asymptotic covariance S (blue circles, left axis) and $S^{\max}$ (red crosses, right axis).

behaviour. In general, more pairs with high simulated similarity have a very small computed similarity. In contrast, the asymptotic covariance in Fig. 9.2D denoted by S shows a strong linear correlation. There are no crucial disagreements between the simulation and the computation. The measure $S^{\max}$ which only captures the highest similarity grows with the simulated values but flattens for high values. Since we only consider the maximum overlap probability, differentiation between highly similar PFMs becomes more difficult. Still, an ordering is possible also for these values as shown in the rank transformed plots in Fig. 9.3.

A quantitive comparison value is given by the Pearson coefficient for the correlation between the simulated measure and the computed similarity measure. We obtain a Pearson coefficient of 0.509 (0.786 after rank transformation) for the χ^2 measure. The Kullback-Leibler distance, which is a distance instead of a similarity, has a negative correlation coefficient of -0.402 (-0.803). Although in this case, the Pearson correlation is a bad measure since the regression line is perpendicular to the x-axis (see Fig. 9.2B). The distance from the Euclidean Tomtom approach has a small correlation coefficient of -0.292 (-0.674). The asymptotic covariance shows a strong linear correlation with a Pearson correlation coefficient of 0.997 (0.993). The measure $S^{\max}$ yields a correlation coefficient of 0.76 (0.986).

9.3.2 Transfac Set

Figure 9.4A shows the analysis on simulated sequences for the pairs of 279 Transfac PFMs. The asymptotic covariance has a strong linear correlation while, again, $S^{\max}$ flattens for higher similarity values. The analysis for human promoter sequences (Fig. 9.4) is similar but in general more scattered. The Pearson coefficient for S on simulated sequences is

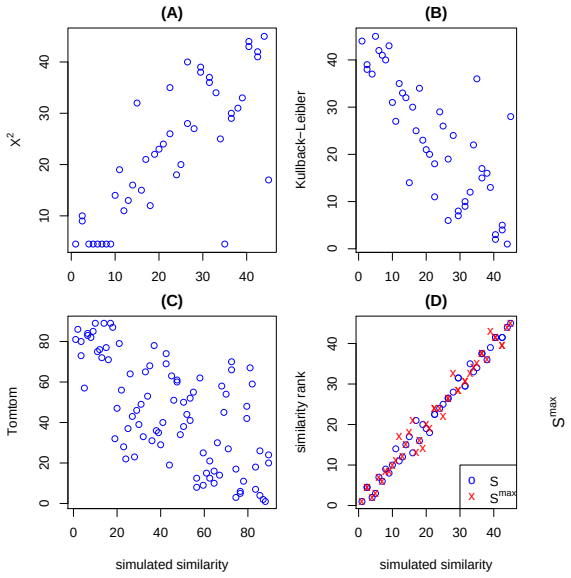

Figure 9.3: Scatter Plot of all ranked pair-wise similarities for the simulation (x-axis) and the calculated similarity (y-axis). (A) contains χ^2, (B) the Kullback-Leibler distance, (C) the Tomtom result using the euclidean distance, and (D) consists of the asymptotic covariance S (blue circles, left axis) and $S^{\max}$ (red crosses, right axis).

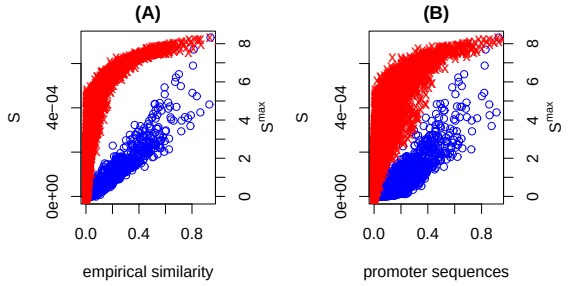

Figure 9.4: Scatter Plot of each pair-wise similarity of Transfac PFMs between S (blue circles) and $S^{\max}$ (red crosses) based on (A) simulated sequences and (B) human promoter sequences.

0.952. The maximum measure yields a Pearson coefficient of 0.615. The Pearson coefficient for the human promoter sequences for S is smaller (0.886) than for simulated sequences. In constrast, the Pearson coefficient increases for the maximum measure to 0.665. This is mainly due to the fact that the correlation is non-linear and the correlation coefficient is supported by the higher variance for low similarity values.

9.4 Discussion

We have introduced two new measures of similarity for PFMs. One main difference is that these new similarity measures depend on the regularization method, the parameter which represents the threshold to detect a hit, and on the background model. To remove redundancies in a set of PFMs, we consider this an advantage because the detected binding

sites in a sequence also depend on these parameters. In fact, the similarity measure is able to capture the differences between the results of two different parameter sets. As an extreme example consider two different PFMs with the same length and both with a threshold such that all words are accepted. Due to their co-occurring hits, these PFMs have the highest similarity. Increasing the threshold decreases the number of words with scores higher than the threshold. Therefore, similarity should decrease as it does in the new approaches. The background model also can have a high influence on the results. For example, two overlapping PFMs without CpG dinucleotides are very similar within a CpG island because both differ from the background. In a CpG poor background model, both PFMs are hidden within the background, thus, neither they achieve a high similarity nor are their hits correlated. Again, this is advantageous for removing redundant PFMs in a set. Furthermore, extending the similarity measure for higher order Markov models is possible although calculation of the 2-dimensional score distribution will become time consuming.

In contrast to the advantageous effect on the removal of redundancies, the dependence of the result on the parameter choices is unwanted for clustering. Although the clustering is robust against small changes, of course, big differences in the parameters do change the result. For example, changing the GC content to 40% changes the composition of four clusters by 1-2 insertions/deletions per cluster and adds a new small cluster of size two. Instead, substituting the regularization method by a simple addition of pseudocounts (0.01) only has a minor effect by changing the composition of one cluster slightly.

The analysis considering 279 Transfac PFMs shows that the similarity measure is not only applicable to artificial PFMs but also to real binding sites. The comparison between simulated sequences and human promoter sequences shows that for no pair of TFs the simulated similarity significantly differs from the theoretical similarity. Large differences, e.g. high simulated similarity and low theoretical similarity, might give evidence for competitive binding due to more overlapping binding sites than expected by chance. Since we do not observe such deviations, either signal to noise ratio is too low or competitive binding sites evolve to be similar regarding their sequence.

In this chapter, we have introduced two new measures of similarity. In contrast to existing measures, we give a natural and general interpretation of the similarity, which is especially useful in practice. We use a statistical framework to derive the measure, resulting in the asymptotic covariance. Of course, the measures can also be applied to arbitrary set of words, i.e. experimentally verified binding sites, although computation becomes inefficient for large sets. In Chapter 11, we apply this measure to compute the quality of representation of PFMs.

Chapter 10

Clustering of PFMs

10.1 Introduction

Suzuki and Yagi (1994) show that TFs of the same family share similarity in the binding profile. The Familial Binding Profile (FBP) is a generalized binding profile capturing this core motif of a family of TFs (Sandelin and Wasserman, 2004). Several approaches perform clustering of TFs into families based on a Bayesian learning algorithm (Narlikar and Hartemink, 2006) and unsupervised neural network (Mahony *et al.*, 2005). Others use as a metrics ungapped local motif alignments (Sandelin and Wasserman, 2004) and similarity measures (Kielbasa *et al.*, 2005). A comparison of DNA sequence based approaches is presented by Mahony *et al.* (2007).

In Chapter 9, we introduced a second similarity measure based on the maximum overlap probability. Since this measure automatically returns the position with the highest overlap probability, we obtain a gapless alignment of the two PFMs. Hence, we can merge the PFMs. Therefore, we develop a clustering algorithm for PFMs, apply it on a set of class labelled Jaspar (Sandelin *et al.*, 2004b) PFMs and compare the class of the members for each cluster. We also automatically obtain familial binding profiles for each cluster. The results are compared with the ones from Mahony *et al.* (2007).

10.2 Algorithm

In this section, we present a clustering approach which yields an FBP for each cluster and which discards TFs which do not have any sufficient similarity. The clustering consists of three main steps:

1. Selection: Select the pair with maximum similarity.
2. Merging: Create the new FBP for the cluster.
3. Verification: Discard the new cluster if not all members share sufficient similarity.

In the following, we describe each step in more detail. Let $\mathcal{T} = \{T_i\}$ be the set of TFs. The goal is to compute a set $\mathcal{C}$ of disjoint classes $C_j \subseteq \mathcal{T}$. The FBP/representative of class j is given by $r(C_j)$ while $r(\mathcal{C})$ yields the set of all FBPs. Furthermore, $c(\cdot)$ returns the class index of a (meta) TF. We initialize the set of classes $\mathcal{C} = \mathcal{T}$ to one class for each TF such that $c(T_i) = i$ and $r(C_i) = T_i$.

Selection

This step selects pairs $X, Y \in r(\mathcal{C})$ which have highest similarity. Here, we use the derived maximum similarity measure $S^{\max}(X, Y)$ (see Section 9.2). This measure also returns the position d^* with the maximum overlap. We select X^*, Y^* such that

$$(X^*, Y^*) = \text{argmax}_{(X,Y) \in r(\mathcal{C}) \times r(\mathcal{C}), X \neq Y} S^{\max}(X, Y).$$

Furthermore, we introduce a threshold q to stop clustering if the similarity is not sufficiently high. We set the parameter q equal to the 95% quantile of all pairwise $S^{\max}$ values from the initial set $\mathcal{T}$.

Merging

After selecting one pair of TFs, we create the new FBP W: Let d^* denote the position of maximum overlap probability. The new FBP consists of the sum of the position count matrices of X and Y shifted by d^* positions. Thus, the length of W is $\ell_W = \ell_A + \ell_B - d^*$. If the maximum similarity occurs for X' or Y', we transform the respective position count matrices accordingly before summation. Before summation, we enlarge both position count matrices such that they overlap for each position of the FBP. The enlarged positions are filled with the background model. It is based on the background frequencies $\mu(a)$ with $a \in \mathfrak{A}$. Since we sum the position count matrices, we have to obtain counts. Therefore, we multiply $\mu(a)$ by the average number of sequences of the corresponding position count matrix. This automatically corrects the information content of positions which do not overlap with all members of a cluster by adding the corresponding fraction of the background distribution. In other words, we take into account the number of motifs without specific signal at these positions.

Verification

Due to the naive merging of TFs, the FBP might get less and less related to its original members after successive mergings. Hence, the clusters are no longer homogeneous but become more and more heterogeneous over the number of clustering steps. To prevent this, we ensure that the new FBP always has a high similarity to each of its members. The merge of the pair X and Y is discarded if any of the following inequalities does not hold:

$$\forall_{V \in C_{c(X)}, W \in C_{c(Y)}} \quad : \quad S^{\max}(V, W) > q$$

In case all inequalities hold, we update $\mathcal{C}$ by removing X and Y and adding the new cluster with its FBP W and members $C_{c(X)}$ and $C_{c(Y)}$. If at least one inequality does not hold, we skip the merging and mark the pair X and Y such that they cannot be merged. Then, the procedure starts with the selection step, again. The three steps are iterated until no non-marked pair of TFs has a similarity greater than q.

10.3 Data

The clustering is based on Jaspar (Sandelin *et al.*, 2004b) PFMs using the data set analyzed by Mahony *et al.* (2007). The set consists of 13 classes each with closely related members. The classes are bZIP cEBP (4 members), bZIP CREB (4), bHLH (10), ETS (7), Forkhead (8), high mobility group (HMG: 6), HOMEO (8), MADS (5), NUCLEAR (8), REL (6), TRP (5), zinc finger DOF (4), and zinc finger GATA (4).

We assess the consistency of the clusters using the leave-one-out-cross-validation (LOOCV) approach following Sandelin *et al.* (2004b) and Mahony *et al.* (2007): For each PFM except the singletons, we remove its contribution to the corresponding FBP. Then, we compute the similarity between the PFM and all FBPs and singletons. If the similarity between the PFM and its corresponding (modified) FBP is maximal, we call it a correct classification. A high percentage of correct classifications suggests a consistent clustering. In contrast to Mahony *et al.* (2007), we do not count singletons as misclassifications. Otherwise, more singletons in the clustering automatically lead to a lower consistency although more homogeneous clusters might have been retrieved. This occurs as soon as some PFMs only share weak similarity with all other PFMs. Hence, we include singletons as FBPs for classifying although we do not classify the singletons. Accordingly, we decrease the total number of classifications to compute the success rate.

10.4 Results

The Jaspar set contains classes of closely related TFs. The clustering of the 13 classes with a total of 79 PFMs yields 14 clusters. Three of four bZIP EBP PFMs are contained in the clustering, forming one homogeneous cluster. All four bZIP CREBs also form one homogeneous cluster. Eight of eleven bHLH are clustered forming two homogeneous clusters (size three and five). All seven ETS factors belong to one homogeneous cluster. All six Forkhead PFMs belong to one cluster which also contains four HMGs in a separate branch. One of the remaining two HMGs is clustered in a small cluster with one HOMEO. Five of the remaining seven HOMEOs are in one homogeneous cluster, as well, as all five MADS PFMs. Seven of eight NUCLEAR receptors are contained in one homogeneous cluster. All six RELs belong to one homogeneous cluster. Two of the five TRPs are contained in a heterogeneous cluster with all four zinc finger DOFs, two of the remaining three TRPs are also clustered together homogeneously. Finally, two of the four zinc finger GATAs are forming one homogeneous cluster. Altogether, eleven of the 14 clusters are homogeneous, containing 49 of 67 PFMs while twelve PFMs are not clustered at all.

We compare our clusters (including the eight zinc fingers) with the corresponding results from the clustering in Mahony *et al.* (2007) based on an ungapped Smith-Waterman alignment with the Pearson correlation coefficient as scoring function. This clustering from Mahony *et al.* (2007) including the eight zinc fingers is very similar to the clustering without the zinc fingers in Figure 8 (Mahony *et al.*, 2007) but yields 16 clusters and two singletons (personal communication). The subtle differences are considered below. We yield seven times the same clusters: ETS, Nuclear Receptor, bZIP CREB Subgroup, bZIP cEBP Subgroup, MADS, HOMEO Subgroup, and the TRP-cluster/IRF Subgroup with zinc finger DOFs. Another five clusters are modified: The REL-like group becomes a homogeneous cluster

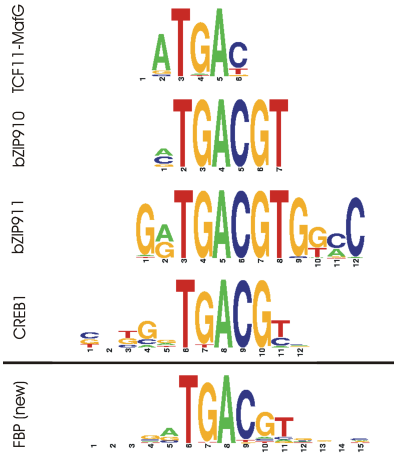

Figure 10.1: FBP of the bZIP CREB cluster with the multiple alignment containing the four TFs TCF11-MafG, bZIP910, bZIP911, and CREB1.

since En1 from the HOMEO group and Chop-cEBP from the bZIP group are removed. The bHLH-ZIP cluster does not contain the correct member Arnt-Ahr any more. The TRP-cluster/Myb Subgroup lacks the correct member MYB-ph3. The HMG/Forkhead Group 1 does not contain the wrong member Pbx from the HOMEO group. Our HMG/HOMEO mix contains different TFs from the same classes HMG and HOMEO, specifically HMG-1 and En1. We also obtain a cluster for the zinc finger GATA PFMs but only containing two in comparison to three in Mahony *et al.* (2007). Instead of the mixed cluster including the zinc finger GATA1, the bHLH TAL1-TCF3 and the Forkhead FOXL1, we obtain an extended bHLH Subgroup cluster with TAL1-TCF3 and the two other bHLHs NHLH1 and MYF. The two latter ones are clustered by Mahony *et al.* (2007) into a single cluster. In fact, all three members share the consensus CA*CTG justifying the extension of the cluster. The heterogeneous cluster HMG/Forkhead Group 2 with two members does not appear in our analysis.

Furthermore, we performed the LOOCV test on our clustering. All 67 PFMs are classified correctly while excluding the twelve singletons. The high number of correct classifications is not surprising since the clustering algorithm intrinsically computes a consistent clustering by testing in each step the similarity between all members and their respective FBP. In comparison, Mahony *et al.* (2007) obtain 72 of 77 correct classifications likewise without counting the two singletons as wrong classifications. Hence, Mahony *et al.* (2007) assign more PFMs to clusters while increasing the number of heterogeneous clusters and decreasing the consistency of the clustering. Instead, we obtain more singletons leading to a more stringent and more consistent clustering.

The clustering automatically generates an FBP for each cluster. All FBPs are given in the Supplementary Material. As an example, the FBP for the bZIP CREBs is printed here in Figure 10.1. In comparison to the FBP of the same cluster in Mahony *et al.* (2007), we obtain the reverse complementary sequence logo. Removing the flanking non-informative sites, we are left with a strong consensus 'TGAC' followed by a weaker consensus of 'GT'. This matches with the FBP in Figure 1 of Mahony *et al.* (2007). However, there the last two letters have a higher information content than in our clustering approach. Since we au-

tomatically consider the number of supporting PFMs per position by using the background model for non-overlapping positions in each merging step, the corresponding positions do not have high information content. Thus, it is the more adequate representation.

10.5 Discussion

The clustering of the Jaspar set yields a high number of homogeneous clusters. In addition, only a minor fraction of PFMs are not clustered at all. Hence, it seems that, indeed, the similarity between PFMs is captured appropriately. Furthermore, the clustering yields a FBP/representative for each cluster containing the characteristic properties of its class members.

Chapter 11

Quality of Representation

11.1 Introduction

A PFM is constructed from a multiple sequence alignment of verified binding sites. Their sequences are discovered by wet-lab experiments (for a review, see Elnitski *et al.*, 2006; Maston *et al.*, 2006). The PFM model represents the binding sites in a compact way, retaining only the frequency of each nucleotide at each position; this *statistical independence of positions* (loss of dependencies between positions) is the basic assumption of the PFM model.

