Dissertation

zur Erlangung des akademischen Grads
doctor rerum naturalium (Dr. rer. nat.)

Probabilistische Modelle zur Beschreibung und Vorhersage regulativer DNA-Sequenzen

Diplom-Informatiker
Rainer Pudimat

25. Oktober 2008

vorgelegt dem Rat
der Fakultät für Angewandte Wissenschaften
der Albert-Ludwigs-Universität Freiburg

Dekan: Prof. Dr. Nebel

Gutachter: Prof. Dr. Backofen
Prof. Dr. Morgenstern

Tag der mündlichen Prüfung: 8. Dezember 2008

Inhaltsverzeichnis

Kapitel 1

Einleitung

In den letzten Jahren wurden die Genome zahlreicher Arten vollständig sequenziert. Die molekularbiologische Forschung steht damit jedoch immer noch am Anfang der Bemühungen, die Funktionsweise eines Lebewesens in seiner ganzen Komplexität zu verstehen. Ein wichtiger Schritt dahin ist die Analyse der Funktionen von Proteinen, jenen Molekülen also, die in ihrer Gesamtheit einen Großteil der Merkmale eines Lebewesens festlegen. Die Funktionsweise eines Proteins wird nicht nur durch eine enzymatische Wirkung definiert, sondern auch durch die Dynamik seines Auftretens in verschiedenen Zelltypen oder -stadien und verschiedenen physiologischen Situationen.

Die Rate, mit der ein bestimmtes Protein produziert wird, bzw. die Aktivität des dazugehörigen Gens unterliegt einem vielschichtigen Regulationsapparat. Die bedeutendste Stufe dieses Apparats ist die transkriptionelle Genregulation. Diese wird durch Transkriptionsfaktoren ausgeübt, speziellen Proteinen, die auf der DNA binden können und Einfluss auf die Aktivität von bestimmten Genen nehmen können. Transkriptionsfaktoren binden bevorzugt eine für sie charakteristische Bindungssequenz (TFBS), einer kurzen Nukleotidfolge von ungefähr zehn bis zwanzig Basenpaaren. Abweichungen von dieser optimalen Sequenz werden je nach Transkriptionsfaktor mehr oder weniger stark toleriert. In komplexeren Lebewesen kooperieren häufig mehrere Transkriptionsfaktoren und wirken als funktionale Einheit auf die Genexpression. Zu wissen, welche Transkriptionsfaktoren welche Gene regulieren, wäre ein Fortschritt bei der Aufklärung der Funktion von Proteinen.

Diese Arbeit beschäftigt sich mit der rechnergestützten Analyse von regulativen DNA-Sequenzen, jenen DNA-Bereichen also, die durch Transkriptionsfaktoren gebunden werden. In drei Teilen werden drei verschiedene Aspekte dieses Forschungsgebiets bearbeitet: 1.) die stochastische Modellierung und Erkennung einzelner TFBS, 2.) die stochastische Modellierung von funktionell zusammenhängenden Gruppen benachbarter TFBS und 3.) das unüberwachte Lernen unbekannter Sequenzmotive.

1.1 Modellierung von Transkriptionsfaktorbindungsstellen

Im ersten Teil der Arbeit wird ein stochastischer Modellierungsansatz für TFBS eines Transkriptionsfaktors entwickelt. Da die auffälligste Gemeinsamkeit dieser TFBS ihre

positionsweise Sequenzähnlichkeit ist, beschränken sich derzeitige Standardverfahren auf dem textuellen Muster der typischen TFBS eines Faktors. Die Fähigkeit dieser Verfahren, präzise neue TFBS in DNA-Sequenzen vorherzusagen, ist begrenzt, da die Kürze von TFBS und die erlaubte Variabilität bedingen, dass bezüglich dieser Modelle optimale Treffer rein zufällig auch in Bereichen auftreten können, in denen eine Regulation durch den untersuchten Transkriptionsfaktor ausgeschlossen werden kann. Die hohen Klassifikationsfehlerraten der Standardmodelle bedingen, dass die Modelle ihrem Zweck, der Vorhersage interessanter Stellen für experimentelle Untersuchungen, häufig nicht gerecht werden.

Der in dieser Arbeit vorgestellte Modellierungsansatz ermöglicht eine flexiblere Beschreibung der charakteristischen Eigenschaften von TFBS eines Transkriptionsfaktors und berücksichtigt zusätzlich statistische Zusammenhänge zwischen diesen Eigenschaften. Ziel ist es, über eine adäquatere Modellierung geringere Klassifikationsfehlerraten zu erreichen als die des Standardmodells.

Individuelle TFBS werden nicht mehr als Wort über dem Alphabet der DNA-Nukleotide aufgefasst, sondern allgemeiner als Merkmalsvektor einer Menge von Merkmalen. Es werden eine Reihe von Merkmalsklassen entwickelt, darunter sequenzabhängige lokale Schwankungen von strukturellen Eigenschaften der DNA, Anfangspositionen von Treffern kurzer Muster innerhalb der TFBS oder Überrepräsentationen von Nukleotiden in einer Umgebung der TFBS. Über Merkmalsauswahlverfahren werden Teilmengen möglicher Merkmale ausgewählt, die besonders gut zwischen TFBS und anderen DNA-Sequenzen diskriminieren können.

Die Merkmale einer solchen Teilmenge werden als diskrete Zufallsvariablen in Bayesschen Netz-Klassifikatoren für dieses Zwei-Klassenproblem modelliert. Bayessche Netze sind im Besonderen geeignet, Merkmale mit verschiedenen Wertebereichen zu integrieren. Außerdem werden in ihnen Abhängigkeiten zwischen Merkmalen berücksichtigt.

1.2 Vorhersage von TFBS-Modulen

Obwohl es mit den Bayesschen Netz-Klassifikatoren für TFBS gelingt, die Klassifikationsfehlerraten zu reduzieren, ist die Anzahl von TFBS-Vorhersagen mit diesen Modellen dennoch höher, als in biologischer Hinsicht zu erwarten ist. Eine weitere Möglichkeit, die Relevanz von TFBS-Vorhersagen zu verbessern, ist die Berücksichtigung von Kooperationen zwischen Transkriptionsfaktoren, die häufig eine Nachbarschaft der korrespondierenden TFBS erfordern. Eine Gruppe funktionell zusammengehörender TFBS heißt TFBS-Modul.

Es werden zwei Modellierungsansätze für solche TFBS-Module entwickelt. Die Modelle beider Ansätze haben gemeinsam, dass sie sich in modularer Weise aus stochastischen Modellen einzelner TFBS zusammensetzen. Sie unterscheiden sich darin, wie der Zusammenhang zwischen korrespondierenden TFBS modelliert wird.

Der erste Ansatz basiert auf bereits bekannten Modellen für TFBS-Module, die einfache TFBS-Sequenzmodelle in ein Hidden Markov Modell (HMM) integrieren. Diese speziellen HMM modellieren den gesamten stochastischen Prozess der Erzeugung von TFBS-Modulen und angrenzenden Sequenzbereichen. Bisher wurden als TFBS-Sequenzmodelle einfache Gewichtsmatrizen eingesetzt. In dieser Arbeit wird der Einsatz der im ersten Teil entworfenen BN-Klassifikatoren als TFBS-Sequenzmodell innerhalb ähnlicher HMM entwickelt. Die Grundidee besteht darin, Bayessche Netze als Ausgabeverteilungen von HMM-Zuständen zu verwenden. Die HMM erzeugen dann keine Nukleotidsequenz, sondern eine Folge von Merkmalsvektoren.

Der zweite Ansatz zur Modellierung von TFBS-Modulen besteht in einer positionsweisen a priori Verteilung über alle berücksichtigten TFBS-Modelle. Diese a priori Verteilungen überlagern die Vorhersagen der TFBS-Modelle und wirken als Filter, der TFBS-Vorhersagen bestraft, wenn sie in einem ungünstigen Kontext auftreten und TFBS-Vorhersagen verstärkt, wenn der Kontext Hinweise auf die Relevanz einer Vorhersage bietet. Ein günstiger Kontext bedeutet, dass notwendige TFBS kooperierender Faktoren in richtiger Entfernung und Orientierung vorhanden sind. Zur Berechnung der a priori Verteilungen werden für jedes TFBS-Modell Bedingungen in Form aussagenlogischer Ausdrücke definiert, die erfüllt sein müssen, damit eine Vorhersage biologisch relevant sein kann. Neben der Nachbarschaft korrespondierender TFBS können die Bedingungen auch Sequenzannotationen enthalten, die beispielsweise Informationen über Gewebespezifität oder Transkriptionsstartstellen des regulierten Gens beschreiben. Die Auswertung solcher aussagenlogischen Ausdrücke wird in speziell konstruierten Bayesschen Netzen probabilistisch umgesetzt. Diese Netze erzeugen für jede Sequenzposition in Abhängigkeit der Wahrheitswerte der logischen Ausdrücke eine a priori Verteilung über allen beteiligten Sequenzmodellen.

1.3 Sequenzmotivsuche unter Verwendung von a priori Informationen

Der dritte Teil der Arbeit widmet sich der Motivsuche, dem Aufspüren ähnlicher Teilsequenzen in einer Menge von Sequenzen. Eine solche Aufgabenstellung ergibt sich beispielsweise, wenn von einer Sequenzmenge bekannt ist, dass jede der Sequenzen einen bestimmten Transkriptionsfaktor bindet, jedoch die genaue Position der Bindungen sowie ein TFBS-Modell unbekannt sind. Ziel ist es gleichermaßen, ein Modell für die TFBS und Treffer dieses Modells in der Eingabe zu erhalten. Diese Lernaufgabe ist unüberwacht, da nicht bekannt ist, welche Teilsequenzen TFBS sind und welche nicht.

Ein bekanntes Verfahren zur Motivsuche, MEME, verwendet das EM-Prinzip, bei dem in iterativer Weise Erwartungswerte für die TFBS-Startpositionen unter Verwendung eines aktuellen TFBS-Modells geschätzt werden und anhand dieser Schätzung das aktuelle TFBS-Modell erneuert wird.

Ein Nachteil von MEME ist, dass alle Startpositionen a priori gleichwahrscheinlich sind. Häufig besteht jedoch zusätzliches Wissen zu den einzelnen Sequenzen, die bestimmte Positionen als wahrscheinlichere Startpositionen erscheinen lassen als andere Positionen. Ein Beispiel sind Informationen über Nukleosombindungsstellen auf der DNA. In diesen Bereichen haben Transkriptionsfaktoren keinen Zugriff auf die DNA, so dass dort eine TFBS unwahrscheinlich ist. Ein weiteres Beispiel sind auch bei der Motivsuche Bindungsstellen für kooperierende Faktoren, die Positionen in deren Nähe wahrscheinlicher machen.

Die in diesem Teil der Arbeit entwickelte Erweiterung von MEME bezieht diese Informationen in Form von a priori Verteilungen über mögliche Startpositionen ein. Dazu werden die in MEME verwendeten probabilistischen Modelle und Schätzalgorithmen angepasst. Da dieses Verfahren in Zusammenarbeit mit meinem damaligen Kollegen Michael Hiller entstanden ist, bezieht sich ein Teil der Evaluierung des Verfahrens auf ein verwandtes Themengebiet, der Suche nach Bindungsstellen für Proteine zur Regulierung von alternativem Splicing auf RNA-Sequenzen. In diesem Anwendungsfall sind Positionen vielversprechende Bindungsstellen, an denen die RNA mit großer Wahrscheinlichkeit einzelsträngig vorliegt.

1.4 Aufbau der Arbeit

Im nächsten Kapitel findet der Leser eine detaillierte Einführung in den molekularbiologischen Kontext dieser Arbeit. Ein Hauptaugenmerk ist darauf gelegt, zu verdeutlichen, dass es sich bei der DNA und den bindenden Proteinen um komplexe Moleküle handelt, die vielfältigen Einflüssen unterliegen. Es soll darauf abzielen, zu verstehen, warum die Modellierung von TFBS nur auf Grundlage der Nukleotidsequenz eine zu starke Vereinfachung darstellt. Im letzten Teil des Kapitels werden einige Labortechniken vorgestellt, mit denen ein Großteil der Sequenzdaten generiert wurden, die in den Lern- und Evaluierungsverfahren der folgenden Kapitel verwendet werden. In Kapitel 3 wird der aktuelle Forschungsstand in der bioinformatischen Analyse regulativer Sequenzen vorgestellt, insbesondere in den drei Bereichen, mit denen sich diese Arbeit beschäftigt. Kapitel 4 bietet eine Einführung in die Modellierung und dem Lernen von Bayesschen Netzen. Zusätzlich werden Techniken diskutiert, die häufig mit dem Lernen von Bayesschen Netzen einhergehen, wie der Merkmalsauswahl und dem Diskretisieren kontinuierlicher Merkmale. In Kapitel 5 wird das TFBS-Modell basierend auf BN-Klassifikatoren entworfen und die Ergebnisse von Leistungsanalysen vorgestellt. Innerhalb dieses Kapitels wird auch das HMM-basierte Verfahren zur Modellierung von TFBS-Modulen eingeführt. Kapitel 6 stellt den zweiten Ansatz zur Modellierung und Erkennung von TFBS-Modulen vor. Kapitel 7 beschäftigt sich mit dem dritten Teil dieser Dissertation, der Motivsuche. In letzten Kapitel der Arbeit werden die Erkenntnisse aller drei Teile zusammengefasst.

Kapitel 2

Molekularbiologische Grundlagen der trankriptionellen Genregulation

Dieses Kapitel behandelt ausführlich die Grundlagen der transkriptionellen Genregulation. Einem Informatiker soll das notwendige Hintergrundwissen gegeben werden, das zum Verstehen der Abhandlungen zur stochastischen Modellierung regulativer DNA-Sequenzen erforderlich ist. Ein detailiertes Betrachten der bei der Genregulation ablaufenden Vorgänge und der beteiligten Moleküle offenbart die Chancen einer rechnergestützten Analyse und die Grenzen einer Modellierung bei unvollständigem Wissen.

Das Kapitel macht deutlich, dass es sich bei der DNA nicht lediglich um eine Zeichenkette mit vier Buchstaben handelt, die mit den Methoden der Informatik bequem zu analysieren wäre. Die DNA wird als komplexe Verbindung beschrieben, die ständigen Veränderungen unterliegt. In dieser Dissertation werden Arbeiten vorgstellt, die darauf zielen, unter teilweiser Berücksichtigung dieser Komplexität und dem Zusammenspiel der regulierenden Komponenten die rechnergestützte Analyse regulativer DNA-Sequenzen zu verbessern.

Abschnitt 2.1 gibt zunächst einen kompakten Überblick über alle molekularbiologischen Schritte, die zur Herstellung von Proteinen gemäß des genetischen Bauplans nötig sind. An den jeweils passenden Stellen werden dazu die beteiligten Stoffe betrachtet. Abschnitt 2.2 betrifft die Transkription genetischer Informationen, insbesondere deren Regulation. Abschnitt 2.3 beschreibt experimentelle Verfahren, deren Ergebnisse die Datenbasis für Entwicklung bioinformatischer Methoden auf dem Gebiet der transkriptionellen Genregulation sein können.

2.1 Genexpression — der Weg vom Gen zum Protein

Allen Lebewesen ist gemein, dass der überwiegende Anteil ihrer Funktionen und ihrer Merkmale durch Proteine und RNA-Molekülen erfüllt und bestimmt wird. Der modulare Aufbau von Proteinen und RNA-Ketten aus Einzelbausteinen ist bei allen Lebewesen identisch, genau wie die Einzelbausteine selbst. Weiterhin gleichen sich alle Lebewesen in dem Punkt, dass sie die Konstruktionspläne für Proteine und RNA-Ketten in ihrem aus DNA aufgebauten Erbgut speichern. Bei mehrzelligen Lebewesen liegt das vollständige Erbgut in jeder einzelnen Zelle vor.

Der Prozess der Herstellung eines Proteins gemäß eines gespeicherten Bauplan, dem *Gen*, wird als *Genexpression* bezeichnet. In diesem Abschnitt werden die wichtigsten Teilschritte und die bei der Genexpression beteiligten Stoffe beschrieben. Unterabschnitt 2.1.1 beschäftigt sich zunächst mit der DNA, der Trägerin der Erbinformation. Außerdem wird dort auf die Organisation der gespeicherten Information eingegangen. Unterabschnitt 2.1.3 beschreibt den ersten Schritt zur Herstellung eines Proteins, die *Transkription*. Die Regulationskonzepte der Transkription, mit denen sich diese Arbeit hauptsächlich beschäftigt, werden dort ausgeklammert, da sich Abschnitt 2.2 im Besonderen diesen Sachverhalten widmet. Der letzte Schritt der Genexpression, die *Translation* wird in Unterabschnitt 2.1.4 beschrieben. Da der Translation in dieser Arbeit eine sehr untergeordnete Bedeutung zukommt, beschränken sich die Darstellungen auf eine entsprechend niedrige Detailstufe.

Das Standardwerk *The Molecular Biology of the Cell* [Alb02] enthält den allgemeinen Wissenstand zur Molekularbiologie und ist eine generelle Referenz für diesen Absatz.

2.1.1 DNA — Träger der Erbinformation

Desoxyribonukleinsäure (DNA — englisch für *deoxyribonucleic acid*) ist ein Biopolymer, dass aus zwei gegenseitig angelagerten Polymerketten besteht, die jeweils durch unverzweigte Verkettung elementarer Untereinheiten, den *Nukleotiden*, gebildet werden. Eine einzelne Kette wird als *DNA-Strang* oder *Einzelstrang* bezeichnet. Zusammen ergeben sie einen *Doppelstrang*.

Elementare Bausteine. Ein Nukleotid besteht aus jeweils einer *Phosphorsäure*, einer *Desoxyribose*[1] und einer organischen Base (siehe Abbildung 2.1a.). In DNA-Molekülen kommen vier verschiedene Nukleotide vor, die sich lediglich in ihren Basen unterscheiden, welche deshalb häufig stellvertretend für das gesamte Nukleotid genannt werden. Die vier möglichen Basen sind *Adenin*, *Guanin*, *Cytosin* und *Thymin*. Sie werden üblicherweise durch ihre Anfangsbuchstaben `A`, `C`, `G` und `T` abgekürzt. Entsprechend ihrer molekularen Struktur lassen sich zwei Basenklassen unterscheiden: die *Purinbasen* `A` und `G` sowie die *Pyrimidinbasen* `C` und `T` (siehe Abbildung 2.1b-e.).