Unfortunately, not all TF binding site motifs fulfill the position independence assumption (Benos *et al.*, 2001; Bulyk *et al.*, 2002); such motifs cannot be well represented by PFMs. More complex models have been developed; they range from PFMs with multi-nucleotides per position (Brendel and Trifonov, 1984; Stormo *et al.*, 1986; Zhang and Marr, 1993) over mixture models (Barash *et al.*, 2003; Hannenhalli and Wang, 2005; Georgi and Schliep, 2006) to variable length permuted Markov models (Zhao *et al.*, 2005) and Bayesian Networks (Barash *et al.*, 2003). Although some of these models implicitly estimate the required complexity (Barash *et al.*, 2003; Georgi and Schliep, 2006), they only compare the more complex model with the simpler (PFM) model, but do not assess the representation quality of either model in absolute terms.

To our knowledge, there is no standard method that assesses how well a binding site model represents the verified binding sites. Here, we derive a general *representation quality measure* that can be efficiently computed for PFMs, and discuss its properties.

Apparently, a reasonable idea is to directly compare the set of words compatible with the model (for a PFM, the words that reach the score threshold) to the set of verified binding site sequences. Since all of these are assumed to be of the same length ℓ, we can classify each length-ℓ DNA word as true positive (TP), false positive (FP; compatible with the model, but not a verified binding site), true negative (TN), or false negative (FN; a verified binding site, but not compatible with the model). This leads to three problems:

1. Even for a model as simple as PFMs, the calculation of the set of compatible words is NP-hard in general (Touzet and Varré, 2007; Zhang *et al.*, 2007).

2. Listing the frequency of FPs or FNs does not take multiplicities in the set of verified binding sites into account.

3. Although one can summarize FP and FN frequencies into quantities such as sensitivity, specificity and precision, one would still obtain three different quality indicators instead of one.

We solve these problems by proposing a different measure that directly describes what binding site models are used for: their ability to recognize the sequences identical to the verified binding sites in long genomic sequences. Therefore we measure representation quality by the correlation between two random variables:

1. $N_n(A)$, the number of occurrences of words compatible with the model A in a random i.i.d. sequence $\mathbf{X}$ of length n,

2. $N_n(\mathcal{W})$, the number of occurrences of verified binding site sequences $\mathcal{W}$ in the same sequence $\mathbf{X}$.

To obtain a quantity that is independent of the sequence length n, we use the *asymptotic correlation* with $n \to \infty$. If the set of compatible words equals the set of verified binding sites, the correlation attains its maximum value 1.

The main computational issue is the calculation of the *asymptotic covariance* between $N_n(A)$ and $N_n(\mathcal{W})$ without enumerating the compatible words of A. For PFMs, this problem can be efficiently solved by extending the similarity measure for PFMs (see Chapter 9).

As an example, consider the four hypothetical 'experimentally verified' binding site sequences 'AAAC', 'ACAC', 'AACC' and 'ACCC'. The two center positions can vary between 'A' and 'C', while the first and last positions are fixed. One would obtain a PSM qualitatively similar to

$$\begin{array}{c} \text{'A'} \\ \text{'C'} \\ \text{'G'} \\ \text{'T'} \end{array} \begin{pmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Setting the threshold to 6 leads to four compatible words, namely the given binding sites. Thus, the PFM/PSM is able to represent the given sequences perfectly, without any false positives and false negatives. Unfortunately, this is an exception.

If we consider four different 'verified' binding site sequences, twice 'AACC' and twice 'ACAC', we obtain the same PSM as above. However, there is no threshold such that the set of compatible words exactly equals the given binding sites. The reason is the position dependency between the second and the third position. If one of positions 2 or 3 contains an 'A', the other one must be a 'C'. The PFM model is not capable of representing such dependencies; in such a case we expect a low (representation) quality value.

In the Methods section, we first introduce the *representation quality* in a general manner, comparing it to other quality measures (sensitivity, specificity, precision). In the case of PFMs, we derive the formulae to compute the asymptotic correlation and briefly explain how it can be computed efficiently.

In the Results section, we argue that the PFM score threshold should be chosen to maximize representation quality. We apply the approach to 101 Transfac PFMs (Matys *et al.*, 2003) and 123 Jaspar PFMs (Sandelin *et al.*, 2004b). To illustrate that PFM representation quality indeed captures position dependencies, we compare the quality value of binding site models that have been previously classified as one- or two-component mixture models (Georgi and Schliep, 2006), and find that the two groups are well separated.

A concluding Discussion summarizes the key features.

11.2 Methods

11.2.1 Sensitivity, Specificity and Precision

For the moment, we assume that we are given a set $\mathcal{W}$ of experimentally verified binding sites (below we show how to extend this to a multi-set, where the same sequence may occur multiple times), and a binding site model A (e.g., a PFM) that implicitly defines a set of compatible words, which may depend on one or several parameters (e.g., the score threshold).

Consider the typical situation that model A is used to scan a long genomic sequence for occurrences of binding sites. The model A assigns a score to every position of the given DNA sequence. There is a *hit of A at position i* if the word that starts at the i-th position of the DNA sequence is compatible with A.

We define the indicator random variable $Y_i(a)$ as $Y_i(a) := 1$ if the word a starts at position i, and $Y_i(a) := 0$ otherwise. Denoting the set of compatible words for A by $\mathcal{A}$, we define an indicator random variable for a hit at position i by $Y_i(A) := Y_i(\mathcal{A}) := \sum_{a\in\mathcal{A}} Y_i(a)$. Similarly, we define $Y_i(\mathcal{W})$ to indicate a hit of the set $\mathcal{W}$ at position i.

Then the events TP, FP, TN, FN (at position i) can be formally written as

$$\begin{aligned}
\mathrm{TP} &:= \{(Y_i(A) = 1, Y_i(\mathcal{W}) = 1)\},\\
\mathrm{FP} &:= \{(Y_i(A) = 1, Y_i(\mathcal{W}) = 0)\},\\
\mathrm{TN} &:= \{(Y_i(A) = 0, Y_i(\mathcal{W}) = 0)\},\\
\mathrm{FN} &:= \{(Y_i(A) = 0, Y_i(\mathcal{W}) = 1)\}.
\end{aligned}$$

The probabilities of these events in a random i.i.d. sequence obviously do not depend on the position i.

Now the *sensitivity* of A is defined by $\mathbb{P}_{H_0}(\mathrm{TP})/(\mathbb{P}_{H_0}(\mathrm{TP}) + \mathbb{P}_{H_0}(\mathrm{FN}))$, the *specificity* by $\mathbb{P}_{H_0}(\mathrm{TN})/(\mathbb{P}_{H_0}(\mathrm{FP}) + \mathbb{P}_{H_0}(\mathrm{TN}))$, and the *precision* by $\mathbb{P}_{H_0}(\mathrm{TP})/(\mathbb{P}_{H_0}(\mathrm{TP}) + \mathbb{P}_{H_0}(\mathrm{FP}))$. We can derive following equations:

$$\begin{aligned}
\text{Sensitivity} &= \frac{\mathbb{P}_{H_0}(Y_i(A)=1, Y_i(\mathcal{W})=1)}{\mathbb{P}_{H_0}(Y_i(\mathcal{W})=1)} \\
&= \frac{\sum_{w\in\mathcal{W}} \mathbb{P}_{H_0}(Y_i(A)=1, Y_i(w)=1)}{\sum_{w\in\mathcal{W}} \mathbb{P}_{H_0}(Y_i(w)=1)}, \\
\text{Specificity} &= \frac{\mathbb{P}_{H_0}(Y_i(A)=0, Y_i(\mathcal{W})=0)}{1-\sum_{w\in\mathcal{W}} \mathbb{P}_{H_0}(Y_i(w)=1)}, \\
\text{Precision} &= \frac{\mathbb{P}_{H_0}(Y_i(A)=1, Y_i(\mathcal{W})=1)}{\mathbb{P}_{H_0}(Y_i(A)=1)}.
\end{aligned}$$

These probabilities can be computed efficiently using a two dimensional score convolution of PSMs (see Section 5.6) after transforming $\mathcal{W}$ to a set of PSMs (see Algorithm section).

Note that in contrast to Rahmann (2003), we define the sensitivity and specificity in terms of a given verified set $\mathcal{W}$, while Rahmann (2003) defines them in terms of the *power* of the model, i.e., its ability to distinguish between the nucleotide distribution of the model and the background distribution. The approach taken here appears more relevant in practice, since a model not supported by any experimental evidence is not useful in practice.

11.2.2 Representation Quality

We compute how well a model A with compatible word set $\mathcal{A}$ represents a set of experimentally verified binding sites $\mathcal{W}$ of size $m := |\mathcal{W}|$. Instead of directly comparing $\mathcal{W}$ with $\mathcal{A}$ (enumeration of the words in $\mathcal{A}$ is NP-hard for PFMs, see Section 2.4.4), we compare the number of occurrences of words from $\mathcal{A}$ and $\mathcal{W}$ in random texts following a specified i.i.d. null model μ.

Using the position-wise indicator random variables Y_i, the number of hits for model A and set $\mathcal{W}$ in a sequence with n potential starting positions can be expressed as the random variables $N_n(A) := \sum_{i=1}^{n-\ell+1} Y_i(A)$, and $N_n(\mathcal{W})$ defined similarly. The correlation between these two random variables is

$$\mathbb{C}\text{or}(N_n(A), N_n(\mathcal{W})) = \frac{\mathbb{C}\text{ov}(N_n(A), N_n(\mathcal{W}))}{\sqrt{\mathbb{V}(N_n(A)) \cdot \mathbb{V}(N_n(\mathcal{W}))}},$$

where $\mathbb{C}\text{ov}(\cdot,\cdot)$ denotes covariance. To obtain a measure independent of the sequence length n, we define *representation quality* $Q(A \mid \mathcal{W})$ as the asymptotic value of the correlation as $n \to \infty$:

$$Q(A \mid \mathcal{W}) := \lim_{n\to\infty} \mathbb{C}\text{or}(N_n(A), N_n(\mathcal{W})). \tag{11.1}$$

The representation quality has a value in $[-1, 1]$. The maximum is achieved if and only if $\mathcal{W} = \mathcal{A}$. The minimum usually does not occur since the construction of the model ensures that at least a subset of the compatible words is recognized by the model.

11.2.3 Multiplicities in the Set of Verified Binding Sites

So far, we have assumed that each words in the set $\mathcal{W}$ of experimentally verified sequences appears only once in the set. In general, binding sites with the same sequence can be observed multiple times during experiments. For example, consider a DNA motif with 100 verified binding sites, where 99 sites consist of the same sequence.

Under the assumption that a higher number of observed instances reflects higher binding affinity, the representation quality should also be higher if a multiple observed word belongs to the set of compatible words. We can therefore more generally assume that each word in $\mathcal{W} = \{W_1, \ldots, W_j, \ldots, W_m\}$ has a weight ϕ_j associated to it. Incorporating weights into the correlation is straight-forward, since the correlation is bilinear and $N_n(\mathcal{W})$ can be written as the sum of the contributions of each (weighted) word:

$$N_n(\mathcal{W}) = \sum_{j=1}^{m} \phi_j \, N_n(W_j);$$

$$\mathbb{C}\mathrm{or}(N_n(A), N_n(\mathcal{W})) := \frac{\sum_{j=1}^{m} \phi_j \, \mathbb{C}\mathrm{ov}(N_n(A), N_n(W_j))}{\sqrt{\mathbb{V}(N_n(A)) \cdot \phi_j^2 \, \sum_{j=1}^{m} \mathbb{V}(N_n(W_j))}}.$$

The correlation remains in the interval $[-1, 1]$. In contrast, it is not straight-forward to apply multiplicities to sensitivity, specificity, or the precision, since these are directly defined via probabilities of events.

11.2.4 Algorithm for PFMs

So far, we have defined representation quality for general binding site models A with respect to a given set $\mathcal{W}$ but have not mentioned how to efficiently compute $Q(A \mid \mathcal{W})$. The complexity of this task depends on the model. Here we focus on PFMs where an efficient computation is possible. Let us specify all steps, starting with PFM construction.

A PFM represents a DNA motif by specifying the probability (relative frequency) of each nucleotide at every position. To ensure nonzero probabilities (nothing should be impossible, merely very improbable), one adds pseudo-counts in a step called regularization. We add pseudo-counts to the position specific distributions according to the information content of the position (Rahmann, 2003): Positions with low information content are shifted towards the background (null) distribution μ (overall nucleotide frequencies in the genome). For positions with high information content, the difference to the background distributions is emphasized.

For a PFM A, the position specific scoring matrix (PSM) is the log-likelihood ratio (here discretized with a granularity of 0.05) of the nucleotide distribution of the PFM and the background probability μ_a for every position and nucleotide $a \in \mathfrak{A}$. The background model μ is an i.i.d. model defined by the genomic or local GC content to make the computation invariant with respect to the choice of the leading strand. A word is compatible with A if its score, summed over all positions, reaches a given threshold t_A, which is a parameter of the model A.

Although the definition of the hit indicator $Y_i(A)$ is based on the set of compatible words, for PFMs it is sufficient to compute the score at each position: $Y_i(A) = 1$ if and only if the word starting at position i reaches the score threshold t_A. This allows an efficient computation of the covariance and hence correlation.

We adapt the idea of Section 5.3.2 where we show how to efficiently compute the asymptotic covariance between the number of occurrences of two PFMs/PSMs to quantify their similarity. The only technical difference is that we compare a PFM with an arbitrary given set $\mathcal{W}$. Since for weighting purposes (see above), we need to decompose $\mathcal{W}$ into m singleton sets anyway, we interpret these singletons as compatible sets of particular PFMs constructed as follows.

To obtain a PSM with compatible set $\{w\}$, we define the score for the nucleotide w_κ occurring at position κ in $w = w_1 w_2 \dots w_\ell$ to be 1, and for all other nucleotides to be 0. We can also give a definition incorporating ambiguous letters (IUPAC code) in w:

$$Score_{\kappa,a}^{(w)} := \begin{cases} 1 & \text{if} \quad a \in f(w_\kappa), \\ 0 & \text{otherwise.} \end{cases}$$

Here $f(a)$ describes the set of nucleotides associated to the possibly ambiguous letter a, e.g., $f(A) = \{A\}$, $f(Y) = \{C, T\}$. The threshold t_w is defined by $t_w := \ell$ since the score is increased at every position by 1.

The transformation of the words w to PSMs is not conceptually necessary but technically convenient, allows easy incorporation of weights, and directly permits using the implementation from Section 5.6.

11.2.5 Threshold and Regularization Optimization

Obviously, the sensitivity, specificity, precision, and representation quality all depend on the parameters of the model A. For PFMs, it is primarily the threshold t_A that defines the (size of) the set of compatible words. A low threshold leads to a set with many compatible words that probably contains many false positives. A high threshold leads to few compatible words, and some of the verified binding sites may not be included (false negatives).

Commonly, the threshold is chosen such that the probability for a type-I error (a hit under the null model) is bounded by or equal to a pre-determined value (Rahmann *et al.*, 2003; Liefooghe *et al.*, 2006). Here, we consider the type-I error on a random i.i.d. sequence of length 500. A more sophisticated method sets the threshold such that the type-I and the type-II (no hit under the binding site model) errors are balanced (Rahmann *et al.*, 2003). However, a choice based on purely statistical considerations without considering the verified set $\mathcal{W}$ is generally unable to represent $\mathcal{W}$ well. Instead, one can optimize the threshold t_A such that the sensitivity, specificity, precision, or representation quality is maximized. In Section 11.3.1, we argue that maximizing the quality is the most natural choice.

Additionally, note that the PFM model depends on the chosen regularization method. Thus, one could also assess different regularization methods by their gain of representation quality. Here, we do not explore this possibility further.

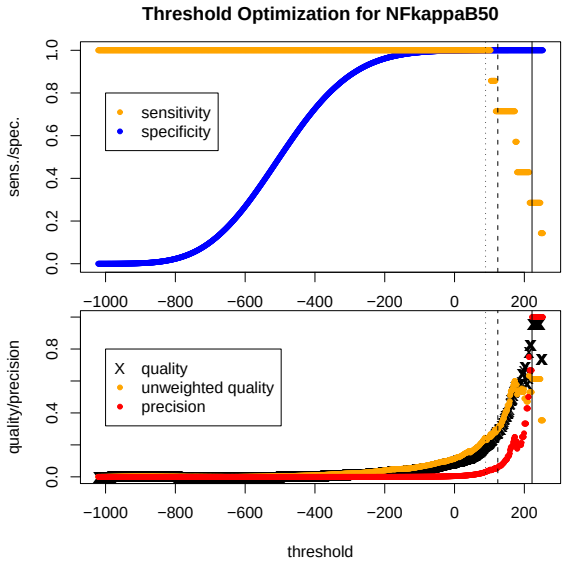

Figure 11.1: Sensitivity (light grey / orange solid circles) and specificity (black / blue solid circles), both upper panel. Quality (black 'X'), unweighted quality (light grey / orange solid circles), and precision (dark grey / red solid circles), all lower panel, for different thresholds (x-axis). The dashed (dotted) line corresponds to the balanced threshold (Rahmann, 2003); the dotted line to a type-I error probabilty of 0.1, and the solid line indicates the quality-optimized threshold.

11.2.6 Example

In the Introduction, we state that the PFM A_1 based on the four verified binding sites 'AAAC', 'ACAC', 'AACC' and 'ACCC' has the maximum quality while the PFM A_2 based on the four verified binding sites with duplicates 'AACC' and 'ACAC' has lower quality. Indeed, the quality for A_1 is 1.0 and for A_2 0.779. Hence, the quality measure does capture these subtle position dependencies. The reported quality values are maximized by optimizing the threshold as we do throughout the remaining article.

11.3 Results

11.3.1 Threshold Optimization

In the following, we illustrate the threshold optimization for a PFM selected from the Transfac database v11.1 (Matys *et al.*, 2003): it is the matrix M00051 for NFκB50 (Kunsch *et al.*, 1992). Figure 11.1 shows the sensitivity, specificity, precision, quality and unweighted quality for NFκB50.

In the upper panel, the specificity smoothly increases with increasing threshold reaching 0.99 at $t = -104$ while the sensitivity decreases in a few steps for very high thresholds and remains 1.0 for thresholds $t \in [-1022, 103]$. The lower panel contains the precision, quality and unweighted quality. For low thresholds, the precision increases with increasing threshold. Around $t = 180$, the precision drops but increases again for higher thresholds.

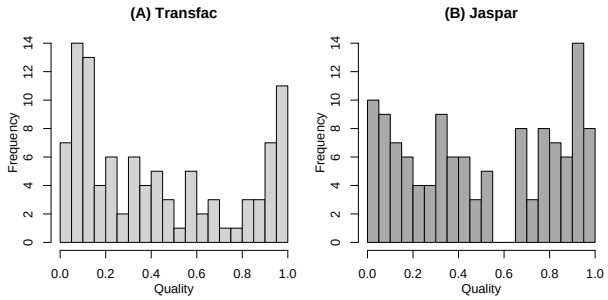

Figure 11.2: Histogram of the optimized quality measures for (A) all vertebrate Transfac PFMs and (B) all core Jaspar PFMs.

The quality also increases with increasing threshold but drops from time to time for higher thresholds. The quality reaches a maximum at $t = 222$ and remains there till $t = 246$ but drops afterwards. The unweighted quality has a similar shape but the maximum occurs for a slightly lower threshold $t = 213$.

The smooth increase of the specificity is due to the huge size of the complement set of experimentally verified binding sites (around one million sequences). Thus, for each increment of the threshold some further words are excluded from the set of compatible words. In contrast, the set of verified binding sites is much smaller (18 sequences). Therefore, the sensitivity can only attain a limited number of values. Since the verified binding sites are already included in the set of compatible words for a high threshold ($t = 103$), the sensitivity remains 1.0 for lower thresholds. The precision equals 1.0 for the highest thresholds, since all compatible words are also in the set of verified binding sites. However, the precision does not reflect how many of the verified binding sites are considered in the set of compatible words. This behavior is typical, and neither sensitivity nor specificity nor precision alone can be used to determine an optimal threshold.

In contrast, the proposed quality measure summarizes all three notions in a reasonable way. The quality does not yield a unique maximum. However, within the threshold interval for maximum quality, the set of compatible words does not change since all trajectories (for sensitivity, specificity, and precision) are constant there. The maximum unweighted quality occurs for a slightly lower threshold than the maximum weighted quality. Hence, the further increase of the threshold removes infrequently observed verified binding sites from the compatible set of words (otherwise, also the weighted quality measure would decrease.)

11.3.2 Transfac and Jaspar PFMs

We apply the quality measure to PFMs taken from the two well known databases Transfac (Matys *et al.*, 2003) and Jaspar (Sandelin *et al.*, 2004b). From Transfac version 11.1, we select the vertebrate PFMs; from Jaspar version 3.0, we take the 'core' set.

Figure 11.2 shows quality histograms for the vertebrate Transfac PFMs (A) and the Jaspar PFMs (B) after threshold optimization for quality. In Fig. 11.2A, the quality values tend to be extreme. About two thirds of the PFMs yield a lower quality than 0.5. For a higher

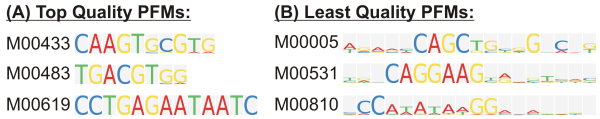

Figure 11.3: Sequence Logos (Crooks *et al.*, 2004) of (A) the three top quality PFMs and (B) the three lowest quality PFMs of the Transfac set.

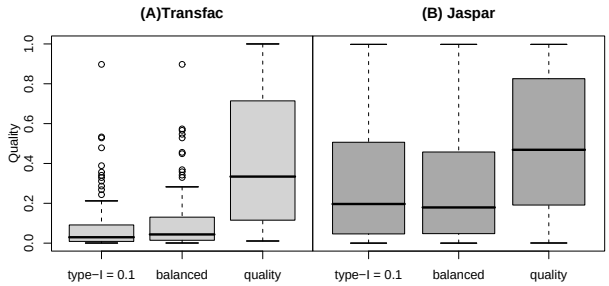

Figure 11.4: Box-plot for quality values based on three different threshold selection methods 'type-I' error bounded to 0.1, 'balanced' type-I and type-II error, and 'quality' optimized for (A) all vertebrate Transfac PFMs and (B) all Jaspar PFMs.

quality than 0.8, the number of PFMs increases until the maximum quality of 1.0. Figure 11.2B for the Jaspar PFMs is more balanced between high- and low-quality PFMs. Still, the numbers of PFMs slightly increase towards extreme quality values.

We have depicted the three highest quality PFMs and the three lowest quality PFMs of the Transfac set in Fig. 11.3. The three top ranking PFMs have a length between 8 and 13. Almost all positions have one strong consensus letter leading to a high information content. In contrast, the three least quality PFMs are of length 18 and have many positions with high variability and, therefore, low information content. The three top ranking PFMs are based on verified binding sites with high multiplicities. In contrast, the sets of verified sequences for the least ranking PFMs are small (between 5 and 27 sequences) without multiple observed sequences.

In general, representation quality correlates with the average per-position information content of the PFM, as might be guessed from the examples in Fig. 11.3. On the Transfac set, the Pearson correlation coefficient is 0.90, on the Jaspar set it is 0.82 (details not shown).

The differences of the quality for the chosen threshold selection method are shown in Fig. 11.4. The three left box-plots correspond to the Transfac PFMs. For the first empirical distribution, the type-I error is bounded by 0.1. The mean of the distribution is 0.09. The second distribution with a mean of 0.11 is based on the balanced threshold while the values of the third distribution with mean 0.43 are the optimized qualities. The quality values of the type-I and balanced method are statistically worse than the values of the optimized quality (p-values of a Kolmogorov-Smirnov test (Birnbaum and Tingey, 1951) are $3.3 \cdot 10^{-16}$ for comparison with the type-I method and $2.4 \cdot 10^{-12}$ for comparison with the balanced method). The last three box-plots show the corresponding distributions for the Jaspar PFMs. The means of the quality values for the 'type-I' and the 'balanced' threshold methods are both 0.30. Again, both distributions have a similar shape. The distribution

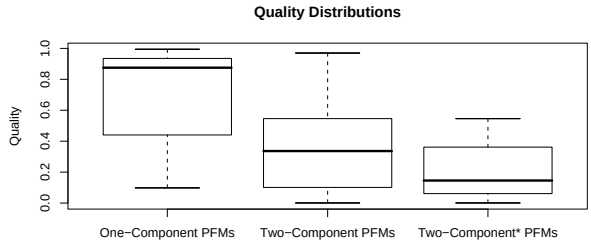

Figure 11.5: Box-plot for the comparison of the quality distributions for PFMs with one mixture component, two components, and two components supported by high conservation scores.

for the optimized quality values has a mean of 0.50. Also for the Jaspar PFMs the quality values significantly improve by maximizing the quality (p-values $1.5 \cdot 10^{-5}$ for comparison with the type-I method and $4.3 \cdot 10^{-6}$ for comparison with the balanced method). Hence, choosing the threshold by maximizing the quality instead of defining the threshold based on the type-I or type-II probabilities significantly increases the quality.

11.3.3 Quality for Mixture Models

We hypothesize that lower quality reflects dependencies between positions. Hence, the representation quality for DNA motifs with position dependencies should be significantly lower than for DNA motifs with independent positions. We test our hypothesis on 64 Jaspar PFMs, divided into three sets, according to the mixture model from Georgi and Schliep (2006), which allows to represent a binding site model by a mixture of PFMs instead of a single PFM.

The first set contains PFMs which only have one mixture component (and hence correspond to a standard PFM). The second set contains PFMs with two mixture components. The third set is a subset of the second and only contains PFMs which additionally reach a high conservation score (for details, see Georgi and Schliep, 2006), indicating that the both components may represent biologically meaningful motifs. Since the number of mixture components is estimated based on the verified binding sequences, we assume that two mixture components reflect position dependencies. Figure 11.5 shows a box-plot of quality values for each set.

The left box-plot corresponds to the empirical distribution of the optimized quality values of the 23 one-component PFMs. The distribution has a mean of 0.72 and a median of 0.88.

The center box-plot corresponds to the 41 two-component PFMs with a mean of 0.36 and a median of 0.34. Based on a Kolmogorov-Smirnov test (Birnbaum and Tingey, 1951), we reject the null hypothesis that the quality values for the one-component PFMs are statistically worse than or equal to those of the two-component PFMs (p-value: 0.00021).

The rightmost box-plot in Fig. 11.5 belongs to the 9 two-component PFMs with high conservation scores. The distribution has a mean of 0.20 and a median of 0.15. Performing

a Kolmogorov-Smirnov test yields no significant difference between the quality distributions of all two-component PFMs and those with high conservation scores (p-value of 0.32). However, there is a significant difference to the one-component PFMs (p-value: 0.00059).

Hence, the representation quality indeed attains significantly higher values for one-component PFMs. Under the assumption that the mixture modelling captures position dependencies, the representation quality can be used to assess the strength of position dependencies of PFMs.

11.4 Discussion

We have proposed a new quality measure to assess the degree of representation of a binding site model regarding the experimentally verified binding sites. The quality measure is based on the asymptotic correlation of the number of occurrences of the model-compatible words and of the verified binding sequences on an infinite random sequence. We emphasize that the quality measure is general and does not depend on the PFM model. Thus, extension to other models like mixture models is possible. However, the computational effort to calculate the two-dimensional score convolution for complex models might become infeasible. For PFM models, the representation quality can be efficiently computed.

Different objective functions can be used to optimize the threshold of a PFM. Here, we use the maximum representation quality which is shown to significantly improve the quality. Some (non-multiple) verified binding sites may not be contained in the set of compatible words. Due to experimental error and erroneous binding detection especially in high-throughput methods like SELEX (Oliphant *et al.*, 1989) and CASTing (Wright *et al.*, 1991), this can be advantageous if stronger support for a binding site is reflected by multiple detection. However, if the experimental evidence for all given binding sites is very strong, one may want to additionally consider the sensitivity value.

The histogram of the optimized quality values for the Transfac and Jaspar PFMs reveals a large fraction of PFMs with poor quality (Fig. 11.2). Analysis of the sequence logos (Fig. 11.3) reveals that these PFMs are long and have a low information content per position since they are based on few verified binding sites. The problem is that many positions exist which are variable but not completely flexible. The more such positions occur, the more combinations of variations exist. If only a few of these combinations are reflected in the set of verified binding sites, position dependencies are automatically introduced. Consider the following set of verified binding sites {'CAAAG,'ACAAG','AACAT','AAACT'}. This leads to a PSM qualitatively like

$$\begin{array}{c} \text{'A'} \\ \text{'C'} \\ \text{'G'} \\ \text{'T'} \end{array} \begin{pmatrix} 10 & 10 & 10 & 10 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 5 \end{pmatrix}.$$

Setting the threshold to $t = 45$ yields two compatible words 'AAAAG' and 'AAAAT', which are both not in the set of verified sequences, leading to a low quality value. Decreasing the threshold to $t = 40$ gives two further compatible words 'AAAAC' and 'AAAAA', again not contained in the set of verified binding sites. The next lower threshold of $t = 36$ is the first

where the set of compatible words overlaps with the set of verified binding sites. In fact, all verified binding sites are compatible words but also the words 'CAAAA', 'CAAAC', 'CAAAT', 'ACAAA', 'ACAAC', 'ACAAT', and so on. Hence, the best quality can be achieved by having all four verified binding sites as compatible words plus 16 non-verified sequences. Of course, this is an extreme example but the combinatorial effect increases with the length of the motif. Thus, it is not surprising that we obtain low quality values for long PFMs based on small verified sets.

We can deal with long motifs and few verified binding sites (that may furthermore be inconsistent – low multiplicities) in two ways. One way is to use more complex models, such as mixtures. The other way is to gather further experimental evidence for the DNA motif. The decision should be made based on the size of the verified set. Without further experiments, the PFM model may still be a reasonable choice due to its ability to generalize the verified sequences. Therefore, a bad quality does not necessarily contradict the use of the PFM model. It just points to the fact that the PFM model does not tightly correspond to the verified binding sites.

Another application of the representation quality measure could be in the discovery of new DNA motifs (for a review, see Wasserman and Sandelin, 2004; Das and Dai, 2007). Strong position dependencies in the DNA motif decrease the quality measure. If the PFM is based on many verified binding sites, this indicates that a more complex model or a set of PFMs might be preferable as model for the DNA motif. Hence, given a set of upstream sequences of putatively co-regulated genes, one can learn new DNA motifs while using our measure to assess the quality of the PFM. If the quality decreases, the PFM might represent two different DNA motifs such that two PFMs might be the better choice.

Chapter 12

Conclusion

The thesis solves many mainly statistical problems in context of DNA motifs. Supplementary to the thesis, a C++ program suite as well as a website http://mosta.molgen.mpg.de can be used to apply the statistics (see Appendix A for a description of the software package and its multi-processor and parallel computing abilities). Mainly, we consider the PFM model for representation of TFBS. However, some - namely the similarity and quality measures - can be directly applied to other DNA motif models. Besides being new solutions to existing problems, all our developments have two innovative properties in comparison to other existing methods:

- We explicitly consider overlap probabilities. However, the set of compatible words is only used implicitly by the two-dimensional score convolution, thus, our algorithms do not need to enumerate them and, therefore, are very fast.
- The complementary strand of DNA is incorporated, thus, we correctly consider multiple occurrences of palindromes by explicit modelling of the position dependency structure.

In this chapter, we first summarize our results. Then, we mention existing problems and roadmaps to their possible solutions.

12.1 Summary

A PFM with a given threshold can be represented by its set of compatible words. The main problem is enumeration of all compatible words. Depending on the threshold selection method (see Section 2.4.3), this takes exponential time (see Section 2.4.4). To be more precise, this is the case if the threshold is selected based on type-I error (false positive) probabilities. So the false positive probability is approximately independent of the length of the PFM. This desirable statistical criterion is important in obtaining comparable results for PFMs of different lengths. Therefore, in practise, this is the favorite threshold selection method. Hence, practical computational tools are required to avoid the costly enumeration of compatible words (see Section 7.5). Since we derive such tools from word counting approaches and, furthermore, word counting approaches can be restrictively applied to small PFMs, this thesis also reviews word counting approaches.

12.1.1 Word Counting

Word counting approaches deal with the difficulty of (self-)overlaps of words (see Chapter 3). Exact solutions are usually computed either by recursive formulae (Gentleman, 1994; Robin *et al.*, 2005), generating functions (first publications are Goulden and Jackson, 1983; Fudos *et al.*, 1996; Koutras, 1997), language decomposition approaches (Régnier, 2000; Régnier and Denise, 2004) or by automatons slided over the sequence (Kleffe and Langbecker, 1990). Surprisingly, exact formulae for homogeneous clumps have only recently be published based on generating functions (Stefanov *et al.*, 2007). However, there are no recursive formulae available. To close this gap, we have newly derived such formulae (see Section 3.3.1), which are straight-forward to apply. Furthermore, we show how to compute the exact expected value and the (limiting) variance.

In practise, exact approaches can only be computed for small sequences since they are computationally very demanding. Approximations can solve this problem. There are two approximation scenarios: First, words occur frequently; second, words occur rarely. For frequent words, one can use a normal approximation based on the asymptotic expected value and the asymptotic/limiting variance (Waterman, 2000). Here, we have contributed a new normal approximation for homogeneous clumps (see Section 3.3.4 and Section 3.5.1). However, PFM occurrences only occur rarely. In this case, a (compound) Poisson distribution is more appropriate (Reinert and Schbath, 1999; Roquain and Schbath, 2007).

Compound Poisson Distribution The compound Poisson distribution can be used in two equivalent ways to model occurrences (Johnson *et al.*, 1995):

- Clumps are distributed like Poisson; the size of each clump follows another (e.g. geometric) distribution independently and identically.
- For each $k \geq 1$, the occurrences of k-clumps are modeled by a Poisson distribution.

Compound Poisson Distribution for Homogeneous Clumps Considering only homogeneous clumps (Reinert and Schbath, 1999), self-overlaps of words are considered in the rate of clumps. Therefore, Chen-Stein error bounds converge to 0 for words which might have self-overlaps but cannot overlap with any other word in the set. However, homogeneous clumps of different words can overlap. Since this overlap cannot be incorporated into the Poisson distribution, the Chen-Stein error bounds do not converge to 0 for sets of overlapping words.

Compound Poisson Distribution for Heterogeneous Clumps The recent approach from Roquain and Schbath (2007) allows heterogeneous clumps. Hence, overlaps - self-overlaps as well as overlaps between different words - are incorporated into the Poisson rate for clumps. Therefore, the Chen-Stein approximation error converges to 0 independent of the overlapping structure between words.

12.1.2 TF Count Statistics

Chapter 5 contains our main contribution. Among this are the exact formula for the variance and an approximation for the count distribution of PFM occurrences without explicitly enumerating the set of compatible words. Furthermore, the complementary strand is incorporated which is important for palindromes. The variance of the number of counts is complicated by computation of second moments. Hence, the essence is the computation of the joint probability of two occurrences shifted by d positions. Since these two occurrences require a score reaching the threshold at both positions, one can use the 2-dimensional score convolution to compute the corresponding probability. This leads to an exact formula for the variance and the limiting variance which can be applied as similarity (see Section 9) and quality measures (see Section 11). Furthermore, we newly derive generating functions for a simplified background model.

Based on the joint occurrence probabilities, we also contributed the first efficient[1] compound Poisson distribution for PFMs. To incorporate the complementary strand, we developed recursive formulae, which explicitly model the extended dependency structure between positions. Other approaches usually ignore the complementary strand due to the complicated dependency structure. Thus, the incorporation of the complentary strand is another important contribution. Furthermore, we derive two characteristic values, which describe the palindromicity and the self-overlap of PFMs. This is the first known quantitative method to assess these two characteristics. Results in Chapter 7 clearly indicate that our approximation is very accurate and that this approach is the only one which considers palindromes correctly. Furthermore, Section 7.5 proves that the exact approach (Zhang *et al.*, 2007) cannot generally be applied to PFMs with threshold selection based on error probabiltities due to computational issues. Hence, our contributed approximation is the only efficient and accurate statistics for count distributions of PFMs.