Polymerisation. Die Verkettung der Nukleotide zu Einzelsträngen geschieht durch Reaktion einer OH-Gruppe der Desoxyribose eines Nukleotids mit der Phosphorsäure eines benachbarten Nukleotids (siehe Abbildung 2.2). Dabei reagiert Phosphorsäure unter Wasserabspaltung zu *Phosphat*. Eine solche Phosphat-Zucker-Kette wird auch *Rückgrat* einer DNA genannt. Die Art der Verkettung bedingt, dass das eine Ende eines DNA-Strangs eine freie OH-Gruppe ausweist. Dieses Ende wird als $3'$*-Ende* bezeichnet, da die OH-Gruppe am dritten Kohlenstoffatom der Desoxyribose sitzt. Aus dem gleichen

[1] Desoxyribose ist ein Zuckermolekül mit 5 Kohlenstoffatomen

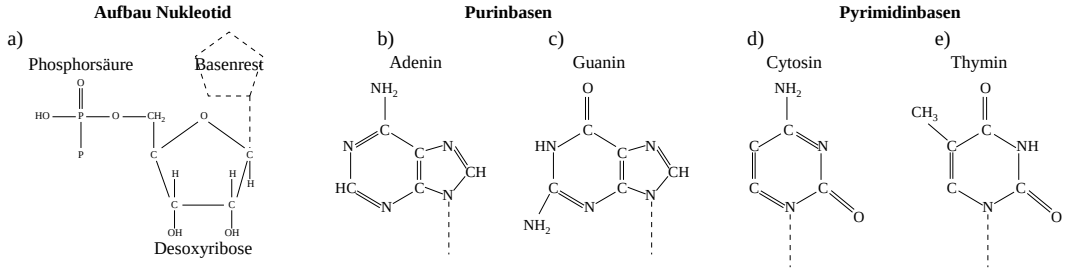


Abbildung 2.1: Aufbau der Nukleotide und die vier Basenreste der DNA.

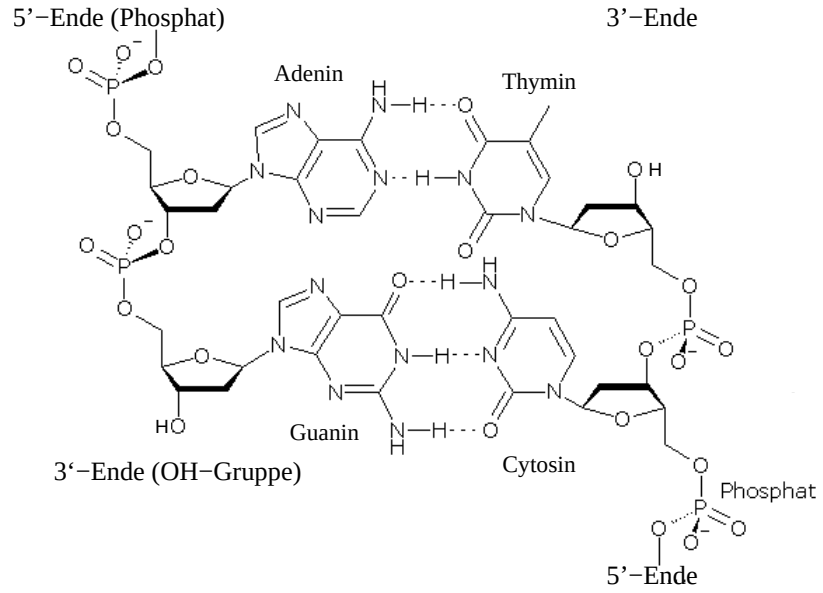


Abbildung 2.2: Polymerisation der Nukleotide und Doppelstrangbildung.

Grund heißt das andere Ende eines DNA-Strangs 5′-*Ende*. Dort befindet sich eine freie Phosphorsäure.

Doppelstrangbildung und Basenpaare. Die Anlagerung der beiden Stränge entsteht durch die Bildung von Wasserstoffbrücken zwischen zwei gegenüberliegenden Basen. Die beiden Stränge verlaufen in Bezug auf ihre verschiedenen Enden gegenläufig. Die Anlagerung ist völlig regelmäßig. Jede Base eines Nukleotids des einen Strangs paart sich mit genau einer Base eines Nukleotids des anderes Strangs, wobei die Partner benachbarter Nukleotide ebenfalls benachbart sind[2]. Gegenüberstehende Basen zweier Stränge bilden ein *Basenpaar*. Es gibt nur zwei mögliche Kombinationen für Basenpaarungen, und zwar zwischen den aus diesem Grund als *komplementär* bezeichneten Basen `A` und `T` sowie `C` und `G`. In beiden Fällen wird eine etwas größere Purinbase (`A` bzw. `G`) mit einer

[2] d.h. es gibt keine Überkreuzungen

kompakteren Pyrimidinbase (`C` bzw. `T`) gepaart. Die Paarung besteht im Falle eines `A-T`-Paares in der Bildung von zwei, im Falle eines `C-G`-Paares von drei Wasserstoffbrücken (siehe Abbildung 2.2). Aus dem Prinzip der Basenpaarung folgt, dass die Abfolge der Nukleotide des einen Strangs eineindeutig die Abfolge der Nukleotide des anderen Strangs bestimmt. Diese Eigenschaft ist von größter Bedeutung für die Herstellung von Kopien der DNA während der Zellteilung oder der Herstellung von RNA-Kopien im Zuge der Genexpression.

Unterscheidung der DNA-Stränge. Ohne dem Ablauf der Transkription vorzugreifen (siehe dazu Abschnitt 2.1.3), sei an dieser Stelle bereits erwähnt, dass für die Herstellung der Kopie das Prinzip der komplementären Basenpaarung ausgenutzt wird. Die Frage, welchem der beiden DNA-Stränge die produzierte RNA entspricht, und in welcher Richtung diese zu lesen ist bzw. verarbeitet wird, stiftet häufig Verwirrung. Um Abhilfe zu schaffen, müssen die beiden DNA-Stränge unterschieden werden. Der Strang, welcher der produzierten mRNA in seiner Basensequenz entspricht, heißt *Sinnstrang*. Die Transkription verläuft bezüglich dieses Strangs in Richtung von dessen $5'$-Ende zu dessen $3'$-Ende. Ein weiterer Name für diesen Strang ist deshalb $5' - 3'$ Strang. Die Anlagerung der RNA-Nukleotide geschieht am anderen DNA-Strang. Bezogen auf diesen Strang ist die Leserichtung $3' - 5'$. Aus Sicht der angelagerten RNA verhält es sich allerdings wie für den Sinnstrang. Der Sinnstrang enthält also nicht nur die gleiche Sequenz wie die RNA, sondern zusätzlich die gleiche Orientierung. Eine in DNA-Sequenzdatenbanken angegebene Sequenz entspricht stets der des Sinnstrangs in $5' - 3'$-Leserichtung. Bezüglich dieser Orientierung spricht man auch vom $5'$-Ende und $3'$-Ende eines Gens.

Im weiteren Verlauf dieser Arbeit wird es häufig nötig sein, die relative Lage zweier Sequenzabschnitte zueinander anzugeben. In der englischen Fachliteratur haben sich dafür die beiden Adjektive *upstream* und *downstream* etabliert. Ein Sequenzstück befindet sich *upstream* eines Anderen, wenn es auf dem Sinnstrang näher am $5'$-Ende liegt als das Andere. Es liegt *downstream*, falls es näher am $3'$-Ende liegt. Im Folgenden werden die beiden Begriffe mit *" oberhalb von etwas "* für *upstream* und *" unterhalb von etwas"* für *downstream* verwendet.

Räumliche Struktur der DNA. Der beschriebe Aufbau der DNA ähnelt dem einer Strickleiter, mit den Basenpaaren als Sprossen und den beiden Phosphat-Zucker-Ketten als Holme. Die chemischen bzw. energetischen Eigenschaften eines DNA-Moleküls bedingen, dass die beiden Rückgrate durch gegenseitige Verwindung eine Doppelhelixstruktur bilden, die *DNA-Doppelhelix*. Eine Windung dieser Doppelhelix enthält ungefähr zehn Basenpaare. Die gewundene DNA besitzt zwei unterschiedlich große Furchen, da die beiden Zuckergruppen eines Basenpaars einen Winkel von $120°$ statt $180°$ zueinander haben. Die Basenpaare tendieren in Richtung der *großen Furche* (englisch *major groove*) und hin zur Windungsachse. Die andere Furche heißt entsprechend dazu *kleine Furche* (englisch *minor groove*). Aufgrund des besseren Zugangs zu den Basenpaaren bindet die Mehrheit DNA-bindender Proteine, welche eine bestimmte Nukleotidfolge erkennen, im

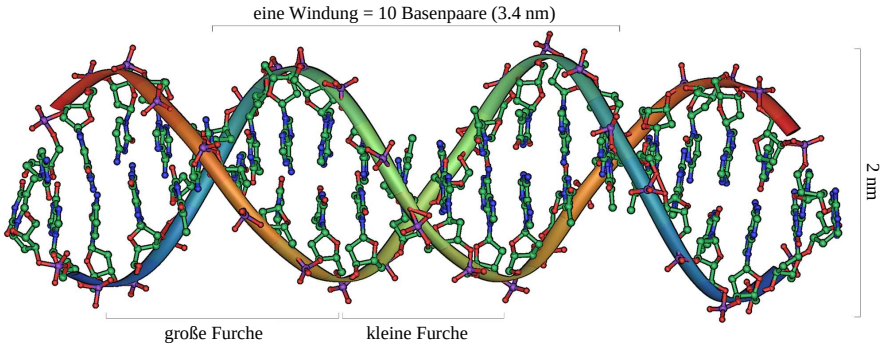


Abbildung 2.3: DNA-Doppelhelix.

Bereich der großen Furche. Eine schematische Darstellung einer DNA-Doppelhelix ist in Abbildung 2.3 dargestellt.

Sequenzabhängige Schwankungen der strukturellen DNA-Eigenschaften. Die beschriebene räumliche Struktur einer DNA unterliegt vielfältigen Schwankungen auf verschiedenen Ebenen [Dic92]. In Bezug auf die Ausprägung der Helixstruktur lassen sich drei Hauptklassen unterscheiden. Die am häufigsten vorkommende Helixform ist die *B-Form*. Sie weißt die im vorherigen Absatz genannten Eigenschaften auf. Die *A-Form*, welche überwiegend bei RNA-Molekülen und DNA-RNA-Komplexen zu finden ist, weißt eine gedrungenere Windung mit größerem Radius auf. Die große Furche ist schmaler zugunsten einer vergrößerten kleinen Furche. Die dritte vorkommende Helix heißt *Z-Form*. Sie ist im Gegensatz zu A- und B- Form links-gedreht und schmaler als die Erstgenannten.

Neben diesen globalen Konfirmationen einer DNA ergeben sich auch auf Ebene einzelner Basenpaare Abweichungen in der Geometrie der Helix, die erstmals Anfang der Achtziger Jahre durch hochauflösende Röntgen-Kristallstrukturanalyse-Verfahren gemessen werden konnten. Zuvor galt die DNA als reguläre aufgebaute Doppelhelix, deren Geometrie in strikter Weise durch die physiko-chemischen Eigenschaften des Zucker-Phosphat-Rückgrats vorgegeben war. Dickerson et al. [Dre81] waren die ersten, die detaillierte Untersuchungen der lokalen Struktur auf nahezu atomgenauer Auflösung durchführten. Ihrem Untersuchungsobjekt, dem Oligomer `CGCGAATTCGCG`, entnahmen sie wichtige Erkenntnisse.

So sind die Basen eines Basenpaars nicht koplanar, wie im ursprünglichen Modell von Watson und Crick [Wat53] angenommen, sondern entlang ihrer Verbindungsachse gegeneinander rotiert (*propeller twist*). Da Purinbasen (`A` und `G`) etwas größer sind als Pyrimidinbasen, ergeben sich Platzprobleme im Falle aufeinander folgender Purinbasen. Diese können auf verschiedene Weise ausgeglichen werden. Erstens werden aufeinander folgende Basenpaare entlang der helikalen Achse nicht so stark gegenseitig verdreht (kleinerer *helical twist*). Zweitens können aufeinander folgende, aneinander geratene Basenpaare

gegenseitig seitlich in Richtung eines DNA-Rückgrats verschoben werden (*helical slide*). Drittens kann sich die DNA an diesen Stellen in Richtung der großen bzw. kleinen Furche biegen, um diese Kollisionen zu vermeiden. In der Tat zeigten sich bei Dickerson et al. Schwankungen in der relativen Anordnung aufeinander folgender Basenpaare, die abhängig von den Basen des Basenpaars selbst sind.

Viele weitere Kristallstrukturanalysen wurden in der Folgezeit untersucht, so dass die Strukturschwankungen für bestimmte Dinukleotide statistisch untersucht werden konnten. Zur Beschreibung der möglichen Ausgleichsbewegungen der Basenpaare zueinander wurden weitere geometrische Parameter eingeführt. Einige sind in Abbildung 2.4 dargestellt. El Hasan et al. untersuchten die Verteilungen dreier Parameter für die verschiedenen Dinukleotide der verfügbaren Datensammlung: den *helical twist*, den *helical roll* und den *helical slide* [EH97]. Danach gibt es unter den Dinukleotiden solche, die für alle drei Parameter relativ konstante Werte aufweisen (die *rigiden* Dinukleotide) und solche mit einer hohen Varianz hinsichtlich der Parameterwerte (die *flexiblen* Dinukleotide). Zur ersten Gruppe gehören `AA`, `AT` und `GA`. Flexible Dinukleotide sind dagegen `GG`, `CG`, `GC`, `CA`, `TA` und `AC`. Für eine Untergruppe der flexiblen Dinukleotide ergibt sich die erhöhte Varianz dadurch, dass es zwei bevorzugte Häufungen im Werthistogramm gibt (`CC`, `GC` und `CG`)[3]. Des Weiteren ist in vielen Fällen ein Großteil der Varianz dann erklärbar, wenn man zusätzlich die flankierenden Nukleotide betrachtet, also die Analyse auf Tetranukleotide ausweitet.

Die beschriebenen lokalen Abweichungen führen auch zu lokalen Änderungen weiterer struktureller, aber auch physikalischer Eigenschaften der DNA. Beispiele für strukturelle Eigenschaften sind die *DNA-Biegsamkeit* [Bru95] oder die Breite der großen bzw. kleinen Furche [Kar96]. Beispiele für physikalische DNA-Eigenschaften sind die Schmelztemperatur [Hog87] oder die lokale Änderung der freien Energie [Sug96]. In der Datenbank B-DNA-VIDEO von Ponomarenko et al. sind 38 sequenzabhängige Strukturparameter, definiert auf Dinukleotiden, verfügbar [Pon99].

Das lokale strukturelle Profil der DNA hat einen großen Einfluss auf die Erkennung von Bindungsstellen durch Transkriptionsfaktoren. Im Rahmen dieser Arbeit werden strukturelle Parameter in Kapitel 5 als potenzielle Merkmale für die Modellierung und Erkennung dieser Bindungsstellen verwendet (siehe Abschnitt 5.1 auf Seite 100).

Chromosomen. Grundlegende Unterschiede in der globalen Struktur und der Aufteilung der DNA zeigen sich zwischen *Prokaryonten* (einzelligen Lebewesen ohne Zellkern) und *Eukaryonten* (Lebewesen mit Zellkern und Cytoskelett). [4] Die gesamte DNA eines prokaryontischen Lebewesens besteht aus einem einzigen ringförmigen Molekül. Die Polarität gemäß der 5′- und 3′-Enden ist demnach aufgehoben.

[3] Aus Symmetriegründen verhalten sind die Werte von`TT` stets mit denen von `AA` identisch, die Werte von `TC` entsprechen denen von `GA`, die von `CC` denen von `GG`, die von `TG` denen von `CA` und die Werte von `TG` denen von `AC`. Deshalb sind hier nicht alle 16 möglichen Dinukleotide betrachtet worden .

[4] Zu den Eukaryonten zählen Pflanzen, Tiere und Pilze. Bakterien und Archaeen bilden dagegen die Klasse der Prokaryonten.

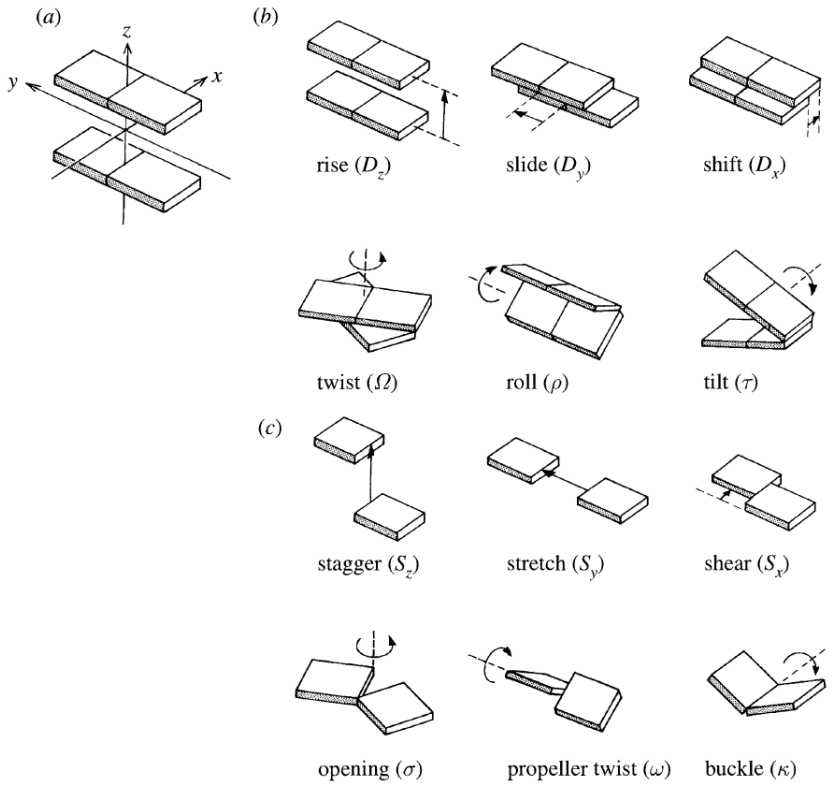


Abbildung 2.4: Sequenzabhängige Strukturparameter (aus [EH97]). a.) Die beiden Scheiben stellen zwei aufeinander folgende Basenpaare dar. Die kleine Furche ist schattiert. b.) Schematische Darstellung von sechs geometrischen Parametern für Dinukleotide c.) Schematische Darstellung von sechs Arten der Verschiebung von Nukleotiden eines Basenpaars zueinander.

Die DNA von Eukaryonten besteht aus mehreren Molekülen, den *Chromosomen*. Diese befinden sich im Zellkern einer jeden Zelle. Der Doppelstrang eines Chromosoms ist mehrfach in stets gleicher Weise um eine Gruppe von Kernproteinen, den *Histonen*, gewunden. Die auf diese Weise gebildeten DNA-Protein-Partikel bilden ein *Nukleosom*. Ein Nukleosom enthält einen DNA-Abschnitt von ungefähr 160 Basenpaaren. Durch weitere Packstufen der Nukleosome untereinander entsteht die äußerst kompakte Struktur eines Chromosoms. Die Positionierung eines Nukleosoms unterliegt verschiedenen Regulationsmechanismen, um Zugriff auf die im Nukleosom enthaltenen DNA-Abschnitte zu gewähren. Diese Remodellierungsvorgänge haben auch einen Einfluss auf die Genexpression, denn DNA, die in Nukleosomen gebunden ist, kann von den genregulierenden Transkriptionsfaktoren nicht gebunden werden.