12.1.3 cis-regulatory modules (CRMs)

Statistical assessment of CRMs is one of the main problems in sequence analysis. Besides the exact approach (Boeva *et al.*, 2007), which is computationally very demanding since exponential in the number of compatible words, no stringent statistics were available. Based on the overlap probabilities retrieved by the 2-dimensional score convolution, we derive the first efficient and accurate approximation for the probability to observe at least one occurrence of each TF in a given set in a window of a sequence (see Chapter 6). If the window is small, the occurrence of each TF determines a CRM. Since TFs are co-operative if they tend to co-occur, we propose to count the number of windows with such co-occurrences and derive formulae to compute the significance. Furthermore, we give Chen-Stein error bounds for the approximation of the significance such that one can also use overlapping windows after quantifying the error. Hence, using our framework, the following questions can be quantitatively answered:

- What is the probability of observing co-occurrences (at least one occurrence of each TF in a given set) in a random sequence window of given length?

[1] This means without enumerating all compatible words.

- How small/large should the window be such that the probability of co-occurrence is still significant?
- How significant is the co-operativity within a set of TFs?
- What is the bound of the co-operativity approximation using overlapping windows?

The results show that similar PFMs - PFMs which can overlap - need smaller windows such that the probability for co-occurrence is significant. The reason is that similar PFMs have a much higher chance of co-occuring in random sequences. In fact, our framework incorporates the similarity between PFMs in the null model. Furthermore, significance of co-operativity yields large bounds for highly overlapping windows. Hence, the best choice are windows, which only overlap at their boundaries. Finally, similar PFMs always yield larger bounds than non-similar PFMs since the approximation error depends on the overlap probabilities.

In comparison to an approach not considering the similarity of PFMs, simulations show that the new approach yields very accurate results. Since the empirical frequency can be incorporated into the approach, empirically more frequent - than theoretically assumed - motifs do not have a bias towards better significance values. Hence, we give the first accurate and statistically sound assessment of co-occurrences and co-operativity of sets of TFs, which is not based on the set of compatible words.

12.1.4 Similarity

Similarity between DNA motifs is important in detecting database redundancies and to remove bias in sequence annotation. So far, only PFM-specific similarity measures are proposed, which mainly rely on the distance between the position-specific distributions (for example, see Schones *et al.*, 2005; Gupta *et al.*, 2007). DNA motifs can be shifted towards each other. Thus, one obtains distances for each possible shift. Until now there has been no stringent way to summarize these distances for all possible shifts/overlaps.

In Chapter 9, we contribute a *general* and *natural* similarity measure for DNA motifs. It is general because it is not restricted to PFMs but can be computed for any DNA motif model as long as the second moments of its count distribution are known. Furthermore, it is natural since we use the tendency for overlapping occurrences captured in the limiting covariance. In addition, one directly obtains a measure for each pair of DNA motifs instead of one measure for each possible shift for each pair of DNA motifs. Based on the exact variance of PFM counts we derived earlier (see Chapter 5), we give explicit formulae for PFMs which can be efficiently computed.

Our new measure is compared to simulated data annotated by artificially generated PFMs and Transfac PFMs (Matys *et al.*, 2003). The results show that the new measure accurately captures the tendency of overlaps. Furthermore, none of the competitive approaches (Schones *et al.*, 2005; Roepcke *et al.*, 2005; Gupta *et al.*, 2007) is able to reflect this tendency. Hence, we give the first *general* and *natural* similarity measure of DNA motifs which has is readily interpretable.

12.1.5 Clustering

The similarity of PFMs can also be used to cluster PFMs (Mahony *et al.*, 2007). This is useful to obtain representative PFMs for sets of similar PFMs, as well as deciphering classes of TFs (Sandelin and Wasserman, 2004). We propose to compute an ungapped alignment based on a similarity measure between two PFMs. As a similarity measure we derive the maximal overlap probability which is related to the similiarty described above (see Chapter 9). In this context, its advantage is that one directly obtains the ungapped alignment. In Chapter 10, we develop a clustering algorithm which ensures if the cluster is extended that all members still share sufficient similarity to one another. Furthermore, we show how to compute a cluster representative (FBP), which reflects all shared characteristics of the cluster members.

The algorithm can be applied to databases like Transfac (Matys *et al.*, 2003) or Jaspar (Sandelin *et al.*, 2004a) to merge redundancies into new cluster representatives, which could then be used to annotate sequences. However, to evaluate the performance of the clustering, we apply it on selected class-labeled Jaspar PFMs and compare our result with the best result of a clustering study by Mahony *et al.* (2007). The new clustering yields as good results as the best - and, furthermore, optimized - method from the clustering study. Furthermore, the clusters are more consistent. Hence, the new clustering algorithm can be used to solve problems of redundancies in databases and finding classes of TFs.

12.1.6 Quality

Models for DNA motifs serve as a unifying framework for analysis of TFBS. A model represents the experimentally verified binding sites. Depending on the model, certain simplifications are done. For example, the PFM model assumes position independence. Thus, DNA motifs with position dependencies cannot be correctly represented by PFMs. Therefore, it is important to measure the quality of representation for DNA motifs and a chosen model. Up to now, this problem has not been tackled generally. There are only specific approaches, which measure the gain of representation by using a specific more complex model usually based on an information criterion like BIC (Bayesian Information Criterion, see Schwarz, 1978) or AIC (Akaike Information Criterion, see Akaike, 1974).

We propose a very intuitive approach: We compare the set of experimentally verified sites with the set of words reflecting an occurrence for the DNA motif (for PFMs: the set of compatible words) based on its similarity. Despite the fact that enumeration of all compatible words is inefficient for PFMs, the main problem is how to compare the sets.

Instead of using sensitivity, specificity or precision, we propose to compute the similarity/limiting covariance between the set of experimentally verified binding sites and the DNA motif model. We give explicit formulae to efficiently compute such a quality measure for PFMs. Since the quality measure depends on the threshold, one can also choose the threshold by optimizing the quality. For Transfac (Matys *et al.*, 2003) and Jaspar (Sandelin *et al.*, 2004a) PFMs, we show that such an optimization significantly changes the the quality of the PFMs. In addition, we prove that the quality measure is able to detect DNA motifs with position dependencies by comparing the quality values to PFMs which have been shown to be better represented by more complex models (Georgi and Schliep, 2006).

Thus, we derive the first general quality of representation measure to assess how well a set of experimentally verified sequences is represented by a DNA motif model.

12.2 Outlook

Although this thesis mainly deals with DNA motifs, our first contribution is an exact formula for the number of clumps for word count statistics. For DNA motifs represented as PFMs, we contribute the first exact variance which is - similar to all following contributions - not based on the enumeration of compatible words. We also derive the first generating functions for the number of counts of PFMs using a simple background model. The core of the thesis regards the count and co-occurrence statistics both incorporating most relevant dependencies - the overlap probabilities and the complementary strand. Based on these formulations, we derive a new general similarity and quality measure as well as a clustering algorithm. Eventually, we suggest to use the quality measure to optimize model parameters such as the threshold. We show the better performance of all contributions by comparison to existing methods. Furthermore, we supply a website to perform these calculations as well as a sophisticated software package which additionally support parallel processing. Although PFMs are already used since more than a decade, we provide the first comprehensive theoretical and practical package of statistics solving relevant statistical issues for DNA motifs represented as PFMs.

Since the single parts of the thesis have already been discussed in the corresopnding chapters, here, we only give some general remarks. Throughout the whole thesis, the sequence model is restricted to the Bernoulli (i.i.d.) sequence model. Although this model suffices to shed some light on biology, an extension to a more sophisticated Markov model would improve the applicability of our derived statistics. In fact, the main problem is computation of the 2-dimensional score convolution. However, there are no principal problems to extend the convolution to deal with higher order Markov models. For a Markov model of first order, one has to assign to each score the last nucleotide. This enlarges the required memory and also the computational time linearly by the alphabet size. Furthermore, one has to ensure that the sequence model is symmetric otherwise it is not obvious how to incorporate the complementary strand.

The similarity as well as the quality are more general measures than the PFM model. In fact, one can apply them to any DNA motif model as long as one can compute the covariance of the number of occurrences. Therefore, deriving explicit formulae for more sophisticated DNA motif models (e.g. Georgi and Schliep, 2006) would increase the applicability of both measures. Furthermore, one could compare DNA motif representations between different models.

So far, the accuracy of the statistics are only shown based on simulations as a proof-of-principle. The simulations use the same sequence model as our statistics. Of course, real data is more complex. Hence, the next step is to apply the statistics to real biological examples and to analyze the correspondence of the results to known biological facts. This might make certain extensions regarding the sequence model necessary. However, after the statistics are confirmed by such analyses, the statistics can be used to gather new biological insights. Hence, one would apply the approaches to new data to generate hypotheses, which can be verified by wet-lab experiments.

Part III

Appendix

Bibliography

Aerts, S., Loo, P. V., Thijs, G., Moreau, Y., and Moor, B. D., 2003. Computational detection of cis -regulatory modules. *Bioinformatics* 19 Suppl 2, ii5–14.

Agarwal, P. and Bafna, V., 1998. Detecting non-adjoining correlations with signals in DNA. In *RECOMB'98*, 2–8. ACM, New York, New York, United States.

Aho, A. and Corasick, M., 1975. Efficient string matching. *CACM* 18, 333–340.

Akaike, H., 1974. A new look at the statistical model identification. *IEEE Transactions on Automatic Control* 19, 716–723.

Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P., 2002. *Molecular Biology of the Cell*. Garland Science, New York.

Aldrich, J., 1997. R.A. Fisher and the making of maximum likelihood 1912-1922. *Statistical Science* 12, 162–176.

Almagor, H., 1983. A Markov analysis of DNA sequences. *J. Theor. Biol.* 104, 633–645.

Altschul, S. and Erickson, B., 1985. Significance of nucleotide sequence alignments: a method for random sequence permutation that preserves dinucleotide and codon usage. *Mol. Biol. Evol.* 2, 526–538.

Arnone, M. and Davidson, E., 1997. The hardwiring of development: organization and function of genomic regulatory systems. *Development* 124, 1851–1864.

Arratia, R., Goldstein, L., and Gordon, L., 1989. Two Moments Suffice for Poisson Approximations: The Chen-Stein Method. *The Annals of Probability* 17, 9–25.

Arratia, R., Goldstein, L., and Gordon, L., 1990. Poisson Approximation and the Chen-Stein method. *Statistical Science* 5, 403–434.

Bailey, N. T. J., 1977. *Mathematics, Statistics and Systems for Health*. Wiley, NY.

Bailey, T. and Elkan, C., 1994. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Intell. Sys. Mol. Biol.* 2, 28–36.

Bailey, T. and Elkan, C., 1995. The value of prior knowledge in discovering motifs with. *Intell. Sys. Mol. Biol.* 3, 21–29.

Bailey, T. and Gribskov, M., 1998. Combining evidence using p-values: application to sequence homology searches. *Bioinformatics* 14, 48–54.

Bailey, T. and Noble, W., 2003. Searching for statistically significant regulatory modules. *Bioinformatics* 19, II16–II25.

Barash, Y., Elidan, G., Friedman, N., and Kaplan, T., 2003. Modeling dependencies in protein-dna binding sites. In *RECOMB '03: Proceedings of the seventh annual international conference on Research in computational molecular biology*, 28–37. ACM, New York, NY, USA.

Barbour, A. D. and Chryssaphinou, O., 2001. Compound Poisson approximation: a user's guide. *Ann. Appl. Probab.* 11, 964–1002.

Barbour, A. D., Holst, L., and Janson, S., 1992. *Poisson Approximation*. Oxford University Press.

Bassino, F., Clément, J., Fayolle, J., and Nicodéme, P., 2007. Counting occurrences for a finite set of words: an inclusion-exclusion approach. In *DMTCS Proceedings of the International Conference on Analysis of Algorithms*, 29–42.

Beckstette, M., Homann, R., Giegerich, R., and Kurtz, S., 2006. Fast index based algorithms and software for matching position specific scoring matrices. *BMC Bioinformatics* 7, 389.

Bender, E. and Kochman, F., 1993. The Distribution of Subwords Counts is Usually Normal. *European Journal of Combinatorics* 14, 265–275.

Benos, P., Lapedes, A., Fields, D., and Stormo, G., 2001. SAMIE: statistical algorithm for modeling interaction energies. *Pac Symp Biocomput* 115–26.

Berg, J., Tymoczko, J., and Stryer, L., 2002. *Biochemistry*. W. H. Freeman and Company.

Berg, O. and von Hippel, P., 1987. Selection of DNA binding sites by regulatory proteins. Statistical-mechanical theory and application to operators and promoters. *J. Mol. Biol.* 193, 723–50.

Berman, B., Pfeiffer, B., Laverty, T., Salzberg, S., Rubin, G., Eisen, M., and Celniker, S., 2004. Computational identification of developmental enhancers: conservation and function of transcription factor binding-site clusters in drosophila melanogaster and drosophila pseudoobscura. *Genome Biol* 5, R61.

Berman, B. P., Nibu, Y., Pfeiffer, B. D., Tomancak, P., Celniker, S. E., Levine, M., Rubin, G. M., and Eisen, M. B., 2002. Exploiting transcription factor binding site clustering to identify cis-regulatory modules involved in pattern formation in the Drosophila genome. *Proc Natl Acad Sci U S A* 99, 757–62.

Biggins, J. and Cannings, C., 1987. Markov renewal processes, counters and repeated sequences in Markov chains. *Adv. Appl. Probab.* 19, 521–545.

Bilu, Y. and Barkai, N., 2005. The design of transcription-factor binding sites is affected by combinatorial regulation. *Genome Biol* 6, R103.

Birnbaum, Z. W. and Tingey, F. H., 1951. One-sided confidence contours for probability distribution functions. *The Annals of Mathematical Statistics* 22, 592–596.

Bleser, P. D., Hooghe, B., Vlieghe, D., and van Roy, F., 2007. A distance difference matrix approach to identifying transcription factors that regulate differential gene expression. *Genome Biol* 8, R83.

Boeva, V., Clément, J., Régnier, M., Roytberg, M. A., and Makeev, V. J., 2007. Exact p-value calculation for heterotypic clusters of regulatory motifs and its application in computational annotation of cis-regulatory modules. *Algorithms Mol Biol* 2, 13.

Borodovsky, M., Sprizhitsky, A., Golovanov, E., and Alexandrov, A., 1986. Statistical patterns in the primary structures of functional regions in the genome of E.coli. *Molek. Biol.* 20, 1014–1035.

Boyd, K. E., Wells, J., Gutman, J., Bartley, S. M., and Farnham, P. J., 1998. c-Myc target gene specificity is determined by a post-DNAbinding mechanism. *Proceedings of the National Academy of Sciences* 95, 13887–13892.

Brazma, A., Jonassen, I., Eidhammer, I., and Gilbert, D., 1998a. Approaches to the automatic discovery of patterns in biosequences. *J. Comput. Biol.* 5, 279–305.

Brazma, A., Jonassen, I., Vilo, J., and Ukkonen, E., 1998b. Predicting gene regulatory elements in silico on a genomic scale. *Genome Res.* 8, 1202–1215.

Breen, S., Waterman, M., and Zhang, N., 1985. Renewal theory for several patterns. *J. Appl. Probab.* 22, 228–234.

Brendel, V., Beckmann, J., and Trifonov, E., 1986. Linguistics of nucleotide sequences: morphology and comparison of vocabularies. *J. Biomol. Struct. Dyn.* 4, 11–21.

Brendel, V. and Trifonov, E., 1984. A computer algorithm for testing potential prokaryotic terminators. *Nucleic Acids Res* 12, 4411–27.

Brown, C., Rust, A., Clarke, P., Pan, Z., Schilstra, M., De Buysscher, T., Griffin, G., Wold, B., Cameron, R., Davidson, E., and Bolouri, H., 2002. New computational approaches for analysis of cis-regulatory networks. *Dev Biol* 246, 86–102.

Buck, M. J. and Lieb, J. D., 2004. ChIP-chip: considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments. *Genomics* 83, 349–60.

Bulyk, M. L., 2003. Computational prediction of transcription-factor binding site locations. *Genome Biol* 5, 201.

Bulyk, M. L., Johnson, P. L. F., and Church, G. M., 2002. Nucleotides of transcription factor binding sites exert interdependent effects on the binding affinities of transcription factors. *Nucleic Acids Res.* 30, 1255–1261.

Bussemaker, H., Li, H., and Siggia, E., 2001. Regulatory element detection using correlation with expression. *Nat Genet* 27, 167–71.

Cereghini, S., Saragosti, S., Yaniv, M., and Hammer, D., 1984. SV40-alpha-globulin hybrid minichromosomes. Differences in DNaseI hypersensitivity of promoter and enhancer sequences. *Eur. J. Biochem.* 144, 545–553.

Chen, L. H. Y., 1975. Poisson approximation for dependent trials. *Ann. Probab.* 3, 534–545.

Choi, I.-G., Kwon, J., and Kim, S.-H., 2004. Local feature frequency profile: A method to measure structural similarity in proteins. *PNAS* 101, 3797–3802.

Choo, K., Vissel, B., Nagy, A., Earle, E., and Kalitsis, P., 1991. A survey of the genomic distribution of alpha satellite DNA on all the human chromosomes, and derivation of a new consensus sequence. *Nucleic Acids Res* 19, 1179–82.

Chryssaphinou, O. and Papastavridis, S., 1988. A limit-theorem for the number of nonoverlapping occurences of a pattern in a sequence of independent trials. *J. Appl. Probab.* 25, 428–431.

Chryssaphinou, O., Papastavridis, S., and Vaggelatou, E., 2001. Poisson Approximation for the Non-Overlapping Appearances of Several Words in Markov Chains. *Combinatorics, Probability and Computing* 10, 293–308.

Claverie, J.-M. and Audic, S., 1996. The statistical significance of nucleotide position-weight matrix matches. *Comput. Appl. Biosci.* 12, 431–439.

Clyde, D., M.S.Corado, Wu, X., Pare, A., Papatsenko, D., and Small, S., 2003. A self-organizing system of repressor gradients establishes segmental complexity in Drosophila. *Nature* 426, 849–853.

Comtet, L., 1974. *Advanced Combinatorics*. Reidel, Dordrecht, The Netherlands.

Corá, D., Herrmann, C., Dieterich, C., Cunto, F. D., Provero, P., and Caselle, M., 2005. Ab initio identification of putative human transcription factor binding sites by comparative genomics. *BMC Bioinformatics* 6, 110.

Crawford, G. E., Holt, I. E., Whittle, J., Webb, B. D., Tai, D., Davis, S., Margulies, E. H., Chen, Y., Bernat, J. A., Ginsburg, D., Zhou, D., Luo, S., Vasicek, T. J., Daly, M. J., Wolfsberg, T. G., and Collins, F. S., 2006. Genome-wide mapping of DNase hypersensitive sites using massively parallel signature sequencing (MPSS). *Genome Res* 16, 123–31.

Crooks, G. E., Hon, G., Chandonia, J.-M., and Brenner, S. E., 2004. Weblogo: a sequence logo generator. *Genome Res* 14, 1188–1190.

Crowley, E., Roeder, K., and Bina, M., 1997. A statistical model for locating regulatory regions in genomic DNA. *J Mol Biol* 268, 8–14.

Dagum, L. and Menon, R., 1998. OpenMP: An Industry-Standard API for Shared-Memory Programming. *IEEE Comput. Sci. Eng.* 5, 46–55.

Das, M. K. and Dai, H.-K., 2007. A survey of DNA motif finding algorithms. *BMC Bioinformatics* 8 Suppl 7, S21.

Day, W. H. and McMorris, F., 1992. Critical comparison of consensus methods for molecular sequences. *Nucl. Acids Res.* 20, 1093–1099.

Dayhoff, J., 1984. Distinguished words in data sequences: analysis and applications to neural coding and other fields. *Bull. Math. Biol.* 46, 529–543.

de Wet, J., Wood, K., DeLuca, M., Helinski, D., and Subramani, S., 1987. Firefly luciferase gene: structure and expression in mammalian cells. *Mol Cell Biol* 7, 725–37.

Dorschner, M. O., Hawrylycz, M., Humbert, R., Wallace, J. C., Shafer, A., Kawamoto, J., Mack, J., Hall, R., Goldy, J., Sabo, P. J., Kohli, A., Li, Q., McArthur, M., and Stamatoyannopoulos, J. A., 2004. High-throughput localization of functional elements by quantitative chromatin profiling. *Nat Methods* 1, 219–25.

Drouin, R., Therrien, J., Angers, M., and Ouellet, S., 2001. In vivo DNA analysis. *Methods Mol Biol* 148, 175–219.

Edmonds, J. and Johnson, E., 1973. Matching, Euler Tours, and the Chinese Postman. *Math. Programm.* 5, 88–124.

Elnitski, L., Jin, V. X., Farnham, P. J., and Jones, S. J., 2006. Locating mammalian transcription factor binding sites: A survey of computational and experimental techniques. *Genome Res.* gr.4140006.

Euler, L., 1736. Solutio problematis ad geometriam situs pertinentis. *Comment. Acad. Sci. U. Petrop.* 8, 128–140.

Feller, W., 1968. *An Introduction to Probability and its Applications*. John Wiley & Sons, New York.

Fickett, J. W., 1996. Coordinate positioning of MEF2 and myogenin binding sites. *Gene* 172, GC19–GC32.

Fitzwater, T. and Polisky, B., 1996. A SELEX primer. *Methods Enzymol* 267, 275–301.

Fleiss, J. L., Levin, B., and Paik, M. C., 2003. *Statistical Methods for Rates and Proportions*. John Wiley & Sons, NY.

Fried, M. and Crothers, D., 1981. Equilibria and kinetics of lac repressor-operator interactions by polyacrylamide gel electrophoresis. *Nucleic Acids Res* 9, 6505–25.

Frith, M., Li, M., and Weng, Z., 2003. Cluster-buster: Finding dense clusters of motifs in dna sequences. *Nucleic Acids Res* 31, 3666–3668.

Frith, M. C., Fu, Y., Yu, L., Chen, J.-F., Hansen, U., and Weng, Z., 2004. Detection of functional DNA motifs via statistical over-representation. *Nucl. Acids Res.* 32, 1372–1381.

Frith, M. C., Hansen, U., and Weng, Z., 2001. Detection of cis -element clusters in higher eukaryotic DNA. *Bioinformatics* 17, 878–889.

Frith, M. C., Spouge, J. L., Hansen, U., and Weng, Z., 2002. Statistical significance of clusters of motifs represented by position specific scoring matrices in nucleotide sequences. *Nucleic Acids Res.* 30, 3214–3224.

Fu, J. and Koutras, M., 1994. Distribution Theory Of Runs: A Markov Chain Approach. *J. Am. Stat. Assoc.* 89, 1050–1058.

Fudos, I., Pitoura, E., and Szpankowski, W., 1996. On Pattern Occurrences in a Random Text. *Information Processing Letters* 57, 307–312.

Galas, D. and Schmitz, A., 1978. DNAse footprinting: a simple method for the detection of protein-DNA binding specificity. *Nucleic Acids Res* 5, 3157–70.

Galas, D. J., Eggert, M., and Waterman, M., 1985. Rigorous pattern-recognition methods for DNA sequences. Analysis of promoter sequences from Escherichia coli. *J. Mol. Biol.* 186, 117–128.

Garner, M. and Revzin, A., 1981. A gel electrophoresis method for quantifying the binding of proteins to specific DNA regions: application to components of the Escherichia coli lactose operon regulatory system. *Nucleic Acids Res* 9, 3047–60.

Gazit, B. and Cedar, H., 1980. Nuclease sensitivity of active chromatin. *Nucl. Acids Res.* 8, 5143–5155.

Gentleman, J. F., 1994. The Distribution of the Frequency of Subsequences in Alphabetic Sequences, as Exemplified by Deoxyribonucleic Acid. *Applied Statistics* 43, 401–409.

Gentleman, J. F. and Mullin, R. C., 1989. The distribution of the frequency of occurrence of nucleotide subsequences, based on their overlap capability. *Biometrics* 45, 35–52.

Georgi, B. and Schliep, A., 2006. Context-specific independence mixture modeling for positional weight matrices. *Bioinformatics* 22, e166–73.

Gorodkin, J., Heyer, L. J., Brunak, S., and Stormo, G. D., 1997. Displaying the information contents of structural rna alignments: the structure logos. *Comput Appl Biosci* 13, 583–586.

Goulden, I. and Jackson, D., 1983. *Combinatorial Enumeration*. Wiley.

Grad, Y. H., Roth, F. P., Halfon, M. S., and Church, G. M., 2004. Prediction of similarly acting cis-regulatory modules by subsequence profiling and comparative genomics in Drosophila melanogaster and D.pseudoobscura. *Bioinformatics* 20, 2738–50.

Graham, R., Knuth, D., and Patashnik, O., 1994. *Concrete Mathematics*. Addison-Wesley, Reading, MA, 2nd edition.

Gross, D. and Garrard, W., 1988. Nuclease hypersensitive sites in chromatin. *Annu. Rev. Biochem.* 57, 159–197.

Grundy, W., Bailey, T., and Elkan, C., 1996. ParaMEME: a parallel implementation and a web interface for a DNA and protein motif discovery tool. *Comput. Appl. Biosci.* 12, 303–310.

GuhaThakurta, D., 2006. Computational identification of transcriptional regulatory elements in dna sequence. *Nucleic Acids Res.* 34, 3585–3598.

GuhaThakurta, D. and Stormo, G. D., 2001. Identifying target sites for cooperatively binding factors. *Bioinformatics* 17, 608–621.

Guibas, L. and Odlyzko, A., 1980. Long repetitive patterns in random sequences. *Z. Wahrscheinlichkeitstheorie* 53, 242–262.

Guibas, L. and Odlyzko, A., 1981a. Periods in strings. *J. Comb. Theory* 30, 19–42.

Guibas, L. J. and Odlyzko, A. M., 1981b. String overlaps, pattern matching, and nontransitive games. *J. Comb. Theory* Ser. A 30, 183–208.

Gunewardena, S. and Zhang, Z., 2008. A hybrid model for robust detection of transcription factor binding sites. *Bioinformatics* 24, 484–491.

Gupta, M. and Liu, J. S., 2005. De novo cis-regulatory module elicitation for eukaryotic genomes. *Proceedings of the National Academy of Sciences* 102, 7079–7084.

Gupta, S., Stamatoyannopoulos, J. A., Bailey, T. L., and Noble, W. S., 2007. Quantifying similarity between motifs. *Genome Biol* 8, R24.

Gusfield, D., 1997. *Algorithms on Strings, Trees, and Sequences - Computer Science and Computational Biology*. Cambridge University Press.

Hanlon, S. E. and Lieb, J. D., 2004. Progress and challenges in profiling the dynamics of chromatin and transcription factor binding with DNA microarrays. *Curr Opin Genet Dev* 14, 697–705.

Hannenhalli, S. and Levy, S., 2002. Predicting transcription factor synergism. *Nucl. Acids Res.* 30, 4278–4284.

Hannenhalli, S. and Wang, L.-S., 2005. Enhanced position weight matrices using mixture models. *Bioinformatics* 21 Suppl 1, i204–12.

Harbison, C. T., Gordon, D. B., Lee, T. I., Rinaldi, N. J., Macisaac, K. D., Danford, T. W., Hannett, N. M., Tagne, J.-B., Reynolds, D. B., Yoo, J., Jennings, E. G., Zeitlinger, J., Pokholok, D. K., Kellis, M., Rolfe, P. A., Takusagawa, K. T., Lander, E. S., Gifford, D. K., Fraenkel, E., and Young, R. A., 2004. Transcriptional regulatory code of a eukaryotic genome. *Nature* 431, 99–104.

Hertz, G., Hartzell, G., and Stormo, G., 1990. Identification of consensus patterns in unaligned DNA sequences known to be functionally related. *Comput. Appl. Biosci.* 6, 81–92.

Hertz, G. and Stormo, G., 1999. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics* 15, 563–577.

Heumann, J., Lapedes, A., and Stormo, G., 1994. Neural networks for determining protein specificity and multiple alignment of binding sites. *Intell. Sys. Mol. Biol.* 2, 188–194.

Hu, J., Li, B., and Kihara, D., 2005. Limitations and potentials of current motif discovery algorithms. *Nucleic Acids Res* 33, 4899–913.

Hubbard, T., Andrews, D., Caccamo, M., Cameron, G., Chen, Y., Clamp, M., Clarke, L., Coates, G., Cox, T., Cunningham, F., Curwen, V., Cutts, T., Down, T., Durbin, R., Fernandez-Suarez, X., Gilbert, J., Hammond, M., Herrero, J., Hotz, H., Howe, K., Iyer, V., Jekosch, K., Kahari, A., Kasprzyk, A., Keefe, D., Keenan, S., Kokocinsci, F., London, D., Longden, I., McVicker, G., Melsopp, C., Meidl, P., Potter, S., Proctor, G., Rae, M., Rios, D., Schuster, M., Searle, S., Severin, J., Slater, G., Smedley, D., Smith, J., Spooner, W., Stabenau, A., Stalker, J., Storey, R., Trevanion, S., Ureta-Vidal, A., Vogel, J., White, S., Woodwark, C., and Birney, E., 2005. Ensembl 2005. *Nucleic Acids Res.* 33, D447–53.

IUPAC-IUB Commission on Biochemical Nomenclature, 1970. Abbreviations and Symbols for Nucleic Acids, Polynucleotides and their Constituents. Recommendations 1970. *J. Biol. Chem.* 245, 5171–5176.

Iyer, V., Horak, C., Scafe, C., Botstein, D., Snyder, M., and Brown, P., 2001. Genomic binding sites of the yeast cell-cycle transcription factors SBF and MBF. *Nature* 409, 533–8.

Jensen, L. and Knudsen, S., 2000. Automatic discovery of regulatory patterns in promoter regions based on whole cell expression data and functional annotation. *Bioinformatics* 16, 326–33.

Johnson, N. J., Kotz, S., and Kemp, A. W., 1995. *Univariate Discrete Distributions*. Wiley.

Kadonaga, J. and Tjian, R., 1986. Affinity purification of sequence-specific DNA binding proteins. *Proc Natl Acad Sci U S A* 83, 5889–93.

Kandel, Matias, Unger, and Winkler, 1996. Shuffling Biological Sequences. *DAMATH: Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science* 71.

Kang, S.-H. L., Vieira, K., and Bungert, J., 2002. Combining chromatin immunoprecipitation and DNA footprinting: a novel method to analyze protein-DNA interactions in vivo. *Nucleic Acids Res* 30, e44.

Keich, U., 2005. sFFT: A Faster Accurate Computation of the p-Value of the Entropy Score. *J. Comput. Biol.* 12, 416–430.

Kemp, C. D., 1967. 'Stuttering-Poisson' distributions. *Journal of the Statistical and Social Enquiry Society of Ireland* 21, 151–157.

Kielbasa, S. M., Gonze, D., and Herzel, H., 2005. Measuring similarities between transcription factor binding sites. *BMC Bioinformatics* 6, 237.

Kim, T. H., Barrera, L. O., Zheng, M., Qu, C., Singer, M. A., Richmond, T. A., Wu, Y., Green, R. D., and Ren, B., 2005. A high-resolution map of active promoters in the human genome. *Nature* 436, 876–80.

Kleffe, J. and Langbecker, U., 1990. Exact computation of pattern probabilities in random sequences generated by Markov chains. *Comput. Appl. Biosci.* 6, 347–353.

Klingenhoff, A., Frech, K., Quandt, K., and Werner, T., 1999. Functional promoter modules can be detected by formal models independent of overall nucleotide sequence similarity. *Bioinformatics* 15, 180–186.

Klose, R. and Bird, A., 2006. Genomic DNA methylation: the mark and its mediators. *Trends Biochem. Sci.* 31, 89–97.

Klug, J., 1997. Ku autoantigen is a potential major cause of nonspecific bands in electrophoretic mobility shift assays. *Biotechniques* 22, 212–4, 216.

Korn, L., Queen, C., and Wegman, M., 1977. Computer analysis of nucleic acid regulatory sequences. *Proc Natl Acad Sci U S A* 74, 4401–5.

Koutras, M., 1997. Waiting times and number of appearances of events in a sequence of discrete random variables. In Balakrishnan, N., ed., *Advances in combinatorial methods and applications to probability and statistics*, 253–261. Statistics and Industry and Technology Series, Birkhauser, Boston.

Krivan, W., 2004. Searching for transcription factor binding site clusters: how true are true positives? *J Bioinform Comput Biol* 2, 413–6.

Kullback, S., 1959. *Information Theory and Statistics*. John Wiley & Sons, New York, USA.

Kunsch, C., Ruben, S. M., and Rosen, C. A., 1992. Selection of optimal kappa B/Rel DNA-binding motifs: interaction of both subunits of NF-kappa B with DNA is required for transcriptional activation. *Mol Cell Biol* 12, 4412–4421.

Lawrence, C., Altschul, S., Boguski, M., Liu, J., Neuwald, A., and Wootton, J., 1993. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science* 262, 208–14.

Lawrence, C. and Reilly, A., 1990. An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins* 7, 41–51.

Lawson, G., Knoll, B., March, C., Woo, S., Tsai, M., and O'Malley, B., 1982. Definition of 5' and 3' structural boundaries of the chromatin domain containing the ovalbumin multigene family. *J. Biol. Chem.* 257, 1501–1507.

Li, N. and Tompa, M., 2006. Analysis of computational approaches for motif discovery. *Algorithms Mol Biol* 1, 8.

Li, S., 1980. A martingale approach to the study of occurrence of sequence patterns in repeated experiments. *Ann. Prob.* 8, 1171–1176.

Liefooghe, A., Touzet, H., and Varré, J.-S., 2006. Large scale matching for position weight matrices. In *Lecture Notes in Computer Science - Combinatorial Pattern Matching*, 401–412.

Lifanov, A., Makeev, V., Nazina, A., and Papatsenko, D., 2003. Uniform clusters in drosophila. *Genome Res* 13, 579–588.

Lim, L. P., Lau, N. C., Garrett-Engele, P., Grimson, A., Schelter, J. M., Castle, J., Bartel, D. P., Linsley, P. S., and Johnson, J. M., 2005. Microarray analysis shows that some microRNAs downregulate large numbers of target mRNAs. *Nature* 433, 769–773.

Liu, J. S., Lawrence, C. E., and Neuwald, A. F., 1990. Bayesian Models for Multiple Local Sequence Alignment and Gibbs Sampling Strategies. *J. Am. Stat. Assoc.* 95.

Liu, X., Brutlag, D., and Liu, J., 2001. BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. *Pac Symp Biocomput* 127–38.

Lonardi, S., 2001. *Global detectors of unusual words design, implementation and application to pattern discovery in biosequences*. Ph.D. thesis, Purdue University, West Lafayette, Indiana.

Lothaire, M., 1983. *Combinatorics on Words*. Addison-Wesley.

Lundstrom, R., 1990. *Stochastic models and statistical methods for DNA sequence data*. Ph.D. thesis, University of Utah.

Ma, B., Tromp, J., and Li, M., 2002. Patternhunter: faster and more sensitive homology search. *Bioinformatics* 18, 440–445.

MacIsaac, K. D. and Fraenkel, E., 2006. Practical Strategies for Discovering Regulatory DNA Sequence Motifs. *PLoS Computational Biology* 2, e36.

MacKay, D. J. C., 2003. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.

Mahony, S., Auron, P. E., and Benos, P. V., 2007. DNA familial binding profiles made easy: comparison of various motif alignment and clustering strategies. *PLoS Comput Biol* 3, e61.

Mahony, S., Golden, A., Smith, T. J., and Benos, P. V., 2005. Improved detection of DNA motifs using a self-organized clustering of familial binding profiles. *Bioinformatics* 21, i283–291.

Manke, T., Dieterich, C., and Vingron, M., 2005. Detecting Functional Modules of Transcription Factor Binding Sites in the Human Genome. In *Lecture Notes in Computer Science*. Springer Berlin/Heidelberg.