2.1.2 Organisation der Erbinformation

Allen Lebewesen gemein ist, dass die gesamte genetische Information, d.h. der Bauplan für alle biologischen Merkmalsausprägungen, in der *Sequenz*, d.h. der Abfolge der vier verschiedenen Nukleotide im Erbgut kodiert ist. Zentraler Begriff ist hierbei das *Gen*. Die Gesamtheit der DNA einer Zelle eines Lebewesens wird als *Genom* bezeichnet. Ein Gen ist der Teil einer DNA-Sequenz, der die genetische Information für die Herstellung eines bestimmten *Proteins* enthält.

Proteine sind, wie die DNA, modular aufgebaute Moleküle, welche aus einer Kette von 20 verschiedenen Aminosäuren bestehen[5]. Die Abfolge dieser Aminosäuren bestimmt über die dadurch festgelegte räumliche Faltung der Aminosäuresequenz in wesentlichen Teilen die Funktion eines Proteins im Organismus. Über seine Funktion prägt ein Protein ein Merkmal des Lebewesens.

Man kann die DNA-Sequenz eines Gens in zwei Komponenten unterteilen. Die erste Komponente kodiert die Abfolge der Aminosäuren im Protein. Dabei definieren drei aufeinander folgende Nukleotide, ein sogenanntes *Codon*, eine Aminosäure. Die Abbildung aus der Menge der Codons in die Menge der Aminosäuren heißt *genetischer Code* und ist in allen Lebewesen gleich. Neben Codons für die 20 verschiedenen Aminosäuren enthält dieser Code Signale für den Anfang (*Startcodon*) und das Ende (*Stopcodon*) einer Aminosäurenkette. Die Vorgänge, die zur Herstellung eines Proteins gemäß der kodierenden DNA eines Gens führen, werden in den folgenden beiden Abschnitten betrachtet.

Die zweite Komponente eines Gens umfasst alle DNA-Abschnitte, die nicht für die Kodierung von Aminosäuren verwendet werden. Sie sind hauptsächlich an der Regulation der Expression des betreffenden Gens beteiligt. Besondere Bedeutung haben unter ihnen jene Sequenzteile, die vom Transkriptionsapparat und seinen regulierenden Transkriptionsfaktoren gebunden werden. Diese heißen gemäß ihrer Funktion *regulative Sequenzen*. Die bedeutendste regulative Sequenz ist der *Promotor* eines Gens, der dem Gen vorgelagert ist. Den regulativen Sequenzen ist Unterabschnitt 2.2.2 gewidmet.

Zwischen Promotor und dem kodierenden Bereich befindet sich die so genannte $5'$-*UTR* (für englisch: *untranslated region*). Es handelt sich dabei im Gegensatz zum Promotor um ein Sequenzstück, dass zwar gemeinsam mit der kodierenden Sequenz in eine mRNA transkribiert wird, jedoch anschließend nicht zur Übersetzung in eine Aminosäurenkette verwendet wird. Analog dazu schließt sich der kodierenden Sequenz die $3'$-*UTR* an. Die $5'$-*UTR* ist häufig einige hundert Basenpaare lang und reicht vom Startpunkt der Transkription bis zum Translationsstartcodon. Der $3'$-*UTR* oft sogar einige tausend Basenpaare. Ihm schließt sich häufig ein als Polyadenylierungsstartpunkt bezeichneter Sequenzbereich an, der fast ausschließlich aus `A`-Nukleotiden besteht. All diese Sequenzbereiche dienen als Bindungsbereich von Proteinen, die an der Stabilisierung und Regulierung der Translation beteiligt sind.

[5] Neben den 20 üblichen Aminosäuren existieren weitere, nichtkanonische Aminosäuren, wie *Selenocystein*. Der Einbau dieser Bausteine unterliegt komplexeren Kodierungsregeln.

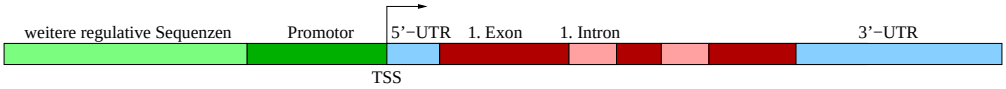


Abbildung 2.5: Aufbau eines eukaryontischen Gens. Die Längenverhältnisse variieren sehr stark.

Ein Chromosom eines eukaryontischen Genoms enthält eine lineare Anordnung von Genen. Während im prokaryontischen Genom der kodierende Teil eines Gens zusammenhängend vorliegt, ist er in eukaryontischen Genen meist mehrfach durch nichtkodierende Bereiche, so genannten *Introns* unterbrochen. Die kodierende DNA ergibt sich in diesen Fällen aus der Verkettung aufeinander folgender kodierender Abschnitte, den *Exons*. Introns enthalten häufig neben den Promotoren Bindungsstellen für regulative Proteine. Des Weiteren dienen sie vermutlich als Experimentierbereiche der Evolution, in denen die Mutationsrate aufgrund fehlenden selektiven Drucks erhöht ist. Sie spielen außerdem eine Rolle beim Alternativen Splicing (vergleiche Unterabschnitt 2.1.3 auf Seite 15), wobei erst der Verarbeitungsprozess der produzierten RNA entscheidet, welche Abschnitte als Exons behandelt werden. Im Laufe der Evolution können Introns zu kodierenden Sequenzen werden oder Bindungsstellen für Transkriptionsfaktoren ausbilden.

2.1.3 Transkription — Produktion von RNA

Im Zuge der Herstellung eines Proteins gemäß einer genomischen Bauvorschrift wird zunächst eine Kopie der betreffenden kodierenden DNA-Sequenz in Form einer einzelsträngigen RNA produziert. Dieser Vorgang heißt *Transkription*.

RNA. Das Produkt der Transkription, die *RNA* (englisch: *ribonucleic acid*), ist, wie die DNA, eine Nukleinsäure. Ihr Name verrät einen ersten Unterschied zur DNA: anstelle des Zuckers Desoxyribose enthalten RNA-Nukleotide den ebenfalls 5-ständigen Zucker *Ribose*. Die vier verschiedenen RNA-Nukleotide enthalten die gleichen Basen wie DNA-Nukleotide, mit Ausnahme von *Uracil* (`U`), dass anstelle von Thymin (`T`) auftritt. Uracil paart sich demnach wie Thymin mit Adenin (`A`). RNA-Moleküle liegen meist einzelsträngig vor, wobei sich komplementäre Basen[6] des Einzelstrangs in üblicher Weise paaren können und so die Bildung einer räumlichen Struktur der RNA veranlassen, welche abhängig von der Nukleotid-Abfolge (Sequenz) ist.

In den letzten Jahren wurde bekannt, dass RNA-Moleküle weitaus mehr Aufgaben übernehmen als bisher angenommen. Neben verschiedenen Funktionen in Signalwegen von Zellen besitzen RNA-Moleküle eine enzymatische Funktion. Die Erforschung der zwei-

[6] Neben den bei der DNA zulässigen Basenpaarungen, auch als Watson-Crick-Paarungen bezeichnet, gibt es bei RNA-Molekülen auch Paarungen zwischen `G` und `U`.

bzw. dreidimensionalen RNA-Struktur rückte vor diesem Hintergrund in den Fokus der Wissenschaft.

Die zum Zweck der Genexpression produzierte RNA-Kopie eines Gens heißt aufgrund ihrer Funktion als Nachrichtenüberträger *messenger RNA* (kurz: mRNA). Neben dieser Funktion treten RNA-Ketten unter anderem als Hilfsmoleküle während der Translation (*tRNA*) und als Konstruktionsmaterial von Ribosomen (*rRNA*) auf[7].

Die Produktion von RNA wird von *RNA-Polymerasen*, Proteinkomplexen mit mehreren Untereinheiten geleistet. Die RNA-Polymerasen für mRNA und tRNA sowie weiteren RNA-Sorten arbeiten ähnlich. Trotz ihrer evolutionären Verwandtschaft unterscheiden sie sich dennoch in einigen Eigenschaften. Insbesondere die Möglichkeit der Einflussnahme weiterer Faktoren, den Transkriptionsfaktoren, ist im Falle der RNA-Polymerase für mRNA, *RNA-Polymerase II*, stärker ausgeprägt. Da sich diese Arbeit mit der Erforschung der Regulation von eukaryontischen Genen beschäftigt, wird im Folgenden vor allem die Transkription von Genen mit Hilfe von RNA-Polymerase II beschrieben.

Ablauf der Transkription. RNA-Polymerase II benötigt zum Einleiten des Transkriptionsprozesses eine Reihe von zusätzlichen Faktoren, den *allgemeinen Transkriptionsfaktoren* (kurz: *GTF* von englisch: *general transcription factor*). Sie heißen in der Reihenfolge ihrer Entdeckung *TFIIA*, *TFIIB* usw. Im Gegensatz zu den Transkriptionsfaktoren, mit denen sich diese Arbeit vorrangig beschäftigt, haben die GTFs keine genspezifische, regulierende Funktion. Sie sind für jeden Transkriptionsprozess unabdingbar. Gemeinsam mit der RNA-Polymerase II bilden sie den *RNA-Polymerase-Komplex*. Dieser wird schrittweise aufgebaut.

Der Promotor vieler Gene enthält eine kurze `T`- und `A`- reiche Sequenz, die so genannte *TATA-Box*. Diese befindet sich in der Nähe des Startpunktes der Transkription, also üblicherweise in der Nähe des Starts der kodierenden Sequenz. Die TATA-Box wird von dem allgemeinen Transkriptionsfaktor *TFIID* bzw. einer seiner Untereinheiten, dem *TATA binding protein* (TBP) erkannt und gebunden. Die Bindung von TFIID erzeugt eine starke lokale Krümmung der DNA im Bereich der TATA-Box [Hor92]. Diese dient zum einen als Kennzeichen für weitere GTFs und ermöglicht zum anderen den Kontakt zwischen weiter entfernt gebundenen, regulierenden Transkriptionsfaktoren mit dem RNA-Polymerase-Komplex. Neben weiteren GTFs lagert sich nun auch die RNA-Polymerase II an. Eine Untereinheit des Faktors *TFIIH* trennt im Folgenden den DNA-Doppelstrang lokal auf. Eine weitere Untereinheit verursacht eine strukturelle Veränderung in dem RNA-Polymerase II-Molekül, welches sich in Folge dessen vom Transkriptionsinitiierungskomplex lösen kann. Die veränderte RNA-Polymerase II arbeitet sich nun entlang der geöffneten DNA und verkettet die am DNA-Strang angelagerten, freien RNA-Nukleotide zu einer mRNA. Währenddessen lösen sich die meisten GTFs von der DNA, um für nachfolgende Transkriptionsprozesse zur Verfügung zu stehen.

[7] Ribosome sind Protein-RNA-Komplexe, die im Cytoplasma einer Zelle vorkommen und in denen die Translation einer Nukleotidsequenz in eine Aminosäuresequenz durchgeführt wird.

Weiterverarbeitung der mRNA durch Splicing. Wie auf Seite 13 bereits erläutert wurde, ist der kodierende Teil eines Gens meist durch Intronsequenzen unterbrochen. Die im Transkriptionsprozess erzeugte mRNA enthält zunächst diese Introns. Da diese Form der RNA nicht zur Translation verwendet wird, heißt sie auch *pre-mRNA*. In einem Nachverarbeitungsprozess, dem so genannten *Splicing* (englisch für spleißen), werden die Introns aus der pre-mRNA heraus getrennt.

Alternatives Splicing. In höher entwickelten eukaryontischen Lebewesen ist die Einteilung eines Gens in Introns und Exons häufig nicht eindeutig. In Abhängigkeit regulierender Mechanismen, die Gegenstand intensiver Forschung sind, können während des Splicings einer pre-mRNA auch Exons herausgeschnitten werden oder bestimmte Introns können in der finalen mRNA verbleiben. Auf diese Weise können durch ein Gen verschiedene Proteine kodiert werden. Die Möglichkeit der Auswahl von kodierenden Sequenzen während des Splicings heißt *alternatives* Splicing. Es ergeben sich dadurch eine Reihe von Vorteilen, die zur höheren Entwicklung der betreffenden Lebewesen beitragen. Zum einen wird durch alternatives Splicing die Informationsdichte der DNA erheblich erhöht [Bre02]. Zum anderen erleichtert alternatives Splicing die Entwicklung von neuen Proteinen, in dem eine leicht veränderte Regulation dieses Vorganges neue Kombinationen von Exonen hervorbringen kann.

2.1.4 Translation

Die letzte Stufe der Genexpression besteht in der Übersetzung der mRNA-Sequenz in eine Aminosäurensequenz mittels des genetischen Codes. Dieser Vorgang heißt *Translation* und findet in speziellen Strukturen, den *Ribosomen statt*, die aus Proteinen und RNA aufgebaut sind und im Cytoplasma von Zellen vorkommen.

Weitere wichtige Bausteine der Translation sind die schon auf Seite 14 erwähnten tRNA-Moleküle. Diese kurzen RNA-Ketten, bestehend aus ungefähr 80 Nukleotiden, sind in charakteristischer, kleeblattartiger Weise gefaltet und können an einem ihrer Enden eine bestimmte Aminosäure binden. In der mittleren Schleife einer tRNA-Kette befindet sich ein Basentriplet, das *Anticodon*, welches komplementär zu einem Codon ist, das die gebundene Aminosäure kodiert. Es gibt 61 verschiedene tRNA-Moleküle, eines für jedes Aminosäuren-kodierende Codon. Für die korrekte Beladung einer tRNA mit seiner Aminosäure sorgen wiederum bestimmte Proteine, die *Aminoacyl-tRNA Synthetasen*.

Im Ribosom lagern sich tRNA-Bausteine entsprechend der Basenpaarungen entlang der mRNA an. Die Aminosäuren benachbarter tRNA-Moleküle werden verkettet und lösen sich von der tRNA. Die aminosäurefreien tRNA-Ketten lösen sich daraufhin von der mRNA. Die entstehende Aminosäurenkette faltet sich bereits während ihrer Bildung. Der Faltungsprozess wird nach Abschluss der Translation vervollständigt. Erst die gefaltete Aminosäurenkette wird als Protein bezeichnet, welches seine Funktion im Organismus erfüllen kann.

2.2 Transkriptionelle Genregulation

Nachdem in Unterabschnitt 2.1.3 der grundsätzliche Ablauf der Transkription eines Gens beschrieben wurde, soll dieser Abschnitt deren Regulation behandeln. Die transkriptionelle Regulation ist die bedeutendste Möglichkeit, Einfluss auf die Genexpression zu nehmen. Der Einfluss wird durch Proteine, den *Transkriptionsfaktoren*, welche auf der DNA binden können, ausgeübt.

Welche Bedeutung die transkriptionelle Regulation für höher organisierte Lebewesen hat, beleuchtet Unterabschnitt 2.2.1. Den Transkriptionsfaktoren ist Unterabschnitt 2.2.3 gewidmet. Zuvor werden in Unterabschnitt 2.2.2 die wesentlichen Gemeinsamkeiten der Promotorsequenzen von Genen besprochen.

2.2.1 Bedeutung der transkriptionellen Genregulation

Ein wesentlicher Teil der proteinkodierende Gene eines eukaryontischen Lebewesen kodiert Proteine, die im Zusammenhang mit der Regulation anderer Gene stehen. Beim Menschen belaufen sich die Schätzungen auf ca. 3.000 Transkriptionsfaktoren bei einer geschätzten Gesamtzahl an Genen von ca. 25.000 [Lat98, Lev03]. Die Komplexität einer Art korreliert nicht zwingend mit der Anzahl der Gene, sondern vielmehr mit dem Anteil von Transkriptionsfaktoren am Gesamtgenom. Dieser Anteil liegt bei mutmaßlich komplexen Lebewesen, wie etwa den Säugetieren, bei 10%, bei den einzelligen Eukaryonten, wie der Hefe, noch unter 5%. Ebenso ist bei komplexeren Lebewesen ein Ansteigen des Anteils regulativer Sequenzen am Genom zu beobachten. Ein typisches menschliches Gen besitzt mehrere *Enhancer* (siehe Abschnitt 2.2.2), die sich in einem Bereich mehrerer tausend Basenpaare befinden [Arn97]. Schätzungsweise ein Drittel des menschlichen Genoms ist für die Regulation der Aktivität der Gene verantwortlich [Lat98]. Die kodierenden Sequenzen der Gene beanspruchen lediglich 2% aller Basenpaare im menschlichen Genom. Ein typischer Hefepromotor beschränkt sich hingegen auf wenige hundert Basenpaare und besteht im Wesentlichen aus einer Erkennungssequenz für den RNA-Polymerase-Komplex sowie wenigen Bindungsstellen für Transkriptionsfaktoren.

Es scheint offensichtlich, dass komplexere Lebewesen einen höheren Regulationsbedarf haben. Viele Genprodukte werden nur in bestimmten Zelltypen benötigt. Andere werden erst in Folge eines physiologischen Reizes oder weiterer Signale produziert. Einige Gene werden nur während eines kurzen Zeitraumes der Entwicklung des Lebewesens exprimiert und sind während der restlichen Lebenszeit komplett inaktiv. Andere, so genannte *Haushaltsgene*, werden in nahezu allen Zellen in relativ konstanter Menge exprimiert. In den folgenden vier Abschnitt werden die vier wichtigsten Facetten der Genregulation besprochen. Für eine detaillierte Übersicht empfiehlt sich die Monographie von David S. Latchman [Lat98].

Reaktion auf physiologische Situationen. Jede Zelle, ob Prokaryonten oder eine beliebige Gewebezelle eines Säugetier, reagiert auf bestimmte Umwelteinflüsse mit einer Anpassung der Expressionsrate bestimmter Gene. Vereinfachend kann davon ausgegangen werden, dass ein bestimmter Reiz einen Transkriptionsfaktor aktiviert. Jedes Gen, dessen Expressionsrate infolge des Reizes angepasst werden muss, besitzt innerhalb seines Promotors eine entsprechende Bindungsstelle. Ein einfaches Beispiel ist die Regulation durch den *Hitzeschockfaktor* (HSF). Werden Zellen einer zu hohen Temperatur ausgesetzt, führt das zu einem Ansteigen der Expression bestimmter Hitzeschockgene. In den Promotoren dieser Gene befindet sich ein Bindungsmotiv für den Transkriptionsfaktor *HSF*. HSF liegt dabei schon vor dem Hitzereiz in ausreichender Konzentration vor. Erst durch Hitzeeinfluss und einer dadurch bedingten Konformitätsänderung ermöglichen es HSF, an den entsprechenden Bindungsstellen anzudocken. In komplexeren Regelkreisen unterliegen die für Transkriptionsfaktoren kodierenden Gene ihrerseits der Regulation durch andere Transkriptionsfaktoren oder sogar durch ihr eigenes Produkt. Häufig werden Aktivierungsvorgänge auch durch Hormone ausgelöst, die durch Reaktion mit einem membranständigen Rezeptorprotein eine Signalkette auslösen, die zur Aktivierung bzw. Synthese bestimmter Transkriptionsfaktoren führt, die wiederum weitere Gene regulieren.