Markstein, M., Markstein, P., Markstein, V., and Levine, M., 2002. Genome-wide analysis of clustered dorsal binding sites identifies putative target genes in the drosophila embryo. *PNAS* 99, 763–768.

Marschall, T. and Rahmann, S., 2008. Probabilistic Arithmetic Automata and Their Application to Pattern Matching Statistics. In Ferragina, P. and Landau, G. M., eds., *CPM*, 95–106.

Maston, G. A., Evans, S. K., and Green, M. R., 2006. Transcriptional Regulatory Elements in the Human Genome. *Annu. Rev. Genomics Hum. Genet.* 7, 29–59.

Matys, V., Fricke, E., Geffers, R., Gossling, E., Haubrock, M., Hehl, R., Hornischer, K., Karas, D., Kel, A. E., Kel-Margoulis, O. V., Kloos, D.-U., Land, S., Lewicki-Potapov, B., Michael, H., Munch, R., Reuter, I., Rotert, S., Saxel, H., Scheer, M., Thiele, S., and Wingender, E., 2003. TRANSFAC(R): transcriptional regulation, from patterns to profiles. *Nucleic Acids Res.* 31, 374–378.

Maxam, A. M. and Gilbert, W., 1977. A new method for sequencing dna. *Proc Natl Acad Sci U S A* 74, 560–564.

McArthur, M., Gerum, S., and Stamatoyannopoulos, G., 2001. Quantification of DNaseI-sensitivity by real-time PCR: quantitative analysis of DNaseI-hypersensitivity of the mouse beta-globin LCR. *J Mol Biol* 313, 27–34.

McGuire, A., Hughes, J., and Church, G., 2000. Conservation of DNA regulatory motifs and discovery of new motifs in microbial genomes. *Genome Res* 10, 744–57.

Mengeritsky, G. and Smith, T., 1987. Recognition of characteristic patterns in sets of functionally equivalent DNA sequences. *Comput. Appl. Biosci.* 3, 223–227.

Mirny, L. A. and Gelfand, M. S., 2002. Structural analysis of conserved base pairs in protein-dna complexes. *Nucleic Acids Res* 30, 1704–1711.

Morgenstern, B., 2004. DIALIGN: multiple DNA and protein sequence alignment at BiBiServ. *Nucleic Acids Res* 32, W33–6.

Müller, F., Williams, D., Kobolák, J., Gauvry, L., Goldspink, G., Orbán, L., and Maclean, N., 1997. Activator effect of coinjected enhancers on the muscle-specific expression of promoters in zebrafish embryos. *Mol Reprod Dev* 47, 404–12.

Myers, L. and Kronberg, R., 2000. Mediator of transcriptional regulation. *Annu. Rev. Biochem.* 69, 729–749.

Nagarajan, N., Jones, N., and Keich, U., 2005. Computing the p-value of the information content from an alignment of multiple sequences. *Bioinformatics* 21, i311–318.

Narlikar, L. and Hartemink, A. J., 2006. Sequence features of DNA binding sites reveal structural class of associated transcription factor. *Bioinformatics* 22, 157–163.

Neuwald, A., Liu, J., and Lawrence, C., 1995. Gibbs motif sampling: detection of bacterial outer membrane protein repeats. *Protein Sci* 4, 1618–32.

Nicodeme, P., Salvy, B., and Flajolet, P., 1999. Motif statistics. In *ESA'99 volume 1643 of Lecture Notes in Computer Science*, 194–211. Springer-Verlag.

Niven, I., 1969. Formal power series. *American Mathematical Monthly* 76, 871–889.

Noonan, J. and Zeilberger, D., 1999. The Goulden-Jackson Method: Extensions, Applications and Implementations. *J. of Difference Equations and Applications 5* 4-5, 355–377.

Notredame, C., 2002. Recent progress in multiple sequence alignment: a survey. *Pharmacogenomics* 3, 131–144.

Notredame, C., Higgins, D. G., and Heringa, J., 2000. T-coffee: a novel method for fast and accurate multiple sequence alignment. *Journal of Molecular Biology* 302, 205–217.

Oliphant, A., Brandl, C., and Struhl, K., 1989. Defining the sequence specificity of DNA-binding proteins by selecting binding sites from random-sequence oligonucleotides: analysis of yeast GCN4 protein. *Mol Cell Biol* 9, 2944–9.

Orlando, V., 2000. Mapping chromosomal proteins in vivo by formaldehyde-crosslinked-chromatin immunoprecipitation. *Trends in Biochemical Sciences* 25, 99–104.

Osawa, S., Jukes, T. H., Watanabe, K., and Muto, A., 1992. Recent evidence for evolution of the genetic code. *Microbiol Rev* 56, 229–264.

Pabo, C. and Sauer, R., 1984. Protein-DNA recognition. *Annu. Rev. Biochem.* 53, 293–321.

Papatsenko, D., 2007. Clusterdraw web server: a tool to identify and visualize clusters of binding motifs for transcription factors. *Bioinformatics* 23, 1032–1034.

Papatsenko, D., Makeev, V., Lifanov, A., Regnier, M., Nazina, A., and Desplan, C., 2002. Extraction of functional binding sites from unique regulatory regions: The drosophila early developmental enhancers. *Genome Research* 12, 470–481. [Preliminary version in Drosophila Workshop, Washington 2001].

Pape, U. J., Grossmann, S., Hammer, S., Sperling, S., and Vingron, M., 2006. A new statistical model to select target sequences bound by transcription factors. *Genome Informatics* 17, 134–140.

Pape, U. J., Klein, H., and Vingron, M., 2008a. Statistical detection of co-operative transcription factors with similarity adjustment. Accepted by GCB 2008.

Pape, U. J., Rahmann, S., Richard, H., and Vingron, M., 2008b. Quality of Binding Site Representation by Position Frequency Matrices and Threshold Optimization. Submitted to Bioinformatics.

Pape, U. J., Rahmann, S., Sun, F., and Vingron, M., 2008c. Compound Poisson approximation of number of occurrences of a Position Frequency Matrix (PFM) on both strands. *J. Comput. Biol.* 15, 547–564.

Pape, U. J., Rahmann, S., and Vingron, M., 2008d. Natural Similarity Measures between Position Frequency Matrices with an Application to Clustering. *Bioinformatics* 24, 350–357.

Pape, U. J. and Vingron, M., 2008. Statistics for Co-Occurrence of DNA Motifs. In Chiquet, J., Glaz, J., Limnios, N., and Moyal, P., eds., *Proceedings of the 4th International Workshop on Applied Probability.*

Pavesi, G., Mauri, G., and Pesole, G., 2004. In silico representation and discovery of transcription factor binding sites. *Brief Bioinform* 5, 217–236.

Pearl, J., 1988. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, San Mateo, Ca.

Pevzner, P. A., Borodovsky, M., and Mironov, A. A., 1989. Linguistics of nucleotide sequences. i: The significance of deviations from mean statistical characteristics and prediction of the frequencies of occurrence of words. *J Biomol Struct Dyn* 6, 1013–1026.

Pietrokovski, S., 1996. Searching databases of conserved sequence regions by aligning protein multiple-alignments [published erratum appears in nucleic acids res 1996 nov 1;24(21):4372]. *Nucleic Acids Res.* 24, 3836–3845.

Pilpel, Y., Sudarsanam, P., and Church, G., 2001. Identifying regulatory networks by combinatorial analysis of promoter elements. *Nat Genet* 29, 153–9.

Pólya, G., 1930. Sur quelques points de la théorie des probabilités. *Annales de l'Institut H. Poincare* 1, 117–161.

Prakash, A., Blanchette, M., Sinha, S., and Tompa, M., 2004. Motif discovery in heterogeneous sequence data. *Pac Symp Biocomput* 348–59.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., 1992. *Numerical Recipes in C. The Art of Scientific Computing.* Cambridge University Press, Cambridge, UK.

Pribnow, D., 1975. Nucleotide sequence of an RNA polymerase binding site at an early T7 promoter. *Proc. Natl. Acad. Sci.* 72, 784–788.

Prum, B., Rodolphe, F., and de Turckheim, E., 1995. Finding words with unexpected frequencies in deoxyribonucleic acid sequences. *J. Roy. Statist. Soc. Ser. B-Methodological* 57, 205–220.

Queen, C., Wegman, M., and Korn, L., 1982. Improvements to a program for DNA analysis: a procedure to find homologies among many sequences. *Nucleic Acids Res* 10, 449–56.

Rahmann, S., 2000. *Word Statistics in Random Texts and Applications to Computational Molecular Biology.* Master's thesis, Mathematische Fakultät der Ruprecht-Karls-Universität Heidelberg.

Rahmann, S., 2003. Dynamic programming algorithms for two statistical problems in computational biology. In *Proceedings of the 3rd Workshop of Algorithms in Bioinformatics (WABI)*, 151–164. Springer Verlag, Heidelberg.

Rahmann, S., Müller, T., and Vingron, M., 2003. On the power of profiles for transcription factor binding site detection. *Stat. Appl. Genet. Mo. B.* 2.

Rajewsky, N., Vergassola, M., Gaul, U., and Siggia, E. D., 2002. Computational detection of genomic cis-regulatory modules applied to body patterning in the early drosophila embryo. *BMC Bioinformatics* 3, 30.

Rateitschak, K., Müller, T., and Vingron, M., 2004. Annotating significant pairs of transcription factor binding sites in regulatory DNA. *In Silico Biology* 4, 479–487.

Rebeiz, M., Reeves, N., and Posakony, J., 2002. Score: a computational approach to the identification of cis-regulatory modules and target genes in whole-genome sequence data. site clustering over random expectation. *Proc Natl Acad Sci USA* 99, 9888–93. Epub 2002 Jul 09.

Régnier, M., 2000. A unified approach to word occurrence probabilities. *Discrete Appl. Math.* 104, 259–280.

Régnier, M. and Denise, A., 2004. Rare Events and Conditional Events on Random Strings. *Discrete Mathematics and Theoretical Computer Science* 6, 191–214.

Régnier, M. and Szpankowski, W., 1998. On pattern frequency occurrences in a markovian sequence. *Algorithmica* 22, 631–649.

Reinert, G. and Schbath, S., 1998. Compound Poisson and Poisson process approximations for occurrences of multiple words in Markov chains. *J Comput Biol* 5, 223–253.

Reinert, G. and Schbath, S., 1999. Compound poisson approximations for occurrences of multiple words. In Seiller-Moiseiwitsch, F., ed., *Statistics in Molecular Biology and Genetics*, 257–275,. IMS Lecture Notes.

Reinert, G., Schbath, S., and Waterman, M., 2005. Probabilistic and Statistical Properties of Finite Words in Finite Sequences. In Berstel, J. and Perrin, D., eds., *Applied Combinatorics on Words*. Cambridge University Press.

Reinert, G., Schbath, S., and Waterman, M. S., 2000. Probabilistic and statistical properties of words: an overview. *J. Comput. Biol.* 7, 1–46.

Ren, B., Robert, F., Wyrick, J. J., Aparicio, O., Jennings, E. G., Simon, I., Zeitlinger, J., Schreiber, J., Hannett, N., Kanin, E., Volkert, T. L., Wilson, C. J., Bell, S. P., and Young, R. A., 2000. Genome-Wide Location and Function of DNA Binding Proteins. *Science* 290, 2306–2309.

Rice, J., 1995. *Mathematical Statistics and Data Analysis*. Duxbury Press.

Rivals, E. and Rahmann, S., 2003. Combinatorics of periods in strings. *Journal of Combinatorial Theory, Series A* 104, 95–113.

Robin, S., 2002. A compound Poisson model for word occurrences in DNA sequences. *J. Roy. Stat. Soc. C-App.* 51, 437–451.

Robin, S. and Daudin, J., 1999. Exact Distribution of Word Occurrences in a Random Sequence Letters. *J. Applied Prob.* 36, 179–193.

Robin, S. and Daudin, J.-J., 2001. Exact distribution of the distances between any occurrences of a set of words. *Ann. Inst. Statist. Math.* 4, 895–905.

Robin, S., Rodolphe, F., and Schbath, S., 2005. *DNA, Words and Models - Statistics of Exceptional Words*. Press Syndicate Cambridge.

Robin, S. and Schbath, S., 2001. Numerical comparison of several approximations of the word count distribution in random sequences. *J. Comput. Biol.* 8, 349–359.

Roepcke, S., Grossmann, S., Rahmann, S., and Vingron, M., 2005. T-Reg Comparator: an analysis tool for the comparison of position weight matrices. *Nucleic Acids Res.* 33, W438–441.

Roquain, E. and Schbath, S., 2007. Improved compound Poisson approximation for the number of occurrences of multiple words in a stationary Markov chain. *Adv Appl Prob* 39, 128–140.

Rosenberg, M. and Court, D., 1979. Regulatory sequences involved in the promotion and termination of RNA transcription. *Annu. Rev. Genet.* 13, 319–353.

Salvy, B. and Zimmermann, P., 1994. GFUN: a Maple package for the manipulation of generating and holonomic functions in one variable. *ACM Trans. Math. Softw.* 20, 163–177.

Sandelin, A., Alkema, W., Engstrom, P., Wasserman, W. W., and Lenhard, B., 2004a. JASPAR: an open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Res.* 32, D91–D94.

Sandelin, A., Bailey, P., Bruce, S., Engstrom, P. G., Klos, J. M., Wasserman, W. W., Ericson, J., and Lenhard, B., 2004b. Arrays of ultraconserved non-coding regions span the loci of key developmental genes in vertebrate genomes. *BMC Genomics* 5, 99.

Sandelin, A. and Wasserman, W. W., 2004. Constrained binding site diversity within families of transcription factors enhances pattern discovery bioinformatics. *J. Mol. Biol.* 338, 207–215.

Sandve, G. K. and Drabløs, F., 2006. A survey of motif discovery methods in an integrated framework. *Biol Direct* 1, 11.

Sanger, F., Nicklen, S., and Coulson, A. R., 1977. DNA sequencing with chain-terminating inhibitors. *Proc Natl Acad Sci U S A* 74, 5463–5467.

Santibanez-Koref, M., 1987. *Statistische Untersuchungen an DNS-Sequenzen und ein Verfahren zum mehrfachen Sequenzvergleich*. Ph.D. thesis, Humboldt Universität zu Berlin.

Schbath, S., 1995a. Compound Poisson approximation of word counts in DNA sequences. *ESAIM: Probability and Statistics* 1, 1–16.

Schbath, S., 1995b. *Étude asymptotique du nombre d'occurrences d'un mot dans une chaîne de Markov et application á la recherche de mots de fréquence exceptionnelle dans les séquences d'ADN*. Ph.D. thesis, Université René Descartes, Paris V.

Schbath, S., Prum, B., and Turckheim, E., 1995. Exceptional motifs in different Markov chain models for statistical analysis of DNA sequences. *J. Comp. Biol.* 2, 417–437.

Scheper, G. C., van der Knaap, M. S., and Proud, C. G., 2007. Translation matters: protein synthesis defects in inherited disease. *Nat Rev Genet* 8, 711–723.

Schneider, T. D., 1997. Information content of individual genetic sequences. *J Theor Biol* 189, 427–441.

Schneider, T. D. and Stephens, R. M., 1990. Sequence logos: a new way to display consensus sequences. *Nucleic Acids Res.* 18, 6097–6100.

Schones, D. E., Sumazin, P., and Zhang, M. Q., 2005. Similarity of position frequency matrices for transcription factor binding sites. *Bioinformatics* 21, 307–313.

Schwarz, G., 1978. Estimating the dimension of a model. *Ann. Statistics* 6, 461–464.

Shannon, C. E., 1948. A mathematical theory of communication. *Bell System Technical Journal* 27, 379–423 and 623–656.

Shannon, C. E. and Weaver, W., 1949. *The mathematical theory of communication.* University of Illinois Press.

Siemen, H., Nix, M., Endl, E., Koch, P., Itskovitz-Eldor, J., and Br?stle, O., 2005. Nucleofection of human embryonic stem cells. *Stem Cells Dev* 14, 378–83.

Sinha, S., Blanchette, M., and Tompa, M., 2004. PhyME: a probabilistic algorithm for finding motifs in sets of orthologous sequences. *BMC Bioinformatics* 5, 170.

Sosinsky, A., Bonin, C., Mann, R., and Honig, B., 2003. Target explorer: an automated tool for the identification of new target genes for a specified set of transcription factors. *Nucleic Acids Research* 31, 3589–3592.

Spiegelman, B. and Heinrich, R., 2004. Biological control through regulated transcriptional coactivators. *Cell* 119, 157–167.

Staden, R., 1984. Computer methods to locate signals in nucleic acid sequences. *Nucleic Acids Res.* 12, 505–19.

Staden, R., 1989. Methods for calculating the probabilities of finding patterns in sequences. *Comput. Appl. Biosci.* 5, 89–96.

Stefanov, V., Robin, S., and Schbath, S., 2007. Waiting times for clumps of patterns and for structured motifs in random sequences. *Discrete Applied Mathematics* 155, 868–880.

Stormo, G., 1990. Consensus patterns in DNA. *Methods Enzymol.* 183, 211–221.

Stormo, G. and Hartzell, G., 1989. Identifying protein-binding sites from unaligned DNA fragments. *Proc Natl Acad Sci U S A* 86, 1183–7.

Stormo, G., Schneider, T., and Gold, L., 1986. Quantitative analysis of the relationship between nucleotide sequence and functional activity. *Nucleic Acids Res* 14, 6661–79.

Stormo, G. D., 1998. Information content and free energy in dna–protein interactions. *J Theor Biol* 195, 135–137.

Stormo, G. D., 2000. DNA binding sites: representation and discovery. *Bioinformatics* 16, 16–23.

Strauss, W., 1996. Transfection of mammalian cells via lipofection. *Methods Mol Biol* 54, 307–27.

Sun, 2002. Sun Grid Engine 5.3 Administration and User's Guide. Sun Systems Inc.

Suzuki, M. and Yagi, N., 1994. DNA recognition code of transcription factors in the helix-turn-helix, probe helix, hormone receptor, and zinc finger families. *Proc Natl Acad Sci U S A* 91, 12357–61.

Tang, W. and Perry, S. E., 2003. Binding site selection for the plant MADS domain protein AGL15: an in vitro and in vivo study. *J. Biol. Chem.* 278, 28154–28159.

Tanushev, M. and Arratia, R., 1997. Central limit theorem for renewal theory for several patterns. *J. Comp. Biol.* 4, 35–44.

Tavaré, S. and Song, B., 1989. Codon preference and primary sequence structure in protein-coding regions. *Bull Math Biol* 51, 95–115.

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, 1113–1122.

Thompson, J., Higgins, D., and Gibson, T., 1994. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res* 22, 4673–80.

Thompson, W., Rouchka, E. C., and Lawrence, C. E., 2003. Gibbs Recursive Sampler: finding transcription factor binding sites. *Nucleic Acids Res* 31, 3580–5.

Tompa, M., Li, N., Bailey, T. L., Church, G. M., Moor, B. D., 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. *Nat Biotechnol* 23, 137–144.

Touzet, H. and Varré, J.-S., 2007. Efficient and accurate p-value computation for position weight matrices. *Algorithms Mol Biol* 2, 15.

Tsien, R., 1998. The green fluorescent protein. *Annu Rev Biochem* 67, 509–44.

Tuerk, C. and Gold, L., 1990. Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage T4 DNA polymerase. *Science* 249, 505–10.

van Helden, J., André, B., and Collado-Vides, J., 1998. Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *J. Mol. Biol.* 281, 827–842.

Wagner, A., 1997. A computational genomics approach to the identification of gene networks. *Nucleic Acids Res.* 25, 3594–3604.

Wagner, A., 1999. Genes regulated cooperatively by one or more transcription factors and their identification in whole eukaryotic genomes. *Bioinformatics* 15, 776–784.

Wang, T. and Stormo, G. D., 2003. Combining phylogenetic data with co-regulated genes to identify regulatory motifs. *Bioinformatics* 19, 2369–2380.

Wasserman, W. and Fickett, J., 1998. Identification of regulatory regions which confer muscle-specific gene expression. *J Mol Biol* 278, 167–81.

Wasserman, W. W. and Sandelin, A., 2004. Applied Bioinformatics for the Identification of Regulatory Elements. *Nature Reviews Genetics* 5, 276–287.

Waterman, M., Arratia, R., and Galas, D., 1984. Pattern recognition in several sequences: Consensus and alignment. *Bulletin of Mathematical Biology* 46, 515–527.

Waterman, M. S., 2000. *Introduction to Computational Biology*, chapter 12: Probability and Statistics for Sequence Patterns. Chapman & Hall/CRC.

Watson, J. and Crick, F., 1953. Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid. *Nature* 171, 737–738.

Wilf, H. S., 1994. *generatingfunctionology*. Academic Press, Inc.

Wilson, R., 1986. An Eulerian Trail through Königsberg. *J. Graph Th.* 10, 265–275.

Wright, W., Binder, M., and Funk, W., 1991. Cyclic amplification and selection of targets (CASTing) for the myogenin consensus binding site. *Mol Cell Biol* 11, 4104–10.

Wu, C., 1980. The 5' ends of Drosophila heat shock genes in chromatin are hypersensitive to DNase I. *Nature* 286, 854–60.

Wu, T. D., Nevill-Manning, C. G., and Brutlag, D. L., 2000. Fast probabilistic analysis of sequence function using scoring matrices. *Bioinformatics* 16, 233–244.

Yamauchi, K., 1991. The sequence flanking translational initiation site in protozoa. *Nucleic Acids Res* 19, 2715–20.

Yoo, J., Herman, L., Li, C., Krantz, S., and Tuan, D., 1996. Dynamic changes in the locus control region of erythroid progenitor cells demonstrated by polymerase chain reaction. *Blood* 87, 2558–67.

Yu, X., Lin, J., Zack, D. J., and Qian, J., 2006. Computational analysis of tissue-specific combinatorial gene regulation: predicting interaction between transcription factors in human tissues. *Nucleic Acids Res* 34, 4925–36.

Yuan, G.-C., Liu, Y.-J., Dion, M. F., Slack, M. D., Wu, L. F., Altschuler, S. J., and Rando, O. J., 2005. Genome-scale identification of nucleosome positions in S. cerevisiae. *Science* 309, 626–30.

Yuh, C.-H., Bolouri, H., and Davidson, E. H., 1998. Genomic Cis-Regulatory Logic: Experimental and Computational Analysis of a Sea Urchin Gene. *Science* 279, 1896–1902.

Zhang, J., Jiang, B., Li, M., Tromp, J., Zhang, X., and Zhang, M. Q., 2007. Computing exact P-values for DNA motifs. *Bioinformatics* 23, 531–537.

Zhang, M. and Marr, T., 1993. A weight array method for splicing signal analysis. *Comput Appl Biosci* 9, 499–509.

Zhao, X., Huang, H., and Speed, T. P., 2005. Finding short DNA motifs using permuted Markov models. *J Comput Biol* 12, 894–906.

Zhou, Q. and Wong, W. H., 2004. CisModule: De novo discovery of cis-regulatory modules by hierarchical mixture modeling. *Proceedings of the National Academy of Sciences* 101, 12114–12119.

Notation and Abbreviations

Notation

$\stackrel{d}{=}$	equality in distribution
$[z^n]g(z)$	nth coefficient of generating function $g(z)$, page 73
$\mathbf{1}[\cdot]$	equals to 1 if expression is true and otherwise 0
$\approx$	approximately
$\lvert \cdot \rvert$	size of a set and absolute value of a scalar
$\sim$	distributed as
$\binom{n}{k}$	binomial coefficient
$\mathbb{C}\text{ov}[\cdot, \cdot]$	covariance of two random variables, page 34
$\mathcal{CP}$	compound Poisson distribution, page 56
$\mathbb{E}[\cdot]$	expected value of a random variable
$\mathscr{P}(\cdot)$	power set
$\mathbb{V}[\cdot]$	variance of a random variable, page 34
$\mathfrak{A}$	alphabet, page 14
$\mathfrak{A}^+$	all possible words from alphabet with at least one letter
$\mathcal{A}$	set of compatible words, page 28
$\mathcal{B}(\cdot, \cdot)$	binomial distribution
$\mathbb{C}\text{or}(\cdot, \cdot)$	correlation between two random variables
$\mathcal{C}$	set of disjoint classes/groups of TFs, page 157
$\mathcal{L}(\cdot)$	distribution of a random variable
$\mathcal{M}(\cdot, \cdot)$	multinomial distribution, page 14
$\mathcal{N}(\cdot, \cdot)$	normal distribution with expected value and variance as parameters

$\mathbb{N}$	natural numbers $1, 2, 3, \ldots$
$\mathbb{P}(\cdot)$	probability of an event
$\mathcal{P}(\cdot)$	Poisson distribution with given rate
$\mathbb{P}_\mu(\cdot)$	probability of an event under the sequence model
$\mathbb{P}_\Pi(\cdot)$	probability of an event under the motif model Π, page 23
$\mathcal{S}(\cdot)$	set of discretized scores of a given PFM, see equation (5.3), page 90
$\mathcal{W}$	set of words with equal lengths, page 57
$\mathcal{X}$	set of sequences, page 18
α	probability of a type-I error, page 24
α_n	probability of at least one false positive in a sequence of length n, page 25
β	probability of a type-II error, page 24
$\epsilon_d(\cdot)$	overlap bit for a shift of d, see equation (3.6), page 34
$\epsilon_d(\cdot, \cdot)$	overlap bit between two words, see equation (3.31), page 58
η	period of a word, page 44
$\bar{\gamma}_d$	conditional probability of an overlapping occurrence after d positions given the preceding occurrence, see equation (5.10), page 98
$\gamma_d(\cdot)$	joint probability for two occurrences shifted by d positions, see equation (3.7), page 34
$\gamma_d(\cdot, \cdot)$	joint probability of overlapping occurrences of two words, page 59
κ	position in the motif, page 14
ℓ	length of motif, page 14
Λ	diagonal matrix containing eigenvalues, page 100
λ_i	ith eigenvalue, page 100
$\mu(\cdot)$	background probability for letter/word
$\boldsymbol{\mu}$	vector of expected values of number of occurrences of a set of words, see equation (3.34), page 63
$\tilde{\boldsymbol{\mu}}$	vector of expected values of number of clumps of a set of words, see equation (3.40), page 69

$\breve{\mu}(\cdot)$	probability for a heterogeneous clump of a given set of words, page 70
$\breve{\mu}_k(\cdot)$	probability of a heterogeneous clump of size k of a set of gievn words, see equation (3.43), page 71
μ	sequence model and vector of nucleotide frequencies, page 20
$\tilde{\mu}(\cdot)$	probability of a clump, page 45
$\tilde{\mu}_k(\cdot)$	probability of a clump of size k, see equation (3.27), page 55
Ω	matrix containing all overlap probabilities of a set of words, page 70
$\omega(\cdot)$	probability that an occurrence is no clump (thus, there is an overlapping occurrence to the left), see equation (3.18), page 45
$\omega_d(\cdot)$	probability that an occurrence is no clump given a preceding clump with a distance of d, see equation (3.23), page 47
$\omega_{\cdot,\cdot}$	overlap probability between two words, see equation (3.42), page 70
$\varpi_d(\cdot)$	covariance between clumps with distance d divided by the probability of a clump, page 48
ϕ	weight to reflect multiplicity of binding sites, page 166
Π	PFM, see equation (2.1), page 15
π_κ	multinomial probability vector characterizing one position of the motif, page 14
Ψ	PSM, page 16
$\psi_{\cdot,\cdot}$	score for given position and nucleotide, page 16
Ψ^*	re-ordered PSM, page 27
$\psi^*_{\kappa,r}$	term re-ordered PSM at position κ and rank r, page 27
$\mathcal{S}_t(\cdot)$	set of discretized scores, which reach a threshold t, of a given PFM, page 90
$\mho(\cdot)$	σ-algebra generated by a given set, page 33
Σ	covariance matrix for the number of occurrences of a set of words, see equation (3.35), page 63
σ^2	variance
$\sigma^2_{\cdot,\cdot}$	covariance of number of occurrences between two words, page 62

$\tilde{\sigma}^2$	variance of number of clumps, page 56
$\mathcal{T}$	set of TFs, page 121
$\boldsymbol{\theta}$	probability vector of clump size distribution, page 96
ϑ	rate of Poisson distribution
ϑ_k	Poisson rate for a k-clump, page 55
$\Upsilon'(\cdot)$	set of principal periods for a given word, page 44
$\Upsilon'_{\mathcal{W}}(\cdot,\cdot)$	set of principal periods between to words with respect to a set of words, see equation (3.41), page 70
$\Upsilon(\cdot)$	set of all periods for a given word, see equation (3.17), page 44
$\Upsilon(\cdot,\cdot)$	set of periods between two words, see equation (3.39), page 68
$\boldsymbol{\xi}$	vector of summed extension factors, page 100
ξ	summed extension factors: 'update-probability' of an additional overlapping occurrence, see equation (5.12), page 99
ξ_d	extension factors: 'update-probability' of an additional occurrence shifted by d positions, page 98
ζ	sum of second moments for all overlapping CRM windows, page 125
ζ_k	second moments of CRM indicator variables with a distance of k, page 125
$a(z)$	generating function of absence probabilities, see equation (4.6), page 78
B_i	neighborhood set for Chen-Stein bounds, page 32
$B_{\cdot,\cdot}$	neighborhood for Chen-Stein bounds for double indexed random variables
C_i	set of TFs, page 157
$d_{\mathrm{TV}}(\cdot,\cdot)$	total variation distance, see equation (3.2), page 32
d	usually, shift between two occurrences, page 34
$d(z)$	generating function of return probabilities of the first ℓ positions, page 76
$\tilde{f}^{(k)}(z)$	generating function of k clumps in a sequence, see equation (4.17), page 83
$f^{(k)}(z)$	generating function of k counts in a sequence, see equation (4.10), page 80

$f_{i,v}^{(k)}(\cdot)$	conditional probability to observe at least k occurrences in a sequence of length $n-i+1$ given that the sequence starts with v, see equation (3.15), page 39
$F_i(\cdot)$	probability to observe at least one occurrence in a sequence of length $n-i+1$, see equation (3.12), page 38
$F_i^{(k)}(\cdot)$	probability to observe at least k occurrences in a sequence of length $n-i+1$, see equation (3.14), page 38
$f_{i,v}(\cdot)$	conditional probability to observe at least one occurrence in a sequence of length $n-i+1$ given that the sequence starts with v, see equation (3.13), page 38
$G(\cdot,\cdot)$	sum of overlap probabilities of two words, page 62
$G(w)$	sum over overlap probabilities, page 41
I	index set for Chen-Stein bounds, page 32
i	usually, a position in a sequence, page 32
m	number of windows, page 122
$\tilde{\mathbf{N}}_n$	vector of random variables for number of clumps of a set of words, page 69
$\mathbf{N}_n$	vector of random variables for number of occurrences for a set of words, page 62
$\tilde{N}_n(\cdot)$	random variable for number of clumps in a sequence of length n, page 44
n	length of the sequence, page 20
$N_n^*(\cdot)$	number of occurrences in a sequence of length n ignoring sequence boundaries, page 114
$N_n(\cdot)$	random variable for the number of words in a sequence of length n, page 32
$N_{\mathbf{X}}(a)$	number of letters a in sequence $\mathbf{X}$, page 18
$N_{obs}(a)$	number of observed letters a, page 18
$O(\cdot)$	$f(x)$ is $O(g(x))$ as $x \to \infty$ if there is a constant c such that $\mid f(x) \mid \leq cg(x)$
$p(\cdot)$	probability of a CRM in a window of given length, see equation (6.1), page 114

P_κ	multinomial random variable characterizing one position of the motif, page 14
q	similarity threshold/quantile, page 158
$Q(\cdot \mid \cdot)$	quality of a PFM with respect to a given set of words, see equation (11.1), page 166
r	rank or rate
$r(z)$	generating function of return probabilities, see equation (4.3), page 76
S	random variable of the score, page 23
s	given score, page 23
$s(\cdot)$	score for a letter/word, page 23
$S(\cdot, \cdot)$	similarity between two objects based on the covariance, see equation (9.1), page 148
$S^{(\cdot)}$	random variable of the nucleotide-score at a given position, page 23
$S^{\max}(\cdot, \cdot)$	similarity between two objects based on maximum overlap probability, page 149
$S_j(\cdot)$	random variable of the score for a given PFM at position j, page 86
$\tilde{t}^{(k)}(z)$	generating function of waiting times till kth clump, see equation (4.16), page 83
$\tilde{T}_m$	random variable for the waiting time till the occurrence of the mth clump, page 49
$\underline{T}_k$	waiting time till the kth occurrence by considering the end of an occurrence, page 76
t	threshold for the score, page 22
$t(z)$	generating function of waiting times, see equation (4.5), page 77
$t^{(k)}(z)$	generating function of waiting times till the kth occurrence, see equation (4.9), page 80
$T_m(\cdot)$	waiting time until the mth occurrence, page 35
t_{bal}	balanced threshold, page 26
$\acute{v}(z)$	generating function of the non-overlapping inter-arrival times, see equation (4.14), page 83

$\acute{V}_k$	random variable for k non-overlapping inter-arrival times, see equation (4.12), page 82
$\grave{v}(z)$	generating function of the overlapping inter-arrival times, page 83
$\grave{V}_k$	random variable for k overlapping inter-arrival times, see equation (4.12), page 82
$\tilde{V}$	random variable of inter-arrival times of clumps, see equation (4.13), page 82
$\tilde{v}(z)$	generating function of the clump inter-arrival times, see equation (4.15), page 83
$v(z)$	generating function of inter-occurrence times, see equation (4.8), page 80
V_k	random variable of inter-occurrence times, see equation (4.7), page 79
W	random variable of the number of CRMs, see equation (6.13), page 122
w	a word from $\mathfrak{A}^+$ or window size
W_i	random indicator variable for a CRM of window i, page 122
$\mathbf{X}$	sequence of letters, page 18
$\tilde{x}$	number of clumps
x	number of occurrences
X_i	random variable for nucleotide distribution of a sequence at position i, page 20
$\breve{Y}_i(\cdot)$	random indicator variable for the start of a heterogeneous clump with no overlapping occurrences of any word of the set of words, see equation (3.36), page 64
$\underline{\tilde{Y}}_i$	random indicator for the end of the first occurrence of a clump, see equation (4.11), page 81
$\tilde{Y}_i(\cdot)$	random indicator variable for a clump starting at position i, see equation (3.16), page 44
$\underline{Y}_i$	random indicator variable for the end position of a word, see equation (4.1), page 75
$Y_i(\cdot)$	random indicator variable for the occurrence of a word, see equation (3.1), page 32
$\breve{Z}_k(\cdot)$	number of heterogeneous k-clumps of a given set of words, page 71

$\tilde{Z}_k(\cdot)$	number of clumps of size k for a given word, page 54
z	dummy variable for generating functions, page 73
$Z(\cdot)$	random variable for the clump size of a given word, page 53
Z_i	random variable for the size of the ith clump, page 53

Abbreviations

'A'	adenine, page 1
'C'	cytosine, page 1
CASTing	cyclic amplification and selection of targets, page 11
cDNA	complementary DNA, page 12
ChIP	chromatin-immunoprecipitation, page 11
CRM	*cis* regulatory module, page 2
DNA	desoxyribonucleic acid, page 1
EM	expectation maximization, page 17
EMSA	electrophoretic mobility shift assay, page 11
FN	false negatives, page 163
FP	false positives, page 163
'G'	guanine, page 1
GFP	green fluorescent protein, page 10
i.i.d.	independently and identically distributed, page 33
IUPAC	International Union of Pure and Applied Chemistry, page 13
LOOCV	leave-one-out cross-validation, page 159
PCM	position count matrix: contains nucleotide counts from multiple sequence alignment, page 15
PCR	polymerase chain reaction, page 10
PFM	Position Frequency Matrix, page 3
PSM	position score matrix: contains log-likelihood scores, page 16
PWM	position weight matrix: contains regularized frequencies, page 15

SELEX	systematic evolution of ligands by exponential enrichment, page 11
'T'	thymine, page 1
TF	transcription factor, page 2
TFBS	transcription factor binding site, page 2
TN	true negatives, page 163
TP	true positives, page 163
TSS	transcription start site, page 2

Index

W

Appendix A: Software availability

The program suite `MOtifSTAtistics` contains several programs to compute statistics developed in this thesis and is available as web-front end and stand-alone programs at http://mosta.molgen.mpg.de. After describing the compilation of the programs, each program and its parameters are presented. Since the computation of clustering of large set of PFMs demands lot of computational time, the corresponding programs support parallel computing (Sun, 2002) and multi-processor machines (Dagum and Menon, 1998). The implementation details to adapt to different parallel computing platforms are given next. For license issues, see last section.