Zelldifferenzierung. Mehrzellige, eukaryontische Lebewesen besitzen verschiedene Zelltypen, die bestimmte Aufgaben übernehmen. Das erfordert eine differenzierte Aktivierung von jeweils den Genen, deren Produkte die Merkmalsausprägungen dieser Zellen ermöglichen und die zellspezifischen Aufgaben übernehmen. Die zelltypspezifische Genexpression wird durch Transkriptionsfaktoren reguliert, die nur in diesen Zelltypen auftreten.

Das klassische Beispiel für den Einfluss der transkriptionellen Genregulation auf die Zelldifferenzierung sind die *HOX-Gene* des *Hoxclusters*. Dabei handelt es sich um eine Gruppe von Genen, die hintereinander angeordnet sind und die alle eine bestimmte Domäne, die *Homeodomäne*, besitzen[8]. Mit dieser binden sie als Transkriptionsfaktor an den Promotor weiterer Hoxgene. Auf diese Weise wird während der Zelldifferenzierung eine Kaskade von Genaktivierungen ausgelöst, die dazu führt, dass sich entlang der Körperachse verschiedene Gliedmaßen entwickeln, je nachdem, welche Gene der Kaskade in einer bestimmten Phase aktiv sind.

Transkriptionsfaktoren und Krankheiten. Die gestörte Regulation bestimmter Gene ist Ursache vieler Krankheiten. Die Störung kann einerseits durch eine sich negativ auswirkende Mutation des regulierenden Transkriptionsfaktors auftreten, andererseits auch durch eine ungünstige Mutation der Bindungsstellen eines Transkriptionsfaktors.

Insbesondere die Entstehung von Tumoren ist meist durch eine fehlerhafte Regulation bestimmter Gene zurückzuführen, vor allem solcher Gene, die direkt an der Steuerung

[8] Alle Hoxcluster-Gene sind durch mehrfache Duplikation aus einem Original entstanden.

des Zellzyklus beteiligt sind. Übermäßige Expression von Zyklinen bei Lymphomen oder Lungenkarzinomen ist ein Beispiel für die Störung eines Zellzyklus-Aktivators. Die Inaktivierung von Zellzyklus-Bremsen durch Genmutationen ist ebenfalls ein wichtiger Beitrag zur Entstehung von Krebszellen. Inhibitoren von Zyklin-abhängigen Kinasen, wie beispielsweise *p16* oder *p27*, sind bei menschlichen Tumoren oft inaktiviert. Schließlich spielen im Zellzyklus auch Tumorsuppressor-Gene, wie beispielsweise das Retinoblastom-Gen oder das *p53*-Gen eine wesentliche Rolle.

2.2.2 Regulative Sequenzen eukaryontischer Gene

Transkriptionsfaktoren binden an kurzen DNA-Abschnitten, den *Transkriptionsfaktorbindungsstellen* (TFBS), die vornehmlich oberhalb des 5′-Endes der kodierenden Sequenz eines Gens liegen. Eine TFBS kann genau ein TF-Molekül binden und ist je nach Faktor zwischen 5 und 30 Basenpaare lang. Jeder TF bevorzugt zur Bindung eine für ihn charakteristische Nukleotidfolge, für die er höchste Bindungsaffinität besitzt. TF tolerieren jedoch bis zu einem gewissen Grad Abweichungen von dieser optimalen TFBS. Die erlaubte Variabilität unterscheidet sich zwischen verschiedenen Transkriptionsfaktoren und zwischen den Positionen der TFBS eines Faktors. So können einige Basen der optimalen TFBS obligatorisch für eine erfolgreiche Bindung sein, andere lediglich begünstigend wirken. Neben der Nukleotidfolge einer TFBS sind auch die lokalen, strukturellen Eigenschaften der DNA, die in Unterabschnitt 2.1.1 auf Seite 9 beschrieben worden, wichtig für die Bindung eines TF-Moleküls. In vielen Fällen erkennen die TF die Nukleotide ihrer Bindungsstellen sogar in geringerem Maße als die strukturellen Abweichungen. Die auffälligen Sequenzähnlichkeiten unter den TFBS sind dadurch erklärbar, dass ähnliche Sequenzen ein ähnliches strukturelles Profil ermöglichen, an dass sich der TF optimal anlagern kann. Beispielsweise tritt TBP (*TATA-Box binding protein*) nur mit den beiden zentralen Nukleotiden der TATA-Box tatsächlich in Wechselwirkung [Kim94], ansonsten korreliert die Bindungsaffinität von TBP stark mit der DNA-Biegsamkeit [Sta95]. Diese ist vor allem für `T`- und `A`-reiche Sequenzen hoch, was den Anteil dieser Nukleotide in den TATA-Boxen erklärt.

Promotor. Historisch wird der gesamte Sequenzbereich oberhalb des Transkriptionsstartpunkts (TSS), innerhalb welchem Transkriptionsfaktoren binden, als *Promotor* eines Gens bezeichnet. In jüngeren Veröffentlichungen [Arn03, Alv03] wird als Promotor ein Sequenzabschnitt einer ungefähren Länge von 200-500 bp bezeichnet, der unmittelbar oberhalb der TSS liegt, und hauptsächlich die nicht genspezifischen GTF des RNA-Polymerase-Komplexes bindet. Häufig werden innerhalb des Promotors der *Kernpromotor* und der *proximaler* Promotor unterschieden [Wer99]. Während der Kernpromotor eine Minimalumgebung für eine erfolgreiche Transkription darstellt, enthält der sich an den Kernpromotor anschließende proximale Promotor Bindungsstellen für eine bestimmte Klasse von Trankriptionsfaktoren, die an der Regulierung der meisten Gene

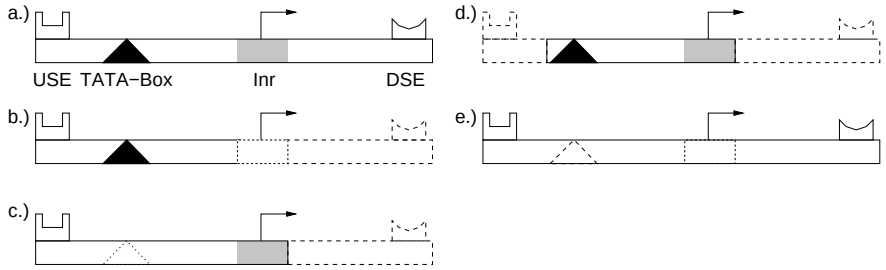


Abbildung 2.6: Schematischer Aufbau eines Kernpromotors. a.) Übersicht aller Kernpromotor-Elemente. b.)-e.) Vier verschiedene Konfigurationen. [Wer99]

beteiligt sind. Diese Faktoren erleichtern die Bindung des RNA-Polymerase-Komplexes, sind allerdings häufig nicht unabdingbar für eine gewisse Grundaktivität des Gens.

Kernpromotor. In Kernpromotoren treten eine Reihe von so genannten *Kernpromotorelementen* auf. Das wichtigste Element ist die in Unterabschnitt 2.1.3 auf Seite 14 beschriebene TATA-Box, welche von dem GTF TFIID erkannt wird. Vor allem in Haushaltsgenen — das sind Gene, welche Proteine kodieren, die in relativ konstanter Konzentration in nahezu allen Zellen benötigt werden — fehlt die TATA-Box häufig. Der Transkriptionskomplex ist dann auf weitere Kernpromotorelemente angewiesen:

- **Inr** für Initiatorregion. Sie bezeichnet einen die TSS umschließenden Sequenzbereich, der jedoch nicht in dem Maße konserviert ist wie die TATA-Box.
- **DSE** für *downstream element*. Dieses Sequenzmotiv befindet sich, wenn es vorhanden ist, etwas 30 bp unterhalb der TSS.
- **USE** für *upstream element* oberhalb einer eventuellen TATA-Box.

Abbildung 2.6 stellt die Lage der möglichen Promotorelemente schematisch dar und zeigt vier tatsächlich auftretende Konfigurationen von Kernpromotoren. Mit Hilfe weiterer, im proximalen Promotor bindenden Faktoren ist eine erfolgreiche Transkription in manchen Fällen sogar bei Abwesenheit jeglicher Kernpromotorelemente möglich. In einem solchen Fall spricht man von einem *Nullpromotor*.

Enhancer und Silencer. Oberhalb des Promotors treten meist weitere regulative Sequenzen auf, die hinsichtlich Lage, Orientierung und Funktion weniger festgelegt sind als der Promotor selbst. Auffällig ist lediglich die Tendenz von Bindungsstellen, gruppiert in *TFBS-Modulen* aufzutreten und nicht völlig frei in dem oft mehrere tausend Basenpaare umfassenden Bereich verteilt zu sein, der potentiell für regulative Prozesse zur Verfügung stünde [Mur04, Ber02]. Komplexbildung mehrerer Transkriptionsfaktoren zu größeren Einheiten, die nur so ihre Wirkung entfalten können, bedingen die Nachbarschaft der entsprechenden Bindungsstellen. Gemeinsam prägen die TFBS eines Moduls

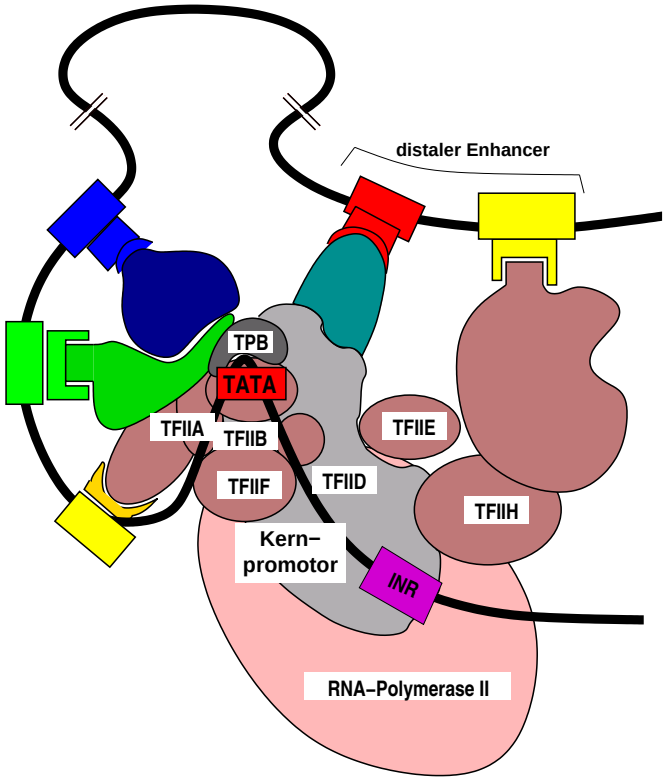


Abbildung 2.7: TFBS-Module und ihre Wechselwirkungen [Wer99]

bzw. die beteiligten Transkriptionsfaktoren einen spezifischen Teil des Expressionsverhaltens eines Gens, insbesondere eine erhöhte Expression als Reaktion auf einen bestimmten Reiz oder die Abschaltung des Gens in bestimmten Zellstadien. Ist die Gesamtwirkung eines Moduls vorwiegend aktivierend, wird es als *Enhancer* (englisch für Verstärker) bezeichnet, und im Falle einer überwiegend repressiven Wirkung als *Silencer* (englisch für Dämpfer) [Pen01, Arn03, Bla98, Ogb98]. Diese Klassifikation der TFBS-Module ist an die Bedingung geknüpft, dass alle beteiligten Faktoren in der optimalen Konzentration vorliegen. Schon geringe Abweichungen in der Konzentration eines Faktors kann die Aktivität der Genexpression maßgeblich beeinflussen. Der Begriff TFBS-Moduls sollte ferner nicht suggerieren, dass es sich bei einer lokalen Häufung funktionell zusammenhängender Bindungsstellen um eine nach außen abgeschlossene Einheit mit einer aus dem Vorhandensein der beteiligten Faktoren vorhersagbaren Wirkung handelt. Das Ensemble der an einem Modul bindenden Faktoren tritt vielmehr häufig in Wechselwirkung mit weiter entfernt bindenden Transkriptionsfaktoren. Verdeutlicht wird das in Abbildung 2.7.

Die hohe Anzahl der Transkriptionsfaktoren, die ein Gen regulieren können, sowie deren Wechselwirkungen untereinander ermöglichen eine äußerst komplexe und feinfühlige Einstellung der Genexpressionsrate als Antwort auf verschiedene äußere Reizsignale in verschiedenen Geweben oder Zellstadien. Da die für Transkriptionsfaktoren kodierenden

Gene ihrerseits diesen Regulationsmechanismen unterliegen, wird die Regulation eines Gens auf mehrere Stufen verteilt. Dieser Umstand erhöht die Komplexität des Regulationsapparates um ein Weiteres.

Boundary Elements. In wenigen Genen tritt eine weitere Form regulativer Sequenzen auf, die *Boundary Elements* (engl. für Abgrenzungselemente) [Arn03]. Durch Bindung bestimmter Faktoren ermöglichen sie die räumliche Trennung von Enhancer-Bereichen und dem Kernpromotor im Falle einer Inaktivierung des Gens. Aktivierenden Einfluss üben sie dann aus, wenn eine regulative Sequenz vor dem repressiven Einfluss der Chromatinbildung schützen. Besonders in Zusammenhang mit den im vorherigen Abschnitt vorgestellten HOX-Clustern wurden Boundary Elements untersucht [Arn02].

CpG-Inseln. Kein eigener Typ regulativer DNA-Sequenzen, jedoch eine statistische Auffälligkeit, die überwiegend in den Promotoren höher entwickelter Eukaryonten auftritt, sind die *CpG-Inseln*. Vielmehr handelt es sich bei CpG-Inseln um genomische Sequenzabschnitte, in denen ein höherer Anteil an CG-Dinukleotiden auftritt. Der mittlere Anteil an CG-Dinukleotiden in vielen Genomen ist aus Gründen, die im folgenden genannt werden, weitaus niedriger als aufgrund des Anteils der beiden Einzelbasen zu vermuten wäre[9]. Gerade im Bereich von Promotorsequenzen treten CG-Dinukleotide $10 - 20$ mal häufiger auf als im genomweiten Mittel.

Wie kommt es zu diesem Ungleichgewicht? In eukaryontischen Zellen ist ein Teil der Cytosin-Nukleotide methyliert. Im Falle eines CG-Dinukleotids entsteht ein palindromisches Methylierungsmuster. In dieser Konstellation sind die methylierten Cytosine labil und können spontan zu Thymidin-Nukleotiden mutieren. Dieser evolutionäre Druck ist in vielen Promotorsequenzen nicht vorhanden, da diese Bereiche weitgehend unmethyliert bleiben. Der fehlende Druck erklärt das gehäufte Auftreten von CG-Dinukleotiden in regulativen Sequenzen. Aufgrund welcher Einflüsse diese Regionen des Erbguts unmethyliert bleiben, ist bisher nicht vollständig geklärt worden. In einigen Fällen steht dies offenbar in Verbindung mit dem Transkriptionsfaktor Sp1 [Mac94].

Die Promotoren von etwa 56% aller menschlichen Gene enthalten eine CpG-Insel, davon nahezu alle Haushaltsgene und 44% aller gewebespezifischen Gene [Ant93]. Diese Überrepräsentation machte die CpG-Inseln zu einem interessanten Aspekt für computergestützte Verfahren zur Promotoren- und Generkennung [Ped99]. Die in der Literatur genannten Kriterien für das Vorhandensein einer CpG-Insel unterscheiden sich stark, die meisten Arbeiten stützen sich jedoch auf eine Definition von Gardiner-Garden und Frommer [GG87a]. Danach ist eine CpG-Insel ein Sequenzabschnitt von mindestens 200 bp, der einen G +C-Gehalt von über 50% hat und in dem die Häufigkeit von CG-Dinukleotiden mindestens 60% von jener theoretischen Häufigkeit entspricht, die aufgrund der Einzelhäufigkeiten der Basen C und G erwartet werden müsste.

[9] im menschlichen Genom beträgt der Anteil an CG-Dinukleotiden 0.8% statt erwarteten 4%.

2.2.3 Transkriptionsfaktoren

Wurden im vorherigen Unterabschnitt jene Sequenzabschnitte eines Gens beschrieben, welche aufgrund ihrer Beschaffenheit den Akteuren der transkriptionellen Regulation ihre Arbeit ermöglichen, soll sich dieser Unterabschnitt mit eben diesen Akteuren beschäftigen. *Transkriptionsfaktoren* sind Proteine, die im Wesentlichen zwei Eigenschaften aufweisen: 1.) die Möglichkeit, auf der DNA zu binden und 2.) eine entweder aktivierende oder repressive Wirkung auf die Expression eines oder mehrerer Gene [Lat98]. Beide Eigenschaften manifestieren sich in funktionellen Teilstrukturen des jeweiligen Transkriptionsfaktors — den *Domänen*. Hinsichtlich jener Domäne, welche die Bindung mit der DNA herstellt, weisen viele Transkriptionsfaktoren starke Ähnlichkeiten auf, die auf eine evolutionäre Verwandtschaft dieser Faktoren schließen lassen. Faktoren mit ähnlicher Bindungsdomäne werden hierarchisch in Klassen zusammengefasst. Die drei Hauptklassen, denen sich die große Mehrheit aller Transkriptionsfaktoren zuordnen lassen, sind:

1. *Homeodomänen-Faktoren.* Dieser Klasse gehören unter nderem die für die embryonale Differenzierung von Zellgeweben wichtigen Hox-Transkriptionsfaktoren an. Eine weitere bedeutsame Variante der Homeodomäne ist die POU-Domäne, die beispielsweise in dem häufig auftretenden Transkriptionsfaktor Oct1 vorkommt. Die Homeodomäne besteht aus vier α-Helix-Abschnitten. Die ersten beiden verlaufen zueinander gegenläufig. Die beiden Hinteren sind im rechten Winkel zu den ersten beiden orientiert. Dieser Winkel wird durch eine zwischen den beiden Zweiergruppen gelegene β-Kurve hergestellt. Diese Anordnung drückt sich in dem ebenfalls gebräuchlichen Namen, dem *Helix-Turn-Helix-Motif* aus (siehe Abbildung 2.8) [Kor93]. Im Falle einer Bindung auf der DNA ordnet sich die zweite α-Helix quer zur großen Furche der DNA aus, während die Dritte teilweise entlang der großen Furche liegt. Dadurch werden Kontakte zwischen Aminosäuren der dritten Helix mit den Basen der DNA ermöglicht. Die genaue Abfolge der Aminosäurenreste in der dritten α-Helix verschiedener Homeobox-Proteinen bestimmen jeweils die passende DNA-Bindungssequenz. Der Nachweis dafür wurde erbracht, indem durch gezielten Austausch einzelner Aminosäuren das Bindungsverhalten eines anderen Faktors erreicht werden konnte. Bindungssequenzen von Homeodomänen-Faktoren sind häufig reich an `A`- und `T`-Basen.
2. *Zinkfinger-Faktoren.* Diese Faktoren enthalten mehrfache Kopien eines stark konservierten Motivs, dem *Zinkfinger*. Dieses Motiv besteht aus etwa 12 Aminosäuren, darunter häufig sowohl zwei Cystesinreste als auch zwei Histidinreste. Diese vier Aminosäuren sind gemeinsam in der Lage, ein Zinkatom zu binden. Namensgebend für das Motiv ist neben dem Zink die Struktur, die sich durch Einschluss des Zinkatoms ergibt. Das Zinkfingermotiv ist verantwortlich für die DNA-Bindung dieser Faktoren. Ein isolierter Finger reicht dafür nicht aus. Vielmehr wird eine Bindung erst durch Wechselwirkungen der mehrfach auftretenden Zinkfinger möglich [Eva88]. Jedes vorkommende Fingermotiv erkennt drei DNA-Basen. Das Bindungsmotiv eines Zinkfinger-Faktors ergibt sich aus der Verkettung dieser Einzelmotive, die sich aufgrund von Unterschieden in der präzisen Folge von Aminosäuren leicht

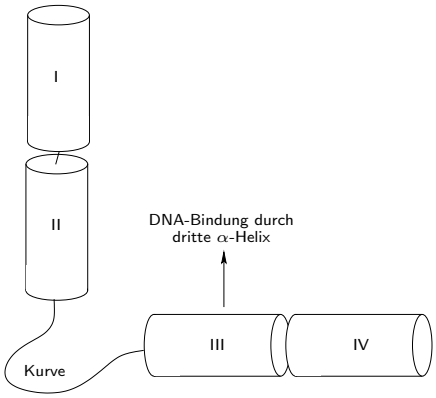


Abbildung 2.8: Schematische Darstellung einer Homeodomäne

unterscheiden können, und aus eventuellen Zwischensequenzen zwischen den Einzelmotiven. In der Regel bestehen die DNA-Bindungssequenzen überwiegend aus *C*- und *G*-Basen. Bekannte Vertreter sind der Transkriptionsfaktor *Sp*1, der an der Regulation vieler Gene beteiligt ist, und der GTF TFIIA.

3. *Leucin-Zipper-Faktoren.* Das dritte bedeutende Motiv stellt die DNA-Bindung nicht direkt her. Vielmehr ermöglicht es die Dimerisierung zweier Proteine, die das Motiv besitzen. Durch diese Protein-Protein-Bindung werden die eigentlichen DNA-Bindungsdomänen der beiden Proteine in eine für eine DNA-Bindung strukturell günstige Position gebracht. Das *Leucin-Zipper-Motiv* ist eine Aminosäurensequenz, die eine α-Helixstruktur ausbildet. Jede zweite Windung enthält die Aminosäure Leucin, die stets zur gleichen Seite der Helix gewandt liegt. Die Leucinreste zweier Proteine treten bei der Dimerisierung in Wechselwirkung. Die Form der Bindung zeigt Ähnlichkeit mit einem Reißverschluss (daher die Bezeichnung *zipper*: engl. für Reißverschluss). Die DNA-Bindungsdomänen liegen in unmittelbarer Nachbarschaft zu dem Leucin-Zipper-Motiv (siehe Abbildung 2.9) [Lam91]. Hier sind verschiedene Strukturen möglich, jeweils mit einer charakteristischen DNA-Bindungssequenz. In der Regel haben nur dimerisierte Leucin-Zipper-Faktoren die Fähigkeit, auf der DNA zu binden. Ein defektes Leucin-Zipper-Motiv nimmt dem betroffenen Faktor diese Fähigkeit, da er nicht mehr dimerisieren kann. Dagegen können Dimere bestehend aus bestimmten Faktorkombinationen unter Verwendung ausschließlich einer Bindungsdomäne auf der DNA binden. Ein bekanntes Beispiel hierfür stellt das Faktorenpaar *MyoD* und *E12* dar, bei dem die DNA-Bindung über E12 hergestellt wird. Weitere bekannte Leucin-Zipper-Faktoren sind die *Jun*- und *Fos*-Transkriptionsfaktoren, deren Bindungssequenz als *AP1-Box* bezeichnet wird.

Die vielfältigen Möglichkeiten der Einflussnahme von Transkriptionsfaktoren auf die Transkription können im Rahmen dieser Arbeit nur ansatzweise besprochen werden. Wie zu Anfang dieses Abschnittes angedeutet, sind Transkriptionsfaktoren meist mo-

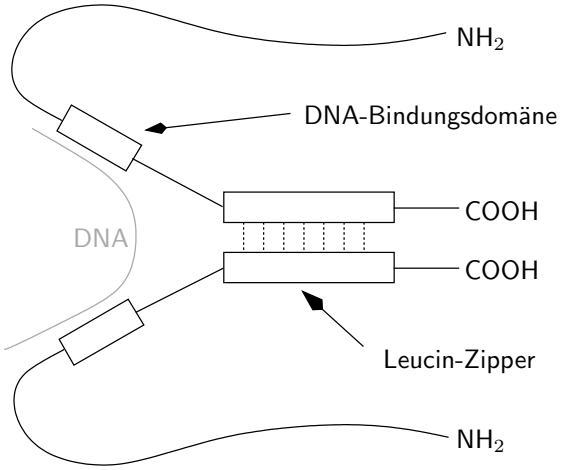


Abbildung 2.9: Leucin-Zipper-Faktoren: Dimerisierung und DNA-Bindung

dular aufgebaut. Neben der DNA-Bindungsdomäne gibt es häufig mindestens eine weitere Domäne, welche die aktivierende oder repressive Wirkung eines Transkriptionsfaktors vermittelt. Diese lassen sich anhand sequentieller oder struktureller Eigenschaften jedoch nicht in klar abgegrenzte Klassen einteilen. Lediglich das häufige Auftreten einer bestimmten Gruppe von Aminosäuren innerhalb der Domäne ist auffällig und ermöglicht die Einteilung in *saure* Domänen[10], *Glutaminreiche* Domänen und *Prolinreiche* Domänen. Eine genaue Wirkungsweise lässt sich anhand dieser Einteilung nicht ableiten. Experimente zeigten jedoch, dass Faktoren mit sauren Domänen im Gegensatz zu Faktoren mit Glutamin-reichen Domänen auch Wirkung entfalten können, wenn sie weit entfernt von der TSS binden. Auch Prolin-reiche Faktoren zeigen nur schwache Wirkung, wenn sie entfernt binden [Sei92].

Die Wirkung eines Transkriptionsfaktors kann einerseits aktivierend sein, andererseits repressiv. Domänenvermittelter Einfluss ist jedoch meistens aktivierend, wohingegen eine repressive Wirkung häufig indirekt, z.B. durch Blockieren der Bindungsstellen für einen aktivierenden Faktor, zustande kommt. Je nach Konfiguration der regulativen Sequenzen können einige Transkriptionsfaktoren auch beide Effekte erzielen. Eine aktivierende Wirkung wird häufig durch direkte Wechselwirkungen der aktivierenden Domäne mit Teilen des Polymerase-Komplexes hergestellt [Gua88]. So erleichtern bestimmte Faktoren durch eine Bindung an dem GTF TFIID dessen Bindung an der TATA-Box eines Promotors, was die Rate des auf Seite 14 beschriebenen Initiierungsprozess erhöht. Wieder andere Faktoren wirken in ähnlicher Weise aktivierend bei der Rekrutierung weiterer GTF wie TFIIB an einen bereits gebundenen TFIID. Die Wirkung kann auch in einer Erhöhung der Stabilität eines finalen Initiierungskomplexes während der Transkription bestehen [Cho93]. Nachgewiesen sind darüber hinaus die Regulation der Verkettungsarbeit der RNA Polymerase II. Bei einigen Proteinen bricht die Verkettung vorzeitig ab.

[10] Übergewicht an Aminosäuren mit sauren Seitenketten

Die Rate, mit der komplette, funktionelle mRNA-Moleküle hergestellt werden, unterliegt in diesen Fällen ebenfalls bestimmten Transkriptionsfaktoren. Neben dem direkten Einfluss auf Komponenten des Transkriptionskomplexes wirken einige Faktoren indirekt aktivierend, indem sie die strukturelle Basis für die Bindung weiterer Faktoren schaffen. Chromatinmodulierende Transkriptionsfaktoren[11], sorgen durch ihre Bindungen für eine Verschiebung der Histonkomplexe eines Nukleosoms. Dadurch werden vorher verborgene Bindungsstellen zugänglich für die jeweiligen Transkriptionsfaktoren [Bea97].

Die häufigsten repressiven Einflussmechanismen von Transkriptionsfaktoren sind ebenfalls indirekter Natur. Zum einen können funktionell bedeutsame DNA-Bindungsstellen durch andere Proteine maskiert werden, wodurch eine Aktivierung der Genexpression verhindert wird [Lan97]. Zum anderen kann eine DNA-Bindung eines Transkriptionsfaktors auch durch die Bindung eines Proteins an seiner DNA-Bindungsdomäne verhindert werden. Eine weitere repressive Wirkung ist durch eine Blockade der aktivierenden Domäne eines Faktors, etwa durch Bindung eines weiteren Proteins, gegeben. Einem Transkriptionsfaktor wird eine direkte repressive Wirkung zugesprochen, wenn diese in der Wechselwirkung mit Komponenten des Polymerasekomplexes und deren dadurch verminderten Aktivität besteht, wobei die Wirkungsweisen denen entsprechen, die zuvor als indirekt repressive Wirkung über dritte Faktoren genannt wurden.

2.3 Molekularbiologische Labortechniken

Im Rahmen dieser Arbeit werden Mustererkennungsansätze vorgestellt, deren Daten zum Lernen und Evaluieren auf verschiedenen molekularbiologischen Experimenten beruhen. Um dem Leser ein Gefühl dafür zu geben, wie gesichert diese Daten sind, und wo die Risiken einzelner Verfahren liegen, werden in diesem Abschnitt die wichtigsten experimentellen Verfahren vorgestellt, die zur Aufklärung der transkriptionellen Genregulation eingesetzt werden.

2.3.1 DNA mobility shift assay

Das *DNA mobility shift assay* ist ein Verfahren zur Beantwortung der Frage, welche Transkriptionsfaktoren eine bestimmte DNA-Sequenz binden. Eine solche Fragestellung ergibt sich immer dann, wenn das Expressionsmuster eines unbekannten Gens erklärt werden soll. Die untersuchte DNA-Sequenz ist meist ein Teil des Promotors des Gens. Es soll untersucht werden, welche Transkriptionsfaktoren dieses Gen regulieren. Grundlage der Methode ist die *Gelelektrophorese*. Dabei wird eine Probe am Rand eines rechteckigen, gelartigen Mediums aufgetragen. Aufgrund einer an das Gel angelegten elektrischen Spannung bewegen sich die Bestandteile der Probe an das gegenüberliegende Ende des

[11] auch unter der Bezeichnung Architekturproteine bekannt

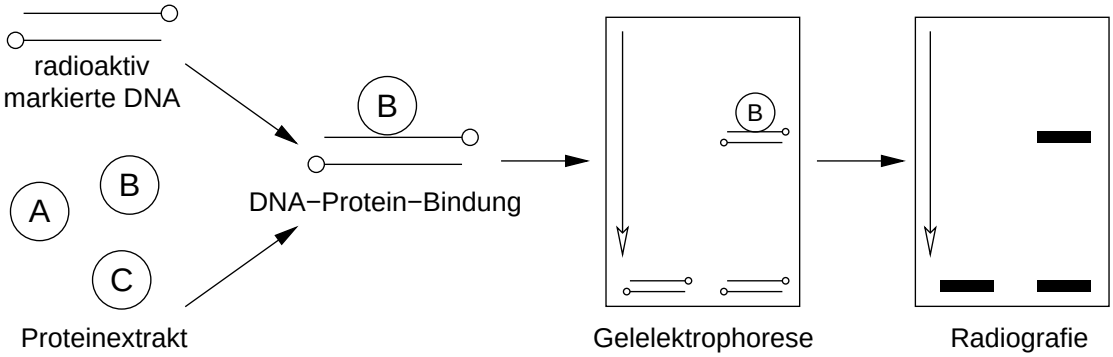


Abbildung 2.10: Ablauf eines DNA *mobility shift assays*. Transkriptionsfaktoren, die eine kurze, radioaktiv markierte DNA-Sequenz binden, können anschließend über eine Gelelektrophorese extrahiert werden.

Gels. Entscheidend ist hierbei, dass sich Moleküle mit kleinem Molekulargewicht schneller durch das Medium bewegen als entsprechend schwerere Moleküle. Besteht die Vermutung, dass die Probe zwei verschiedene Typen von Molekülen mit stark unterschiedlichem Gewicht enthält, kann dies durch die Bildung zweier Banden im Gel bestätigt werden. Insbesondere ermöglicht die Bandenbildung die Trennung der Bestandteile der Probe.

Beim *DNA mobility shift assay* werden zunächst die zu untersuchenden DNA-Sequenzen radioaktiv markiert, um sie später auf dem Gel radiografisch sichtbar zu machen. Die DNA wird anschließend einer Probe aller in einer bestimmten Zelle vorkommenden Moleküle ausgesetzt. Es bilden sich unter Umständen Bindungen zwischen der DNA-Sequenz und Proteinen aus. Diese werden dann als langsame Bande im Gel identifiziert. Die gebundenen Faktoren können in weiteren Schritten aus der Bande isoliert und identifiziert werden. Treten dagegen keine DNA-Proteinbindungen ein, so bildet sich keine langsame Bande im Gel aus. Die grundlegende Vorgehensweise ist in Abbildung 2.10 dargestellt. Häufig werden solche Experimente mit Extrakten aus verschiedenen Geweben parallel durchgeführt, denn vielfach treten Transkriptionsfaktoren nur in bestimmten Zelltypen auf. Der Nachweis einer DNA-Bindung in einem Extrakt eines bestimmten Typs lässt Rückschlüsse auf die Funktion des untersuchten Gens zu. Neben der Bestimmung von relevanten Transkriptionsfaktoren kann *DNA mobility shift assay* auch zur Bestimmung der optimalen Bindungssequenz für einen Faktor verwendet werden. Ausgangspunkt ist eine experimentell bestätigte Bindungssequenz, die als radioaktiv markierte DNA-Sequenz vorliegt. Der Experimentierprobe wird nun eine Menge einer zweiten DNA-Sequenz hinzugefügt, die nicht radioaktiv markiert ist. Diese sollte in einer um Dimensionen größeren Konzentration vorliegen als die markierte DNA-Sequenz. Handelt es sich bei markierter und unmarkierter DNA um dieselbe Nukleotidfolge, so ist der untersuchte Transkriptionsfaktor in der Lage, auch die unmarkierte DNA-Sequenz zu binden. Aufgrund der Mengenverhältnisse wird er dies auch bevorzugt und zu Lasten der markierten Sequenz tun. In einer anschließenden Gelelektrophorese ist in diesem Fall keine bedeutsame Bande zu erkennen. Unterscheidet sich die unmarkierte Sequenz zu stark

von der Bindungssequenz des Faktors, kann der Faktor nur an der markierten Sequenz binden. Man beobachtet die erwartete Bande der radioaktiv markierten und gebundenen DNA. Durch gezielte Punktmutationen der anfänglich bekannten und markierten DNA-Sequenz werden schrittweise unmarkierte Proben hergestellt. Der Grad der Auslöschung der DNA-Protein-Bande ist ein Maß für die Bindungsaffinität des Faktors mit der unmarkierten Sequenz. Die optimale Bindungssequenz wird Schritt für Schritt bestimmt. Obwohl dieses Verfahren sehr zeitaufwendig ist, weißt es doch gegenüber automatisierten Methoden, wie dem in Unterabschnitt 2.3.4 beschriebenem SELEX-Verfahren, eine höhere Aussagekraft der Ergebnisse auf.

2.3.2 DNaseI footprinting assay

Das Verfahren *DNaseI footprinting assay* (englisch für *DNaseI* Fußabdruck-Untersuchung) ermöglicht die Lokalisierung von DNA-Transkriptionsfaktor-Interaktionen innerhalb einer DNA-Sequenz. Dabei wird die Tatsache ausgenutzt, dass DNA an Stellen, an denen ein Protein gebunden ist, zu einem gewissen Grad vor enzymatischen Spaltungen geschützt ist. Namensgebend ist das Enzym *Desoxyribonuklease* (kurz: *DNase*), welches zur DNA-Spaltung verwendet wird.

Die untersuchten DNA-Ketten sind am Ende eines der beiden Stränge radioaktiv markiert. Analog zum *DNA mobility shift assay* werden die DNA-Doppelstränge einer Probe von Zellproteinextrakt oder einer Probe von spezifisch ausgewählten Transkriptionsfaktoren zugegeben. Es bilden sich unter Umständen DNA-Protein-Bindungen aus. Durch Zugabe von DNase entstehen DNA-Bruchstücke verschiedenster Länge. Insbesondere sind jene Bruchstücke interessant, die das markierte Ende enthalten, denn nur diese sind bei der sich anschließenden Gelelektrophorese sichtbar. Nach Wirkung der DNase werden die Proteine von der DNA getrennt. In der Gelelektrophorese ergibt sich eine fast kontinuierliche Verteilung der Bruchstücke, da radioaktiv markierte Bruchstücke fast jeder Länge vorkommen und deren zurückgelegter Weg im Gel pro Zeiteinheit proportional zu ihrer Länge ist. Da die DNase an Transkriptionsfaktor-gebundenen Stellen nicht spalten konnte, fehlen markierte Bruchstücke mit einer Länge in einem bestimmten Intervall. Eine solche Lücke im visualisierten Gel wird als *Fußabdruck* (englisch: footprint) eines Proteins bezeichnet (siehe Abbildung 2.11). Die Position eines Fußabdrucks ermöglicht Rückschlüsse auf die Position der Bindung des Proteins auf der untersuchten DNA-Sequenz. Fußabdrucktechniken haben gegenüber dem *DNA mobility shift assay* den Vorteil einer präziseren Lokalisierung der Proteinbindung auf der DNA-Sequenz. Insbesondere eine verwandte Methode, das *dimethyl sulphat protection footprinting*, welches ebenfalls auf dem Schutz proteingebundener DNA-Abschnitte vor äußeren Einflüssen basiert, spielt diesen Vorteil bis zur basenpaargenauen Bestimmung der Bindungsstelle aus. Hierbei wird die Eigenschaft von Dimethylsulfat (DMS) genutzt, Guanin-Basen zu methylieren. Mit Hilfe von Piperiden können die DNA-Moleküle an der methylierten Stelle gespalten werden. Spaltprodukte in einem Bereich der Proteinbindung fehlen, was in der sich anschließenden Gelelektrophorese sichtbar wird [Spi98, Las89].