Compilation

Compilation is done by standard GNU C++ compiler. To use multi-processors, Open MP (Dagum and Menon, 1998) has to be installed and the flag `-openmp` passed to the compiler. Compilation is started by

```
>make all
```

and to clean

```
>make clean
```

cstat

Returns the count statistics (see Chapter 5 and Chapter 7).

Call

```
$ ./cstat <gc> <[list:]transfac-file> <threshold-method> <threshold-parameter>
. <[regularize]> <[output]> <[sequence length]>
```

where the parameters have to be as followed:

- <gc>: gc content, e.g. '.4', for the background model
- <[file:]transfac-file>: file describing the position count matrices (PCM) in transfac format, e.g. 'data/A1.mat'. See data/A1.mat as an example. The program assumes the line tag ID to occur first. Next, it searches for P0, 01, 02 and so on until the next line does not contain the next number. Different PCMs in the file have to be separated by a line only containing '//'. If filename is preceded by list: one can pass a file containing a list of transfac filenames as parameter. Each file of the list is supposed to contain exactly one PCM.
- <threshold-method>: the treshold method (see Section 2.4.3)
  - typeI: set threshold such that typeI error is equal to threshold-parameter.
  - typeII: set threshold such that typeII error is equal to threshold-parameter.
  - balanced: set threshold such that typeI error equals typeII, threshold-parameter can be any number (is not used but has to be passed as parameter)
  - typeIext: set threshold to balanced threshold if it's possible such that the probability of a false positive on a sequence length of 500 is less or equal to the threshold-parameter. Otherwise, set typeI error equal to threshold-parameter.
  - threshold: threshold-parameter contains the threshold.
  - nrwords: define the number of words higher than the threshold (is only accurate if gc content is 50%.)
- <[regularize]>: if not set or set to a true value (1), the regularization method from Rahmann (2003). Otherwise, we just add pseudocounts (see Section 2.4.1 for details).
- <output>: if this parameter exist, the running time of the statistical calculation (without reading input/preparing PSSM, and so on) is printed. Depending on the choice of this parameter, following output is printed:
  - parameter: only xi, xi', xi'0, alpha, theta1
  - lambda: in addition: lambda1, lambda2
  - theta: in addition, all thetas until ¡ precision
  - rate: in addition, rate r (give sequence length in next parameter) without theta

- cpd: in addition, all $\mathbb{P}(X \geq x)$ until $<$ precision (give sequence length)

- <[sequence-length]>: length of the sequence

The typeI error is measured as the probability of at least one false positive in a region of length 500 (Pape *et al.*, 2006, α_{500}).

Examples

- Assumes gc-content of 40%, uses matrix given in data/matrixA.mat and sets the threshold to 30.

  ```
  ./cstat .4 data/matrixA.mat threshold 30
  ```

- Iterates over the transfac files given in data/matrix.list and sets for each matrix the threshold such that typeI error is equal to typeII error or (it not possible) the typeI error to .1.

  ```
  ./cstat .3 list:data/matrix.list typeIext .1
  ```

- Returns hit statistic for a sequence of length 10000 with gc-content 40% after setting the threshold for the non-regularized (only pseudo-counts added) such that the type I error is equal to 10%.

  ```
  ./cstat .4 data/matrixA.mat typeI .1 0 cpd 10000
  ```

sstat

Returns the similarity between PFMs (see Chapter 9).

Call

```
$ ./sstat <gc> <[list:]transfac-file> <threshold-method> <threshold-parameter>
. [<partial-execution>] [<return diagonal>] [<bregularize>]
```

Most parameters are the same as for cstat. The new parameters are:

- <partial-execution>: integer i: if not given, whole similarity matrix is computed. if given, only the ith and the n-i th line of the similarity matrix are computed and return in special format (to be read by scluster). if -1 then simstat uses SGE cluster itself.
- <return diagonal>: default: 0 (false). If set to 1, we also return the similarity of each matrix and itself. (Useful for computing the variance for the univariate count distributions.)

Output

Matrix with following columns:

- matrixA: first matrix
- matrixB: second matrix
- Smax: Similarities summarized by using the maximum
- Ssum: Similarity measured by covariance
- imax: Position with the highest similarity (B is shifted against fixed A!)
- bimaxp: maximum similarity is a reverse complementary hit (1) otherwise (0)
- alphaA: probability of a false positive for matrixA
- alphaB: probability of a false positive for matrixB

Examples

Compute similarities between all pairs of matrices from data/matrix.list using a balanced threshold:

```
./sstat .4 list:data/matrix.list balanced .1
```

scluster

Returns a clustering of PFMs (see Chapter 10).

Call

```
$ ./scluster <gc> <[list:]transfac-file> <threshold-method> <threshold-parameter>
. [<use-sge>] [<p=.95>] [<LOOCV>] [<regularize>]
```

where most parameters are the same as for cstat. New parameters:

- <use-sge>: 0/1 (standard: 0) uses sge engine to build similarity matrix
- <p>: Two PFMs are considered for merging only if their Smax value is higher than the maximum of the quantile p of all pairwise Smax values and 0.
- <LOOCV>: Performs a Leave-One-Out-Cross-Validation.

Output

Matrix with following columns:

- matrixA: first matrix
- matrixB: second matrix
- QA: power of matrix A
- QB: power of matrix B
- icA: information content of matrix A
- icB: information content of matrix B
- Smax: Similarities summarized by using the maximum
- imax: Position with the highest similarity (B is shifted against fixed A!)
- bimaxp: maximum similarity is a reverse complementary hit (1) otherwise (0)
- Q: power of new matrix
- ic: information content of new matrix

Furthermore, following files are written (in the same directory where the input files are):

- <transfac-file>.matrices: contains all familial binding profiles (cluster representatives) for each cluster including intermediate representatives.
- <transfac-file>.cluster: contains the final familial binding profiles for each cluster and all remaining singletons.

Example

Computes clustering of all pairs of matrices from data/matrix.list using a balanced threshold.

```
$ ./scluster .4 list:data/matrix.list balanced .1
```

costat for co-occurences

This program returns the probability (or the rates you need to compute this probability) to have at least one hit of TF A and at least one hit of TF B in a window (see Chapter 6). The calculation is performed for all pairs of the given set of TFs.

Call

```
$ ./costat <gc> <[list:]transfac-file> <threshold-method> <threshold-parameter>
. [<window size>] [<bregularize>] [<file with empirical alphas>]
```

where most parameters are the same as for cstat. New parameters:

- <window size>: if the parameter is not given, the program only outputs the rates. In case of a given window size, it returns the probability to have at least one hit for A and one hit of B.
- <file with empirical alphas>: instead of using the theoretically derived alphas (probability for a false positive - at one position!), you can supply the empirical probability (count the number of hits and divide by twice of the sequence length due to the complementary strand). This corrects against a bias occuring for unexpected frequent motifs. The file should contain one probability per line in the same order as the matrices in the transfac file.

Output

Output if window length was not given Matrix with following columns:

- matrixA: first matrix
- matrixB: second matrix
- rA: rate for the occurence of matrix A
- rB: rate for the occurence of matrix B
- rAB: rate for the occurence of matrix A and B
- alphaA: typeI error for matrixA
- alphaB: typeI error for matrixB

Output for given window length Matrix with following columns:

- matrixA: first matrix
- matrixB: second matrix
- p: probability to observe at least one hit of matrixA and one hit of matrixB in a window of given length.

Example

For rare rate output:

```
$ ./costat .4 list:data/matrix.list typeIext .1
```

and for probability output with a window size of 500

```
$ ./costat .4 list:data/matrix.list typeIext .1 500
```

bsanno for clustering

Annotates sequences with binding sites.

Call

```
$ ./bsanno <sequence-file> <[list:]transfac-file> <threshold-method>
. <threshold-parameter> [<bregularize>] [<statistics>]
. [<gc content for global option>]
```

where most parameters are the same as for cstat. New parameters:

- <sequence-file>: a FASTA file containing sequences for annotation, e.g. data/seq.fasta
- <statistics>: default: false; if true then pvalue per sequence per PFM are reported, otherwise position of binding sites are reported.
- <gc-content for global option>: if this parameter is not set, we use for each sequence annotation for the background model (for PSSM generation and threshold determiniation) the gc content given by the selected sequence (default option!). If a gc content (like .3) is given, we define the background model (for PSSM generation and thresholding) by this given gc content (here .3) for all sequences.

Output

Matrix with following columns:

- matrix: matrix for binding site
- gene: name of gene
- strand: 1 for binding site on given sequences, -1 for reverse complementary strand
- pos.start: starting position of the binding site (ignoring its orientation) counted for upstream sequences (this means we assume the TSS at the end of the given sequences, therefore, the last sequence position is 0, the second last 1, and the first position is the sequence length-1), in fact, we enumerate the sequence positions from right to left.
- pos.end: corresponding ending position of binding site, pos.end¡pos.start always holds.
- seq.start: starting position of the binding site while enumerating the positions from right to left. Thus, first position of the sequence corresponds to 0 and the last position is sequence length-1. Again, we ignore the orientation of the binding site.
- seq.end: ending position of the binding site. seq.end > seq.start.

Example

Annotates sequences in seq.fasta by the binding sites contained in matrix.list using a balanced threshold.

```
./bsanno data/seq.fasta list:data/matrix.list balanced .1
```

pfmqual to compute quality of PFMs

Computes the correlation between set of binding sites (given in Transfac file in lines starting with BS <sequence>) and the PFM (see Chapter 11). Make sure that BS is followed either by a tab and the sequence or by two spaces.

Call

```
$ ./pfmqual <gc> <[list:]transfac-file> <threshold-method> <threshold-parameter>
. [<bregularize>] [<bnr>]
```

where all parameters except <threshold-method> are the same as before.

- <threshold-method>: 'optimize' to optimize the threshold such that quality is maximixed. bnr: if 1 then binding sites are transformed to unique sequences, if 0 (default) we use the binding sites as given in the transfac file.

Output

Matrix with following columns:

- matrix: name of matrix
- quality: quality of PFM
- quality.nr: non-redundant (unweighted) quality
- sens: sensitivity
- spec: specificity
- prec: precision
- threshold: the chosen threshold (important for 'optimize')
- alpha: alpha error
- beta: beta error

Example

```
./pfmqual .4 list:data/matrix.list balanced .1
```

Parallel Computing

Some of the programs support parallel computing. Since we support OMP (one memory, multiple processors) (Dagum and Menon, 1998) and the Sun Grid Engine (Sun, 2002) - multiple memory, multiple processors - we divide this section correspondingly.

OMP

If you have an OMP ready compiler, you just have to uncomment the two lines in the Makefile:

```
compileoption += -fopenmp
linkoption = -lgomp -o
```

And, perhaps, change the parameter for using openmp fitting to the compiler you use. We are using C++ compiler v. 4.2.0 for 64bit machines. If you have compiled the program with OMP enabled, the clustering will perform much faster in recomputing the similarities of each new representatives with all other nodes.

The Sun Grid Engine

As the parameters suggested (above), the programs support the Sun Grid Engine - although we have to admit that the implementation is rather proprietary. Anyways, some inspection of the sge.h and sge.cpp should clarify the implementation and give the possibility to extend it. In fact, all classes which can use the SGE engine (CSimilarityMatrix and CClusterMatrix since it is inherited from CSimilarityMatrix but does not need any further adjustment.) are derived from the interface ISGEClient. Two adjustment might be needed:

1. In the client class CSimilarityMatrix you might want to modify the path of sstat. We assume that it is contained in the path - then - you don't need any modifications.
2. In `sge.cpp`, three main task are done - and might need some adjustments:

    a) Initialization: The constructor of CSGEMaster needs a temporary directory (default: sgetemp). Be aware that each construction might delete files within this directory.

    b) Job Submission: Implemented in the method submit. Here, you have to change the format/directives and the program to submit the job (default:submit2sge) as well as the queue and other parameters such that they fit you environment.

    c) Waiting: After the jobs are submitted, the class waits until all jobs are done (method finish()). Here, we use the program qstat to see which jobs are done (using the job id caught at submission) and also perform some basic error handling. Depending on your output of qstat, your error logs and so on, you might want to change some code there, as well. (By the way, if we see that a job was finished successfully, we call the callback function sge_merge (for which we in fact use the interface ISGEClient) to read the output).

License

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, see http://www.gnu.org/licenses/.

Appendix B: Summary (German)

Zusammenfassung

Transkriptionsfaktoren (TF) spielen eine entscheidende Rolle in der Regulation von Genen. Sie interagieren mit spezifischen Bindestellen oder Motifen auf der DNA Sequenz. Daher ist eine wichtige Aufgabe der Bioinformatik, potentielle Bindestellen von TF *in silico* vorherzusagen. Nimmt man einen statistischen Standpunkt ein, dann ist die DNA Sequenz ein langer Text bestehend aus vier verschiedenen Buchstaben 'A', 'C', 'G' und 'T' für die vier verschiedenen Basen. Bindet ein TF an eine Bindestelle, so ist dies gleichbedeutend damit, dass das Wort, welches die Bindestelle beschreibt, in dem Text vorkommt. Daher kann man für verschiedene Statistiken auf schon bekannte zurückgreifen und somit Fragen nach der Wahrscheinlichkeit eine bestimmte Anzahl von Wörtern zu beobachten oder der Distanz zwischen zwei Vorkommen beantworten. Jedoch tritt bei der Herleitung solcher Statistiken immer wieder das gleiche Problem auf: Die Wörter können überlappen. Daher entstehen Abhängigkeiten zwischen den zugrunde liegenden Zufallsvariablen. Dadurch gibts es z.B. bisher noch keine exakte Formel - die nicht auf erzeugenden Funktionen beruht - zum Berechnen der Wahrscheinlichkeit eine bestimmte Anzahl von nicht-überlappenden Wörtern zu sehen. Wir leiten diese Formel her und erhalten dadurch auch eine Normalverteilungs-Approximation.

Leider bindet ein TF aber nicht nur ein an einzelnes Wort, sondern normalerweise gibt es innerhalb des Wortes Position, die Variationen zu lassen. Daher werden TF meist in dem statistischen Modell PFM dargestellt. Dieses Modell weist jedem Buchstaben auf jeder Position ein Gewicht zu. Wenn die Summe aller Gewichte für eine gegebene Sequenz der Länge des Motifs einen Schwellenwert übersteigt, so ist diese Sequenz eine Bindestelle. Daher kann man auch alle derartigen Wörter aufzählen und erhält so eine Menge von Wörtern, die ein Motif beschreibt. Allerding kann diese Menge sehr gross werden. Z.B. für ein Motif der Länge 15 ist die Anzahl normalerweise um die 500.000. Abgesehen davon, dass das Aufzählen der Wörter exponentielle Laufzeit hat, kommen auch die bekannten Statistiken bei einer so grossen Anzahl von Wörtern an ihre Grenzen. Das heisst, sie sind nur sehr aufwändig zu berechnen und die Näherungsergebnisse sind nicht sehr genau.

Daher werden neue Statistiken und effiziente Algorithmen benötigt. Wir haben solche Statistiken entwickelt. Dabei nutzen wir aus, dass wir die Wahrscheinlichkeit für überlappende Bindestellen ausrechnen können ohne die Wörter aufzuzählen. Genauer gesagt, benutzen wir das PFM Modell um eine zwei-dimensionale Gewichtsverteilung zu berechnen. Von dieser können wir besagte Wahrscheinlichkeit ablesen. Von diesem Ergebnis ausgehend, leiten wir die exakte Varianz der Anzahl von Vorkommen her. Ausserdem können wir die Verteilung der Vorkommen durch eine zusammengesetzte Poisson Verteilung beschreiben. Simulationen zeigen, dass dies die beste bekannte Approximation ist. Auch

können wir für nicht überlappende Vorkommen entsprechende Statistiken auf Basis einer Poisson Verteilung berechnen. Erweiterung auf mehrere verschiedene DNA Motife führt zur Berechnung der Signifikanz von gemeinsamen Vorkommen und der Kooperation von TF. Zusätzlich führen wir die Kovarianz als Maß für die Ähnlichkeit von DNA Motifen ein. Dadurch erhalten wir ein natürliches und vor allem generelles Ähnlichkeitsmaß, das nicht von einem speziellen Modell ausgeht. Explizite Formeln leiten wir für das PFM Modell her und Vergleich mit Simulationen und anderen Maßen zeigt, dass unser Maß tatsächlich die von uns definierte Ähnlichkeit am Besten wiedergibt. Ein verwandtes Maß verwenden wir zum Gruppieren von Klassen von TF. Auch hier zeigt ein Vergleich mit optimierten Gruppierungsalgorithmen, dass wir vergleichbar gute Ergebnisse erhalten. Schließlich nutzen wir die Ähnlichkeit, um herauszufinden, wie gut ein DNA Motif mit einem bestimmten Modell dargestellt werden kann. Hierfür berechnen wir die Kovarianz zwischen den experimentell verifizierten Sequenzen und dem Modell. Dies entspricht der Representations-Qualität von DNA Motif Modellen. Wiederum leiten wir für PFMs explizite Formeln her. Darauf basierend zeigen wir, dass die Qualität auch dafür genutzt werden kann, Modellparameter (in unserem Fall der Schwellenwert) zu optimieren. Außerdem zeigen wir, dass die Qualität für Motife, die den Annahmen des PFM Modells nicht entsprechen, auch signifikant niedriger ist.

Curriculum Vitae

Der Lebenslauf ist in der Online-Version
aus Gründen des Datenschutzes nicht enthalten

Ehrenwörtliche Erklärung

Hiermit erkläre ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel und Quellen verwendet habe.

Berlin, August 2008 Utz J. Pape