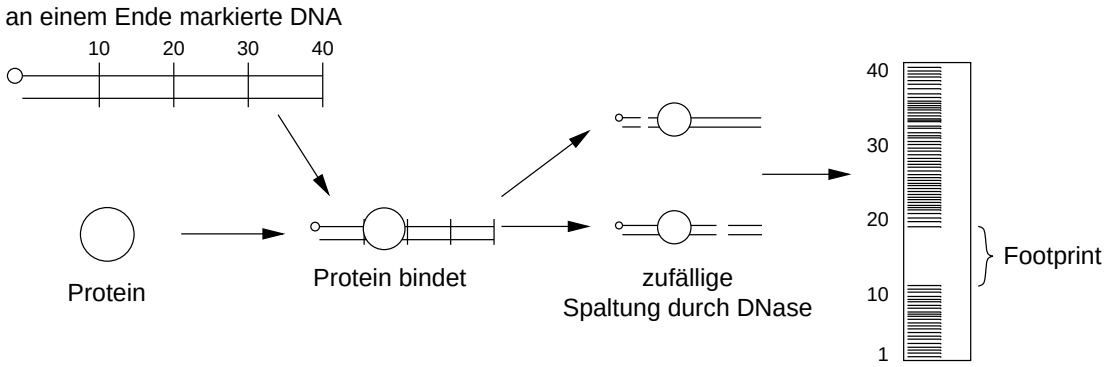


Abbildung 2.11: Ablauf eines *DNaseI footprinting assay*. Nach Bindung eines Transkriptionsfaktors wird die DNA-Sequenz durch DNase in viele Stücke zerlegt. Ein anschließendes Gel erzeugt Banden für Bruchstücke fast jeder Länge. Da der Transkriptionsfaktor jedoch die DNA vor der DNase schützt, fehlen Bruchstücken bestimmter Längen. So kann die Position einer DNA-Protein-Bindung bestimmt werden.

2.3.3 Methylation-Interferenz-Untersuchung

Wie bei der im vorherigen Unterabschnitt eingeführten DMS-Fußabdruck-Methode wird bei der *Methylation-Interferenz-Untersuchung* die Methylierungsfähigkeit von DMS ausgenutzt. Jedoch interessiert hier nicht der Schutz dieses Vorganges durch ein gebundenes Protein, sondern die Auswirkung einer bestimmten methylierten G-Base auf die Bindungsaffinität des Proteins.

Dabei werden zunächst die DNA-Moleküle dem DMS ausgesetzt, indem im Durchschnitt in jeder DNA-Kette lediglich ein G methyliert wird. Anschließend wird ein *DNA mobility shift assay* (siehe Unterabschnitt 2.3.1) durchgeführt. Es bilden sich Banden mit und ohne gebundenem Protein. Beide Banden werden nun getrennt der Spaltung durch Piperiden unterzogen. Falls eine spezifische methylierte G-Base die Proteinbindung verhinderte, wird diese Position anschließend nur in der ungebundenen DNA-Menge vorhanden sein. Hat ein methyliertes G hingegen keinen Einfluss auf die DNA-Bindung gehabt, zeigen sich Spaltprodukte der entsprechenden Länge in beiden Proben.

Die Methylation-Interferenz-Untersuchung erlaubt eine noch höhere Genauigkeit als die DNA-Fußabdruck-Methode, mit der die Bedeutung bestimmter Basen bei der Proteinbindung untersucht werden können. Neben DMS gibt es verwandte Verfahren, die in ähnlicher Weise für Adenin-Basen arbeiten.

2.3.4 SELEX

SELEX (englisch: *Systematic Evolution of Ligands by Exponential enrichment*) ist ein Verfahren, welches die simultane Untersuchung einer riesigen, zufällig verteilten Men-

ge kurzer DNA oder RNA-Abschnitte (*Oligonukleotide*) in Hinblick auf eine bestimmte Eigenschaft ermöglicht [Ell92, Klu94]. Im Zusammenhang mit der Erforschung der transkriptionellen Regulation wird SELEX verwendet, um die bevorzugten Bindungssequenzen eines Transkriptionsfaktors experimentell zu bestimmen. Neben SELEX ist auch die Bezeichnung *in vitro Selektion* gebräuchlich.

Ein typisches SELEX-Experiment mit DNA-Molekülen läuft folgendermaßen ab: Zunächst wird mit Hilfe eines DNA-Oligonukleotid-Synthetisierers eine große Menge DNA-Oligonukleotide, ein *Pool*, erzeugt. Die DNA-Moleküle entstehen durch völlig zufällige Verkettung der vier Nukleotide. An ihren Enden werden Erkennungssequenzen hinzugefügt (englisch: *primer*). Dieser initiale Pool kann bis zu 10^{15} Oligonukleotide enthalten. Dieser enorme Umfang rechtfertigt die Vermutung, dass sich in dieser Menge einige wenige Moleküle mit der gewünschten, untersuchten Eigenschaft befinden.

Diese wenigen Sequenzen werden in mehreren, sich wiederholenden Schritten, selektiert. Für eine Suche nach Bindungssequenzen wird der Transkriptionsfaktor in die Lösung der initilalen DNA-Menge eingebracht. Die Transkriptionsfaktoren erkennen Oligonukleotide, die eine günstige Bindungsstelle enthalten und binden diese. Anschließend erfolgt ein Auswaschen der nicht gebundenen Sequenzen, beispielsweise durch eine Gelelektrophorese. Die verbliebenen Sequenzen werden mittels einer PCR[12] vervielfältigt. Mit der um funktionalen Sequenzen angereicherten Oligonukleotidmenge beginnt der Selektionsschritt von neuem. Die mehrfache Durchführung von Selektion und Waschung erhöht die Wahrscheinlichkeit, dass sich in der finalen Oligonukleotidmenge ausschließlich funktionale DNA-Sequenzen befinden.

Der Vorzüge des SELEX-Verfahrens liegen in der Möglichkeit, funktionelle Oligonukleotide aus einem zufällig erzeugten, in seinem Umfang der Menge aller Sequenzen einer bestimmten Länge vergleichbaren Pool zu bestimmen. Im Allgemeinen ist dafür keinerlei biologisches Vorwissen über die Struktur und Funktion der Sequenzen in der Zielmenge nötig. Bei der Interpretation der Ergebnisse sollten jedoch bekannte Fehlerquellen und deren Effekte des SELEX-Verfahrens berücksichtigt werden. Zunächst muss die Zusammensetzung des initial erzeugten Pools und dessen Abdeckung des Sequenzraums kritisch betrachtet werden. Zwar ist bei der genannten Anzahl $10^{15} \sim 4^{25}$ zu erwarten, dass bei einer Sequenzlänge von 25 jede mögliche Sequenz vorkommt. Die Art ihrer Erzeugung bedingt jedoch, dass die Nukleotide an den einzelnen Positionen völlig gleichverteilt sind, und zudem statistisch unabhängig von Nachbarpositionen. Gerade Letzteres spiegelt nur unzureichend den Sequenzraum eines reellen Genoms wieder. Bei der Selektion der funktionellen Sequenzen kann es vorkommen, dass keine der Sequenzen wirklich die gewünschte Eigenschaft zeigt, trotzdem jedoch einige wenige Sequenzen völlig unspezifisch wegen ihrer Primersequenz erkannt werden. Selbst nach mehrfacher Waschung

[12] Die Polymerase-Kettenreaktion (englisch Polymerase Chain Reaction, PCR) ist ein automatisiertes Verfahren zur Vervielfältigung von DNA-Sequenzen. Die drei Hauptverarbeitungsschritte werden wiederholt ausgeführt, wobei sich in jeder Iteration die Menge der DNA-Sequenzen verdoppelt: 1.) Erhitzen, um die DNA-Doppelstränge zu trennen, 2.) Abkühlen, um kurzen Anfangsstücken, den Primern, die Anlagerung an die Einzelstränge zu ermöglichen, 3.) Anlagerung weiterer Nukleotide an die Anfangssequenz und Verkettung durch DNA-Polymerase.

können eigentlich zusammenhangslose Sequenzen erhalten bleiben. Ebenso wie der stochastische Prozess der Sequenzerzeugung ist die Kürze der untersuchte Sequenzen und ihr immer gleicher Kontext (die Primersequenz) Anlass, die Übertragbarkeit der Ergebnisse auf genomische Sequenzen in Lebewesen anzuzweifeln. Zuletzt sei angemerkt, dass die Waschung und die anschließende Vervielfältigung der ausgewaschenen Sequenzen selbst ein selektiver Prozess ist, der mit der untersuchten Eigenschaft nichts zu tun hat. Schlimmer noch könnten gerade Sequenzen, die die zu selektierende Eigenschaft besitzen bei der Waschung und Vervielfältigung benachteiligt sein. Beispielsweise sind RNA-Moleküle, die bestimmte Sekundärstrukturen ausbilden, bei der Gelelektrophorese benachteiligt, obwohl sie hinsichtlich der Bindung eines Splicingfaktors die interessantesten Objekte wären [Klu94].

2.3.5 ChIP on Chip

ChIP on chip ist die Bezeichnung für ein modernes Verfahren, dass in der kombinierten Anwendung von *Chromatin-Immunopräzipitation* und *Microarray*-Analysen besteht. Die Kombination ermöglicht es, für ein gesamtes Genom jene Regionen zu bestimmen, die von einem bestimmten Transkriptionsfaktor gebunden werden. Das wurde im Jahr 2000 von Ren et al. vorgestellt [Ren00]. Seit dem hat es sich zu einem der wichtigsten Experimentalmethoden in der Erforschung der Genregulation entwickelt. Zur Beschreibung eines *ChIP on Chip*-Experiments müssen zunächst die beiden Komponenten erläutert werden, die ihrerseits große Bedeutung für die Erforschung der transkriptionellen Genregulation haben.

Microarrays. Microarray ist eine Sammelbezeichnung für moderne molekularbiologische Untersuchungssysteme, die die parallele Analyse von mehreren tausend Einzelnachweisen in einer geringen Menge biologischen Probenmaterials erlauben. Es gibt verschiedene Formen von Microarrays [Mü04]. Für Experimente hinsichtlich der Genregulation sind besonders DNA-Microarrays interessant. Mit ihnen können RNA-Mengen eines Zellextrakts für tausende Gene gleichzeitig nachgewiesen werden, die Rückschlüsse auf die jeweilige Genaktivität zulassen. Dabei handelt es sich um einen Glasträger, auf dem an fest definierten Rasterpositionen entweder eindeutige Oligonukleotide oder cDNA[13] platziert werden. Diese so genannten *chips* werden industriell gefertigt. Zur Durchführung eines Microarray-Experiments wird zunächst ein Zellsubstrat aus dem zu untersuchenden Objekt entnommen. Aus diesem werden alle mRNA-Moleküle isoliert, die durch *Reverse Transkriptase* in cDNA zurück kodiert werden. Die cDNA-Moleküle werden je nach Probe unterschiedlich mit einem Fluoreszenzfarbstoff markiert und auf den Chip aufgetragen. Die RNA-Moleküle lagern sich an den passenden DNA-Molekülen an. Nach Abwaschen der nicht gebundenen cDNA-Stücke wird ein Fluoreszenzsignal jeder Position des DNA-Microarrays mittels eines Lasers ausgelesen. Die Intensität eines Lichtflecks korreliert mit der Aktivität des Gens im gesamten Zellextrakt. Die Farbe des Lichtflecks

[13] cdna???

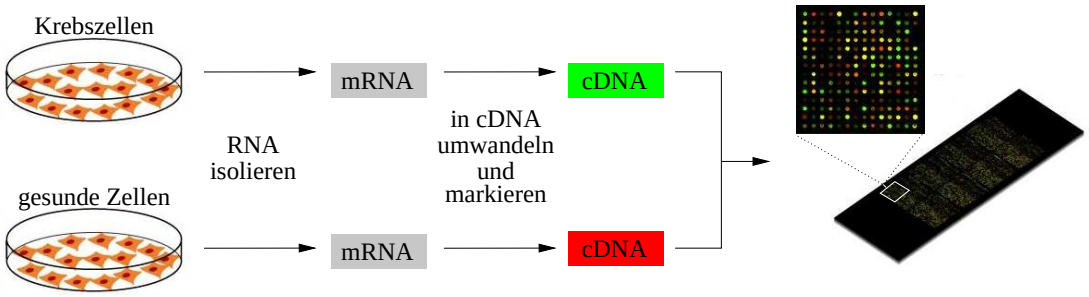


Abbildung 2.12: Ein typisches Microarray-Experiment. Aus Krebszellen und gesunden Zellen werden RNA-Moleküle isoliert, cDNA-Sequenzen hergestellt und diese unterschiedlich farbig markiert. Gene, die hauptsächlich in Krebszellen exprimiert wurden, erscheinen im Microarray grün, die nicht differentiell exprimierten gelb, und die Gene, die nur in der gesunden Zelle aktiv waren, rot.

korreliert mit der Aktivität in einer bestimmten Probe. Eine typische Versuchsabfolge ist es, Zellproben verschiedener Phenotypen[14] zu verschiedenen Zeitpunkten zu messen und differentiell exprimierte Gene zu identifizieren (siehe Abbildung 2.12). Die Verarbeitung von Microarraydaten ist sehr schwierig, da Experimente verschiedener Chips nicht so einfach miteinander verglichen werden können, sondern vorher in komplexen Verfahren normalisiert werden müssen. Auch die Identifikation der differentiell exprimierten Gene ist eine heikle Aufgabe, da häufig nur wenige Proben zu wenigen Zeitpunkten vorliegen. Mit der Analyse von Microarraydaten befasst sich deshalb ein ganzer Zweig der Bioinformatik.

Chromatin-Immunopräzipitation. Das Grundprinzip der *Chromatin-Immunopräzipitation* (ChIP) beruht darauf, die zu einem Zeitpunkt bestehenden Protein-DNA-Bindungen durch Fixierung mit Formaldehyd festzuhalten. Anschließend werden die Zellen zerstört und das Chromatin mittels Ultraschall in Stücke einer Länge von einigen hundert Basenpaaren Länge zertrümmert. Jene DNA-Stücke, die das gewünschte Protein gebunden haben, werden mit einem für das Protein spezifischem Antikörper immunopräzipiert. Die isolierten DNA-Protein-Komplexe werden gelöst. Die Identität der jeweiligen DNA-Stücke kann auf verschiedene Weise geklärt werden: Besteht eine Hypothese über den wahrscheinlichen Bindungsort im Genom, kann eine PCR unter Verwendung von Primern, die spezifisch für die vermutete DNA-Region sind, durchgeführt werden. Da das Verfahren in vivo arbeitet, unterscheidet es sich wesentlich von anderen Verfahren, die das gleiche Ziel verfolgen, wie z.B. das *DNAase Footprinting Assay*. Das größte Problem einer ChIP-Untersuchung ist es, tatsächlich spezifische Antikörper für das untersuchte Protein zur Verfügung zu haben.

[14] z.B. kranker und gesunder Proband

ChIP on Chip. Bei einer *ChIP on Chip*-Untersuchung werden die DNA-Fragmente, die in einem ChIP-Experiment das interessierende Protein gebunden haben, vervielfältigt. Beide DNA-Fraktionen, d.h. der Teil, der das Protein nicht gebunden hat und der Teil, der es gebunden hat, werden auf je ein Microarray aufgetragen, das alle genomischen, nichtkodierenden DNA-Sequenzen zwischen zwei Genen enthält. Durch Vergleich der Signale beider Proben und durch Mehrfachuntersuchungen kann die relative Stärke der Bindung des untersuchten Proteins an jeder einzelnen Promotorregion bestimmt werden.

Der schnellen Bestimmung von Bindungsregionen und vermeintlich regulierten Genen eines Transkriptionsfaktors stehen Fehlerquellen gegenüber, die bei der Interpretation der Daten berücksichtigt werden müssen [Jol05]. Zum einen setzt eine lückenlose Identifizierung aller Zielgebiete eine ausreichend hohe Konzentration des Faktors in der Zelle voraus. Das ist ein Umstand, der im Falle der aktuellen Inaktivität des Faktors nicht gewährleistet ist. Bindet der Transkriptionsfaktor nur gemeinsam mit weiteren Proteinen, dann müssen diese ebenfalls in ausreichend hoher Konzentration zur Verfügung stehen. Anderenfalls würde das Experiment lediglich eine Teilmenge aller möglichen Bindungsregionen ausgeben. Zudem besteht begründeter Verdacht, dass die durch die Messung veränderte Physiologie der Zelle das Bindungsverhalten einiger Faktoren verändert. Beim Verwenden der Daten zum Erstellen regulatorischer Netzwerke muss berücksichtigt werden, dass eine Bindung des Proteins nicht zwangsweise eine regulative Funktion zur Folge hat. Diese könnte erst durch das gleichzeitige Vorhandensein kooperierender Faktoren erfüllt werden. Aufgrund dessen sind in jüngster Zeit häufig Untersuchungen darauf gerichtet, Schnittmengen von Sequenzen zu ermitteln, die in verschiedenen *ChIP on chip*-Experimenten von jedem einzelnen der kooperierenden Faktoren gebunden werden [Boy05].

Die genomweite Identifizierung der durch einen TF gebundenen Promotorsequenzen liefert wertvolle Daten, um Regulationszusammenhänge zwischen dem Faktor und Genen abzuleiten, etwa für die Konstruktion regulativer Netzwerke. Die Ergebnisse eines *ChIP on Chip*-Experimentes lassen sich zudem zur Vorhersage der punktgenauen Bindungssequenzen in den Fragmenten verwenden. Aufgrund der Länge der ausgegebenen DNA-Fragmente von mehreren Hundert Basenpaaren stellen *ChIP on Chip*-Daten eine im Vergleich zu SELEX-Daten große Herausforderung für die in Abschnitt 3.2 vorgestellten Verfahren zur Motivsuche dar.

Kapitel 3

Algorithmische Ansätze zur Modellierung regulativer DNA-Sequenzen

Die im Abschnitt 2.3 des vorherigen Kapitel vorgestellten experimentellen Verfahren zur Untersuchung transkriptioneller Regulation erfordern enorme zeitliche und finanzielle Ressourcen. Ein *DNA mobility shift assay* ermöglicht die Identifizierung jener Transkriptionsfaktoren, die einen bestimmten Promotor binden. Ohne stichhaltige Hinweise, die eine Eingrenzung der möglichen Proteine auf eine sehr kleine Menge rechtfertigen, ist dieses Verfahren jedoch kaum praktikabel. Ebenso verhält es sich bei der positionsgenauen Bestimmung der Bindungsstellen eines Transkriptionsfaktors innerhalb eines Genoms mittels einer kombinierten Anwendung von *ChIP on chip*-Versuchen und *DNaseI footprinting assays*. Auch in diesem Fall scheinen Hilfsmittel zur Reduzierung des Experimentieraufwands unerlässlich.

Schon sehr früh entwickelte sich aufgrund dieser Bedarfslage eine eigene Forschungsrichtung innerhalb der Bioinformatik, die sich mit algorithmischen Methoden zur Vorhersage von regulatorischen DNA-Sequenzen beschäftigt. Dieses Kapitel gibt einen methodischen Überblick jener Teilgebiete, mit denen sich diese Dissertation beschäftigt.

Ein wichtiger Aspekt ist hierbei die Modellierung der Bindungssequenzen (TFBS) eines Transkriptionsfaktors. Von einfachen, zeichenkettenbasierten Sequenzmotiven bis hin zu komplexen Modellen der DNA-Protein-Bindung wurden in den vergangenen 20 Jahren eine Vielzahl von Arbeiten veröffentlicht. Abschnitt 3.1 stellt die wichtigsten Vertreter vor. Abschnitt 3.2 widmet sich der Extraktion von TFBS-Modellen aus vorhandenen Sequenzdaten, vornehmlich in dem schwierigen und interessanten Fall, dass die Lerndaten keine alignierten TFBS eines untersuchten Transkriptionsfaktors enthalten. Wie bereits aus Abschnitt 2.2 auf Seite 19 hervorging, binden und agieren verschiedene Transkriptionsfaktoren häufig gemeinsam auf *Enhancer*- oder *Silencer*-Sequenzen. Die integrierte Modellierung von Gruppen zusammengehöriger TFBS ist eine Möglichkeit, eine erhöhte Vorhersagekraft bei der Erkennung regulativer Sequenzen zu erreichen. Die diesbezüglich wichtigsten Ansätze werden in Abschnitt 3.3 vorgestellt.

3.1 Repräsentation von Transkriptionsfaktorbindungsstellen

Es gilt als bewiesene Tatsache, dass Transkriptionsfaktoren bevorzugt an für sie jeweils charakteristischen DNA-Abschnitten binden. Beim Vergleich experimentell bestimmter Bindungssequenzen eines Faktors fallen besonders im Bereich des direkten Kontakts mit dem Faktor die hohen Sequenzähnlichkeiten auf. Die frühesten Ansätze zur TFBS-Modellierung basieren aus diesem Grund auf zeichenkettenbasierten *Sequenzmotiven*, die im begrenzten Maße die Variabilität zwischen den einzelnen TFBS abbilden können. Unterabschnitt 3.1.1 stellt die wichtigsten Sequenzmotivklassen vor. Eine Weiterentwicklung stellt die *Positionsgewichtsmatrix* (PWM) dar, die eine quantitative Beurteilung der Übereinstimmung einer Sequenz mit der optimalen TFBS ermöglicht.

Sowohl Sequenzmuster als auch Gewichtsmatrizen werden noch immer vielfach angewendet. Gleichwohl hat sich bereits in ihrer Anfangszeit eine gewisse Ernüchterung über die Vorhersageleistungen beider Modellklassen eingestellt. Besonders die hohen Raten von Falsch-Positiven (Fehler erster Art) lassen eine isolierte Anwendung dieser Modelle zur Eingrenzung vielversprechender experimenteller Versuchsläufe nicht lohnend erscheinen. Eine mögliche Erklärung für die hohen Klassifikationsfehlerraten ist, dass die ausgenutzten Sequenzähnlichkeiten zwar für das menschliche Auge offensichtlich sind, möglicherweise aber nur Konsequenz unbekannter, komplexerer Erfordernisse an die Bindungsstellen sind. Möglicherweise charakterisiert die Nukleotidfolge einer TFBS diese verborgenen Eigenschaften nur unzureichend. Das Finden und die direkte Berücksichtigung der versteckten Eigenschaften verspreche dann eine verbesserte Klassifikation.

Die in Kapitel 5 vorgestellte Arbeit zielt darauf ab, einen Modellierungsansatz für TFBS zu entwickeln, der die Vorzüge von Gewichtsmatrizen aufgreift, zudem aber eine genauere und flexiblere Beschreibung der wesentlichen biologischen Eigenschaften von TFBS ermöglicht. Aufgrund dieses "Wettbewerbs" mit den Gewichtsmatrizen ist ihnen ein eigener Unterabschnitt 3.1.2 gewidmet. Das Ziel, die mäßige Erkennungsleistung einer Gewichtsmatrix durch komplexere Modelle zu übertreffen, wurde von einigen Arbeitsgruppen verfolgt, deren Arbeiten in Unterabschnitt 3.1.3 kurz vorgestellt werden.

3.1.1 Zeichenkettenbasierte Sequenzmotive

Die Systematik der verschiedenen Klassen von Sequenzmotiven orientiert sich an der Veröffentlichung [Bra98] von Brazma et al., beschränkt sich jedoch auf DNA-Sequenzen. Demnach wird von einem Basisalphabet $\Sigma_{DNA} = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$ ausgegangen. Weiterhin sei ein Alphabet $\Sigma_{IUPAC} = \{\mathtt{R}, \mathtt{Y}, \mathtt{M}, \mathtt{K}, \mathtt{W}, \mathtt{S}, \mathtt{B}, \mathtt{D}, \mathtt{H}, \mathtt{V}, \mathtt{N}\}$ gegeben, dessen Zeichen für bestimmte Teilmengen von Σ_{DNA} stehen (siehe Tabelle 3.1). Fasst man die Symbole des DNA-Alphabets auch als Einermengen der jeweiligen Nukleotide auf, handelt es sich bei der Vereinigung $\Sigma_{DNA} \cup \Sigma_{IUPAC}$ um den IUPAC-Code für Nukleotidsequenzen[1].

[1] IUPAC: *International Union of Pure and Applied Chemistry* maßgebliche Institution in Bereichen der Nomenklatur chemischer Stoffe.

Zeichen	Teilmenge von Σ_{DNA}
R	A, G (Purinbasen)
Y	C, T (Pyrimidinbasen)
M	C, A
S	C, G
K	T, G
W	T, A
B	C, T, G (nicht A)
D	A, T, G (nicht C)
H	A, T, C (nicht G)
V	A, C, G (nicht T)
N	A, C, G, T (beliebig)

Tabelle 3.1: Der IUPAC-Code für DNA-Sequenzen

Besondere Beachtung verdient das Zeichen $\mathtt{N} \in \Sigma_{IUPAC}$, da es als Platzhalter für ein beliebiges Nukleotid $s \in \Sigma_{DNA}$ dient. Für zwei beliebige natürliche Zahlen p und q mit $p \leq q$ bezeichne $\mathtt{N}\ (p, q)$ die Menge aller Zeichenketten $\boldsymbol{s}$ über Σ_{DNA}, deren Länge zwischen p und q liegt. Die von diesem Ausdruck erzeugte formale Sprache ist demnach $\mathcal{L}(\mathtt{N}(p,q)) = \{\boldsymbol{s} \in \Sigma^*_{DNA} \,|\, p \leq |\boldsymbol{s}| \leq q\}$. Sei weiterhin $\Sigma_{\mathtt{N}}$ das (unendliche) Alphabet aller möglichen $\mathtt{N}\ (p, q)$.

DEFINITION 3.1: *Ein* **zeichenkettenbasiertes Sequenzmotiv** *ist die formale Sprache* $\mathcal{L}(\boldsymbol{A})$ *einer Zeichenkette* $\boldsymbol{A}$ *über dem Alphabet*

$$(\Sigma_{DNA} \cup \Sigma_{IUPAC} \cup \Sigma_{\mathtt{N}}). \tag{3.1}$$

Es setzt sich aus den formalen Sprachen der einzelnen Zeichen A_i *der Zeichenkette zusammen:*

$$\begin{aligned} \mathcal{L}(\boldsymbol{A}) &= \mathcal{L}(A_1)\mathcal{L}(A_2)\cdots\mathcal{L}(A_W) && (3.2)\\ &= \{\boldsymbol{s}_1 \cdots \boldsymbol{s}_W | \forall_{i=1\ldots W}\, \boldsymbol{s}_i \in \Sigma^*_{DNA} \wedge \boldsymbol{s}_i \in \mathcal{L}(A_i)\}. && (3.3) \end{aligned}$$

Die durch ein Zeichen $A \in \Sigma_{IUPAC}$ definierte Sprache besteht trivialerweise aus den einsymboligen Zeichenketten der korrespondierenden Nukleotidteilmenge, wie in Tabelle 3.1 angegeben. Analoges gilt auch für die Nukleotide $s \in \Sigma_{DNA}$. Eine Sequenz $\boldsymbol{s} \in \Sigma_{DNA}$ heißt *Treffer* für ein Sequenzmuster $\boldsymbol{A}$, wenn gilt: $\boldsymbol{s} \in \mathcal{L}(A)$.

Brazma et al. unterscheiden verschiedene Klassen von Sequenzmotiven. Die Motivklassen sind gemäß ihrer Beschreibungsmacht sortiert. Die Klasse der niedrigsten Stufe enthält Sequenzmotive, die ausschließlich aus Nukleotidzeichen bestehen. Die Sprache eines solchen Motivs besteht selbstverständlich nur aus dem Muster selbst. Die Motivklasse der

sechsten Stufe gestattet eine variable Anzahl beliebiger Zeichen zwischen konservierten Zeichen. Diese Klasse wird im Zusammenhang mit der Modellierung von Proteinfamilien in der Datenbank PROSITE verwendet (natürlich mit Aminosäurenalphabet).

Zur Beschreibung der Bindungsstellen eines Transkriptionsfaktors haben sich vor allem Sequenzmotive der dritten Stufe etabliert. Motive dieser Klasse heißen *Consensus*sequenzen und sind Wörter über dem Alphabet $\boldsymbol{A} \in \Sigma_{DNA} \cup \Sigma_{IUPAC}$. Demnach haben alle Elemente der Sprache einer Consensussequenz die gleiche Länge. Die Treffer der Consensussequenz `ACRY` sind beispielsweise die Wörter `ACAC`, `ACAT`, `ACGC` und `ACGT`.

Consensus-Sequenzen können leicht aus einem lückenlosen Alignment bekannter TFBS gebildet werden. Die trivialste Strategie, sich an jeder Position i für das Zeichen $A_i \in \Sigma_{IUPAC}$ zu entscheiden, welches alle in der Alignment-Spalte vorkommenden Nukleotide vertritt, führt jedoch selbst bei hoch konservierten Daten nicht zum Erfolg. Die formalen Sprachen der resultierenden Consensussequenzen wären zu groß, da an den Positionen zu viele Freiheiten gewährt würden. Bei der Anwendung des Sequenzmotivs zur Suche von möglichen TFBS in langen Sequenzen ergäben sich bei Weiten zu viele Treffer. Konstruktionsstrategien, die stringentere Consensussequenzen erzeugen, nehmen in Kauf, dass nicht alle Sequenzen der Trainingsmenge auch Treffer der Consensus-Sequenz sind. In [Day92] werden die Consensussequenzen verschiedener Strategien hinsichtlich ihrer Eignung zur trennscharfen Vorhersage miteinander verglichen. Bis heute hat die Methode von Cavener [Cav87] eine große Bedeutung, da sie zur Berechnung der Consensussequenzen in der Datenbank TRANSFAC (siehe Unterabschnitt 3.1.4) verwendet wird. Ausgehend von den relativen Häufigkeiten $p_{\mathtt{A}i}$, $p_{\mathtt{C}i}$, $p_{\mathtt{G}i}$ und $p_{\mathtt{T}i}$ der Nukleotide in einer Alignment-Spalte i, werden zur Bestimmung des betreffenden Consensuszeichens folgende vier Regeln der Reihe nach überprüft.

1. Kommt ein Nukleotid $s \in \Sigma_{DNA}$ mit einer Häufigkeit $p_{si} > 0.5$ in Spalte i vor, und ist dies bereits mehr als doppelt so häufig wie das zweit-häufigste Nukleotid, dann wähle $A_i = s$.
2. Kommen zwei Nukleotide s und r zusammen mit einer Häufigkeit $p_{ri} + p_{si} > 0.75$ und trifft Regel 1 nicht zu, dann wähle $A_i \in \Sigma_{IUPAC}$ genau so, dass die korrespondierende Nukleotidmenge genau und ausschließlich s und r enthält.
3. Kommt ein Nukleotid s in der Alignment-Spalte gar nicht vor, und können die Regeln 1 und 2 nicht angewendet werden, so wähle $A_i \in \Sigma_{IUPAC}$ so, dass die korrespondierende Nukleotidmenge $\Sigma_{DNA} \setminus \{s\}$ entspricht.
4. Können die ersten drei Regeln nicht angewendet werden, so wähle $A_i =$`N`.

Abbildung 3.1 beschreibt die Bildung einer Consensus-Sequenz nach dieser Strategie.

Consensussequenzen zeigen trotz ihrer einfachen Struktur eine erstaunlich gute Eignung für die Beschreibung von TFBS. Das trifft vor allem in den häufigen Fällen zu, in denen alle bekannten Bindungssequenzen die gleiche Länge besitzen und sich leicht ein lückenloses und starkes Alignment erstellen lässt. Consensussequenzen wurden in den letzten Jahren größtenteils durch Positionsgewichtsmatrizen abgelöst, spielen aber bei

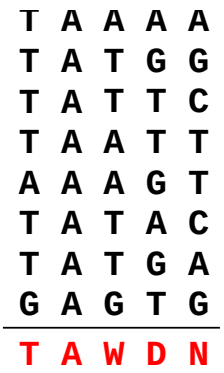


Abbildung 3.1: Konstruktion einer Consensussequenz aus einem lückenlosen Alignment nach [Cav87]. Für die ersten beiden Spalten wird Regel 1 angewendet, für die dritte Spalte Regel 2, für die vierte Regel 3 und für die letzte Spalte Regel 4.

der Visualisierung von gemeinsamen Eigenschaften von TFBS eines Faktors, etwa in Datenbanken oder Veröffentlichungen, eine große Rolle. In Kapitel 5 werden unter Anderem die relativen Startpositionen der Treffer kurzer Consensussequenzen als Eigenschaften von TFBS modelliert. Consensussequenzen und weitere zeichenkettenbasierten Sequenzmotivklassen wurden in diversen Verfahren zur Motivsuche eingesetzt. Im Einzelfall wird in Abschnitt 3.2, Unterabschnitt 3.2.1 näher darauf eingegangen.

3.1.2 Positionsgewichtsmatrizen

Eine natürliche Weiterentwicklung der Consensussequenzen sind *Positionsgewichtsmatrizen* (Abkürzung PWM gemäß der englischen Bezeichnung *Position Weight Matrix*[2]). Der Fortschritt besteht in einer Gewichtung der alternativen Zeichen an den einzelnen Positionen. Eine Sequenz erhält durch eine PWM eine *Bewertung*, die ausdrückt, in welchem Maße sie dem durch die PWM beschriebenen Sequenzmotiv ähnelt. Im Gegensatz zu textuellen Sequenzmotivbeschreibungen, die für jede Sequenz eindeutig entscheiden können, ob diese eine gültige Instanz des Sequenzmotivs ist, obliegt es dem Anwender einer PWM, über eine Bewertungsschranke eine solche Entscheidungsregel zu definieren. Für die Wahl einer günstigen Schranke werden häufig Methoden der klassischen Statistik eingesetzt (siehe Seite 40).

DEFINITION 3.2: *Sei Σ ein endliches Alphabet geordneter Symbole $x_1, \ldots, x_T$. Eine* **PWM** *der Länge W ist eine Matrix $M \in \mathbb{R}^{T \times W}$. Ein Matrixeintrag m_{ij} heißt* **Gewicht** *des Symbols x_i an Position j.*

[2] Neben "PWM" findet sich gerade in der jüngeren Fachliteratur häufig die Bezeichnung PSSM (*Position Specific Score Matrix*)

Eine PWM weist einer beliebigen Zeichenkette $s_1 s_2 \ldots s_W$ *mit* $s_i \in \Sigma$ *eine Bewertung vermöge*

$$S(s_1 s_2 \cdots s_W) \quad = \quad \sum_{j=1}^{W} m_{i(s_j)j} \tag{3.4}$$

zu, wobei $i(s_j)$ *den Alphabetindex des Zeichens* s_j *bezeichnet.*

Das den PWM inne wohnende, additive Bewertungsschema impliziert die paarweise Unabhängigkeit zwischen den Spalten, bzw. zwischen den Positionen des Sequenzmotivs. Der Beitrag von Spalte j zur Bewertung $S(s_1 s_2 \cdots s_W)$ hängt ausschließlich vom Zeichen s_j ab. Im Rahmen dieser Arbeit interessiert ausnahmslos das Alphabet Σ_{DNA} der DNA-Nukleotide. *PWM* wird im Weiteren gleichbedeutend für Gewichtsmatrizen dieses Alphabets verwendet. Die Reihenfolge der Symbole und damit der 4 Zeilen einer PWM erfolgt gemäß der lexikographischen Ordnung. Abbildung 3.2 zeigt eine PWM samt dem Sequenzalignments, aus dem sie konstruiert wurde.

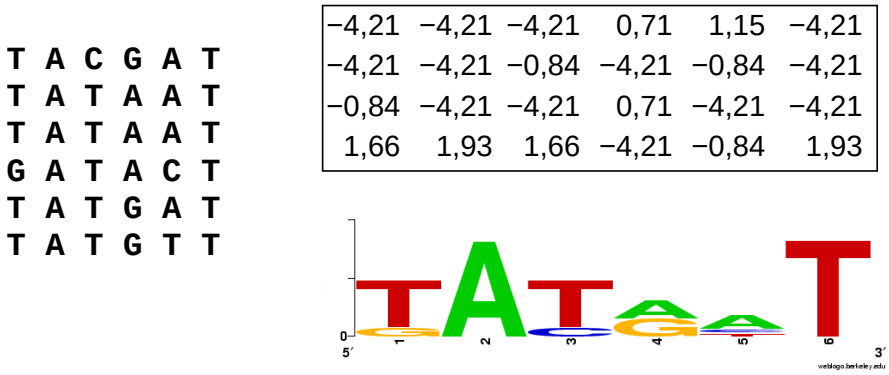


Abbildung 3.2: Konstruktion einer PWM (rechts oben) aus einem lückenlosen Alignment (links). Die Gewichte entsprechen den Beiträgen einzelner Zeichen zum Informationsgehalt einer Spalte, d.h. den *log-odds*-Bewertungen. Unter der PWM ist eine Visualisierung der PWM nach [Sch90] abgebildet. Die Buchstabengröße korreliert mit der Bedeutung eines Nukleotids in einer Spalte.

Berechnung der Gewichte. Eine wesentliche Frage bei der Beschäftigung mit PWM ist die der Berechnung der Gewichte m_{ij}. Seit dem Zeitpunkt der ersten Veröffentlichung des PWM-Ansatzes im Jahre 1982 durch Gary D. Stormo [Sto82] wurden hierfür eine Reihe von Vorschlägen gemacht. Diese erste Arbeit, die sich mit Startsequenzen der Translation innerhalb von mRNA-Sequenzen beschäftigt, verwendet Methoden aus dem Bereich der neuronalen Netze. Ausgehend von einer Stichprobe bekannter Translationsstartpunkte und einer Negativstichprobe beliebiger anderer RNA-Sequenzen sollte eine Matrix bestimmt werden, die diese beiden Stichproben möglichst gut unterscheiden kann.

Um dies zu erreichen, wurde ein Perzeptron[3] mit Hilfe der beiden Stichproben trainiert. Die gelernten Perzeptrongewichte wurden als Gewichte der PWM übernommen.

Die naheliegende Herangehensweise, eine verfügbare TFBS-Stichprobe zu verwenden, um relative Häufigkeiten $p_{\mathtt{A}j}$, $p_{\mathtt{C}j}$, $p_{\mathtt{G}j}$ und $p_{\mathtt{T}j}$ der vier Nukleotide an den Positionen j zu bestimmen, und diese als Gewichte zu verwenden wurde zwei Jahr darauf von Staden veröffentlicht [Sta84]. Anstelle der relativen Häufigkeiten werden ihre negativen, natürlichen Logarithmen verwendet. Auf diese Weise wird die PWM zu einem *probabilistischen Sequenzmodell* und die Bewertung $S(s_1 s_2 \cdots s_W)$ ist die negative logarithmierte Wahrscheinlichkeit, mit der die Sequenz $s_1 s_2 \cdots s_W$ von dem PWM-Modell erzeugt wird.

$$S(s_1 s_2 \cdots s_W) = \sum_{j=1}^{W} -\ln p_{s_j j} \tag{3.5}$$

$$= -\ln \prod_{j=1}^{W} p_{s_j j}. \tag{3.6}$$

Als Sequenzmodell beschreibt die PWM dann die gemeinsame Verteilung von Sequenzen der Länge W, die alle gültige TFBS des modellierten Typs sind. Die angewendete Produktregel ist nur dadurch zulässig, dass die paarweise statistische Unabhängigkeit von gemäß den Spaltenverteilungen $p_{cdot\,j}$ verteilten Zufallsvariablen S_j angenommen wird. Für die Verwendung der logarithmischen Werte ist es erforderlich, dass in den Spaltenverteilungen keine Nullwahrscheinlichkeiten auftreten. Dies wird beispielsweise durch Addition aller Zählstatistiken um einen kleinen Wert erreicht, was einer Glättung der Spaltenverteilungen entspricht. Die probabilistische Version des PWM entspricht einem speziellen *Hidden Markov Model* (HMM) (siehe Kapitel 5, Abschnitt 5.5) , dessen verborgene Zustände in strikter Reihenfolge mit Übergangswahrscheinlichkeit 1.0 hintereinander eingenommen werden und als Ausgabeverteilungen jeweils die Spaltengewichte des PWM besitzen. Derart berechnet, lassen sich PWM in zusammengesetzte stochastische Modelle integrieren, etwa zur Vorhersage von TFBS-Modulen (siehe dazu Abschnitt 3.3 oder Abschnitt 5.5).

Die Stärke des durch eine PWM beschriebenen Sequenzmotivs, d.h. der Grad der Abweichung von durchschnittlichen Sequenzen, wird durch die Summe der *relative Entropien* (auch *Kullback-Leibler-Abstand*) zwischen den Basenverteilungen der einzelnen Spalten und der im Genom auftretenden Verteilung[4] $\{q_s\}$ für $s \in \Sigma_{DNA}$ der Nukleotide angege-

[3] Ein Perzeptron ist ein einfach strukturiertes neuronales Netz. Seine Neuronen sind in einer oder mehreren Schichten (*multi-layer perceptron* – MLP)angeordnet. Das Perzeptron erzeugt bei Anlegen eines Eingabevektors an die Eingabeschicht gemäß seiner Gewichte einen Ausgabevektor. Die Dimensionen dieser Vektoren entsprechen den Anzahlen der Neuronen in der Ein- und Ausgabeschicht. Es existieren sowohl für einlagige als auch für mehrlagige Perzeptrons effektive, überwachte Lernverfahren (siehe [Min88].

[4] Anstatt einer genomweiten Nukleotidverteilung werden meist lokale Nukleotidverteilungen des Typs von Sequenzen, die durchsucht werden sollen (z.B. Promotoren), verwendet

ben. Dieser Wert wird *Informationsgehalt* I_{PWM} der PWM genannt [Sch98]

$$I_{PWM} = \sum_{j=1}^{W} \mathcal{H}(j) \tag{3.7}$$

$$= \sum_{j=1}^{W} \sum_{s \in \Sigma_{DNA}} p_{sj} \log_2 \frac{p_{sj}}{q_s}. \tag{3.8}$$

Durch Verwenden des Logarithmus zur Basis 2 ergibt sich als Einheit des Informationsgehalts das *Bit*. Im Spezialfall, dass die Nukleotide im gesamten Genom gleichverteilt sind, $q_s = 0.25$ für alle $s \in \Sigma_{DNA}$, kann der Beitrag einer Spalte zwischen 0 Bit (die p_{sj} sind ebenfalls gleichverteilt) und 2 Bit (für ein s gilt $p_{sj} = 1.0$) liegen. Die relative Entropie einer Spalte ist der Erwartungswert der Ausdrücke $\log_2 \frac{p_{sj}}{q_s}$. Diese Ausdrücke heißen *log-odds*-Bewertungen (englisch: *log-odds-scores*) und werden in vielen Fällen selbst als Gewichte des PWM verwendet. Sie werden außerdem eingesetzt, um eine intuitive Visualisierung einer PWM zu ermöglichen (siehe Abbildung 3.2 und [Sch90]).

Eine weitere Art von PWM-Gewichten erfordert im Lernprozess zusätzlich experimentell bestimmte, quantitative Daten über die Bindungsaffinitäten der TFBS aus der Lernstichprobe. Diese Affinitäten verringern sich mit jeder Abweichung von der optimalen Bindungssequenz eines Transkriptionsfaktors. Eine optimale TFBS besitzt minimale Bindungsenergie. Die Gewichte der PWM werden mittels eines Optimierungsverfahrens so gewählt, dass sie möglichst gut die quantitativen Daten erklären. Das bedeutet, dass Sequenzen mit niedriger Bindungsenergie eine entsprechend hohe Bewertung erhalten [Sto98]. Auch hier versteckt sich eine tiefgreifende Unabhängigkeitsannahme. Aufgrund des additiven Bewertungsschemas wird voraus gesetzt, dass einzelne Spalten unabhängig von anderen Spalten einen definierten Beitrag zur gesamten Bindungsenergie der Bindungssequenz liefern. Diese und weitere Modellierungsannahmen vorausgesetzt kann gezeigt werden, dass die negativen *log-odds*-Bewertungen unter allen PWM-Gewichten jene sind, welche die Wahrscheinlichkeit, dass jede der Stichproben-TFBS auch tatsächlich durch den TF gebunden wird, maximieren [Sto98].

Statistische Signifikanz. Für die Vorhersage von TFBS in einer Eingabesequenz mit einer entsprechend trainierten PWM wird zunächst für jedes mögliche Sequenzfenster $s_i \cdots s_{i+W-1}$ die Bewertung $S(s_i \cdots s_{i+W-1})$ bestimmt. Die Bewertung einer Position ist Grundlage für die Klassifikation der dazugehörigen Teilsequenz in eine der beiden Klassen *TFBS* oder *nicht-TFBS*. Die Entscheidungsregel ist durch eine Bewertungsschranke S_{TFBS} definiert. Teilsequenzen, die eine größere Bewertung erreichen, werden als TFBS klassifiziert.

Die Wahl einer günstigen Schranke S_{TFBS} ist eine folgenreiche Entscheidung bei der Verwendung von PWM. Es stellt sich als Abwägen zwischen den beiden konträren Zielen einer möglichst hohen *Sensitivität* (Anteil richtiger Vorhersagen an allen Vorhersagen) und einer möglichst hohen *Spezifität* (Anteil der richtig als Nichttreffer klassifizierten

Beispiele unter allen Negativklassifikationen) dar. Eine Schranke, mit der zumindest alle TFBS, die zum Erlernen des PWM verwendet wurden, als Treffer klassifiziert werden, ist schon aufgrund eines stärker abweichenden Stichprobenelements so niedrig, dass die Anzahl von TFBS-Vorhersagen völlig unhaltbare Dimensionen annimmt. Um dies zu verhindern, muss die Nichterkennung der schlechteren aber experimentell bestätigten TFBS in Kauf genommen werden. Eine zu hohe Schranke trägt die Gefahr einer Überanpassung des PWM-basierten Klassifikators in sich.

Da es das größte Problem von PWM ist, dass sie leicht um Größenordnungen mehr Treffer liefern, als unter biologischen Gesichtspunkten erwartet werden könnten, haben sich eine Reihe von Arbeiten mit der sinnvollen Wahl einer Bewertungsschranke beschäftigt. Vielfach wurden dabei Verfahren aus der klassischen Statistik eingesetzt. Dazu wird eine Verteilung der PWM-Bewertungen angenommen, und bezüglich dieser die statistische Signifikanz, ausgedrückt durch den p-Wert, einer neu gemessenen PWM-Bewertung bestimmt. Der p-Wert einer PWM-Bewertung S ist die Wahrscheinlichkeit dafür, mit der PWM, angewendet auf eine zufällige Sequenz, eine höhere oder gleiche Bewertung als S zu erreichen.

Trivialerweise könnte der p-Wert einer Bewertung S exakt bestimmt werden, indem alle Sequenzen der PWM-Länge aufgezählt werden und ausgezählt wird, wie häufig eine Sequenz eine bessere Bewertung als S bekam. Dafür ist jedoch die Bewertung von $\mathcal{O}(4^W)$ Sequenzen nötig, für längere PWM-Modelle nicht tragbar. Unter Umständen genügt es, eine bestimmte Anzahl Sequenzen der Länge W zufällige Sequenzen von einem Sequenzmodell (z.B. einer einfachen Markovkette) erzeugen zu lassen, dass durchschnittliche Sequenzen modelliert. Anhand der Bewertungen dieser zufälligen Sequenzen wird eine empirische, kumulative Statistik aufgebaut, von der eine Näherung des exakten p-Wertes abgelesen werden kann. Laut Barash et al. [Bar04] werden zur verlässlichen Bestimmung von p-Werten der Größenordnung 10^{-T} mindestens 10^{T+2} zufällige Sequenzen benötigt. Das liegt daran, dass besonders hohe Bewertungen in zufälligen Sequenzen selten vorkommen und deshalb die empirische Verteilung in diesem Bereich besonders schlecht abgedeckt ist. Seine CIS-Methode (für englisch *Compound Importance Sampling*), bei dem die Forscher anstatt eines Hintergrund-Sequenzmodells einen Mischverteilungsansatz zum Erzeugen der Sequenzen einsetzen, um gezielt Sequenzen hoher Bewertungen zu erzeugen und diesen Bias anschließend mit einem mathematischen Trick wieder herausrechnen, benötigt wesentlich weniger Sequenzen. Eine weitere Möglichkeit ist es, die empirische Verteilung im kritischen Bereich großer Bewertungen durch größere Intervalle zu vergröbern, und nur noch bestimmte Signifikanzschranken zu zulassen, die jedoch vertrauenswürdiger zugesichert werden können [Wu00].

Für PWM-Modelle, deren Gewichte ganze Zahlen in einem bestimmten Bereich sind, lassen sich exakte p-Werte auch mit dynamischen Programmieralgorithmen berechnen [Sta89a, Wu00]. Auch Hertzberg et al. [Her05] führen eine korrekte Berechnung des p-Wertes für den die *log-odds*-Bewertung einer PWM auf einer DNA-Sequenz durch. Dazu wird zunächst die höchste Bewertung S^* einer bestimmten PWM auf einer langen DNA-Sequenz bestimmt. Anschließend werden mittels einer *Branch & Bound*-Methode die

Menge aller möglichen DNA-Sequenzen der PWM-Länge W aufgezählt, die mindestens eine gleich hohe Bewertung durch die PWM erhalten würden. Die Wahrscheinlichkeit, mindestens eine dieser kurzen Sequenzen in einer zufälligen (langen) DNA-Sequenz zu beobachten, wird als p-Wert verwendet.

Eine weitere Herangehensweise ist die Annahme einer parametrischen Verteilung für die Wahrscheinlichkeitsverteilung von PWM-Bewertungen zufälliger Sequenzen. Eine beliebte Verteilungsannahme ist hierbei eine Extremwertverteilung [Gol94], die auch in Kapitel 5 auf Seite 113 angewendet wird. Beckstette et al. zeigen jedoch in [Bec07, Bec06], dass diese Familie von Wahrscheinlichkeitsverteilungen nicht besonders gut die wahre Verteilung von PWM-Bewertungen approximieren. Der Vorteil einer Extremwertverteilung ist sicher, dass ihre Verteilungsfunktion, die zur Berechnung der p-Werte benötigt wird, im Gegensatz zur Verteilungsfunktion einer Normalverteilung, analytisch berechenbar ist.

Die bekannte Arbeit von J. M. Claverie und S. Audic [Cla96] geht dennoch davon aus, dass die Bewertungen S einer PWM, also Summe vieler ähnlicher Zufallsvariablen, gemäß $\mathcal{N}(\mu, \sigma^2)$ normalverteilt sind. Sei $c(S) \approx P(S \leq S')$ die numerisch angenäherte Verteilungsfunktion dieser Normalverteilung. Laut Claverie und Audic eignet sich diese Verteilungsfunktion nicht für die Berechnung der gesuchten p-Werte, da sie nicht trennscharf genug ist. Vielmehr sollte die Länge L der mit PWM durchsuchten Sequenz berücksichtigt werden. Dies wird erreicht, in dem die Wahrscheinlichkeit $c(S)^{L-W+1}$ dafür, dass jede der möglichen $L - W + 1$ Sequenzfenster eine Bewertung hat, der kleiner als S ist, verwendet wird. Der gesuchte p-Wert ist dann einfach $p(S) = 1 - c(S)^{L-W+1}$. Im Prinzip folgen die Autoren in dieser Arbeit auf ungewöhnlichem Weg dem Prinzip der Anpassung der Signifikanzschranke bei multiplen statistischen Tests.

Stärken und Schwächen von Gewichtsmatrizen Die Gewichtsmatrix ist auch in der heutigen Zeit das vorherrschende TFBS-Modell, obschon meist integriert in umfassendere Systeme zur Vorhersage von regulativen Sequenzen. Es sprechen eine Reihe von Gründen für die Verwendung dieser einfachen Modelle. Die Repräsentation von Bindungsstellen in solchen Matrizen ist intuitiv und auch für Nichtinformatiker leicht zu erschließen. Die einfache mathematische Struktur ermöglicht zum Einen effiziente Lernverfahren, zum Anderen die analytische Herleitung der statistischen Signifikanz von Vorhersagen. Weitere Vorzüge für Gewichtsmatrizen ergeben sich dagegen eher aus ihrer derzeitig "marktbeherrschenden" Stellung. PWM stehen in großer Anzahl in mehreren Datenbanken zur Verfügung (vergleiche Unterabschnitt 3.1.4), neue Bindungssequenzen wurden in unzähligen Veröffentlichungen als Matrix dargestellt. Große genomische Datenbanken lassen sich bei Eingabe einer Menge von PWM mit eingebetteten Programmen durchsuchen.

Einige der genannten Vorteile bedingen auch die zwei großen Unzulänglichkeiten von PWM-Modellen. Zum einen ist dies die prinzipielle Unabhängigkeit des Beitrages einer PWM-Spalte zur Bewertung von benachbarten (bzw. anderen) Spalten. Die Unabhängigkeitsannahme ergibt sich durch das additive Bewertungsschema. Jedoch wurde in meh-

reren Veröffentlichungen gezeigt, dass diese Annahme häufig nicht zutreffend ist (siehe Seite 44).

Zum anderen können TFBS mit PWM-Modellen nur auf Grundlage der spaltenweisen Sequenzähnlichkeiten beschrieben werden. Verborgene oder komplexere Eigenschaften einer gültigen TFBS können nicht adäquat beschrieben werden. Im günstigen Fall resultieren verborgene Eigenschaften in starken Sequenzähnlichkeiten, und werden komplexere Eigenschaften indirekt in den Spaltengewichten ausgedrückt. Dafür ist jedoch die Anzahl der PWM-Parameter (die Gewichte) häufig unnötig groß. Im ungünstigen Fall kann keine Sequenzähnlichkeit festgestellt werden, die nützliche Information verborgener und zusammenhängender Eigenschaften wird zur Klassifikation dann nicht verwendet. Da ein Hauptproblem der PWM die hohe Anzahl von Falsch-Positiv-Treffern ist, ist eine Berücksichtigung der *wahren* TFBS-Eigenschaften ein Ansatzpunkt für Verbesserungen.

Ein Problem, dass nicht nur bei der Verwendung von PWM auftritt, jedoch aufgrund der häufigen Verfügbarkeit hier nicht außer Acht gelassen werden darf, ist die Qualität der Modelle in Fällen, in denen nur wenige TFBS für die PWM-Konstruktion zur Verfügung standen. PWM, die auf nur wenigen (z.B. fünf) Sequenzen basieren, sind in TRANSFAC keine Seltenheit. Die Wahl einer Bewertungsschranke wird hier zu der Wahl zwischen den Extremen, keine Treffer zu erhalten (extreme Überanpassung) und viel zu vielen Treffern. Rahmann et al. [Rah04] zeichnen ein ernüchterndes Bild der Qualität von TRANSFAC-PWM-Modellen.

3.1.3 Komplexe TFBS-Modelle

Die begrenzten Erkennungsleistungen von Gewichtsmatrizen und zeichenkettenbasierten Repräsentationen sorgen seit ihrer Einführung für vielfältige Anstrengungen, geeignetere TFBS-Modelle zu entwickeln. In diesem Unterabschnitt werden einige Vertreter solcher Modellierungsansätze vorgestellt.

Eine einfache Erweiterung des PWM-Modells ist das *generalized profile* von Bucher et al. [Buc94]. Es besteht im Wesentlichen aus zwei getrennten PWM-Modellen, die in einem variablen Abstand auf die zu durchsuchenden Sequenzen angewendet werden, wobei jeder Abstand einen eigenen Anteil an der Bewertung einer Teilsequenz besitzt. So können wahrscheinliche Abstände zwischen zwei Teilen einer flexibel langen TFBS mit hohen Bewertungsbeiträgen versehen werden, unwahrscheinliche Abstände dagegen mit einer sehr niedrigen Bewertung. Erfolgreich eingesetzt wurden diese Sequenzmodelle in [Rou00] auf CTF/NFI-Bindungssequenzen, die aus zwei Teilen bestehen, die in einem variablen Abstand voneinander stehen.

Djordjevic et al. [Djo03] stellen ein Modell und einen Algorithmus (*QPMEME* für *Quadratic Programming Method of Energy Matrix Estimation*) vor, mit dem die versuchen, in direkter Weise die Bindungsenergie zwischen einem Transkriptionsfaktor und einer kurzen DNA-Teilsequenz abzuschätzen und diese Abschätzung zur Erkennung von TFBS

zu verwenden. In der detaillierten mathematischen Modellierung bedienen sie sich eines Ansatzes von Stormo und Fields [Sto98], in dem sowohl der Beitrag einzelner Basen zur Bindungsenergie als auch der gegenseitige Einfluss benachbarter Basen berücksichtigt wird. Für die anschließende Maximum-Likelihood-Schätzung muss jedoch die Berücksichtigung des gegenseitigen Einflusses fallen gelassen werden. Obwohl die Forscher ihre Methode gegen informationstheoretische PWM abgrenzen, müssen sie eingestehen, dass in Fällen, in den die grundlegende Affinität eines TF für die Bindung an DNA gering ist, ihr Modell äquivalent zu den gewöhnlichen PWM ist. Für TF, die sehr unspezifisch binden, können sie jedoch gegenüber diesen PWM eine bedeutende Steigerung der Erkennungsleistung vorweisen.

Eine einfache Kombination von drei Ansätzen zur Erkennung regulativer Sequenzen in einer komplexeren Entscheidungsregel schlagen Levy et al. [Lev02] vor. Diese drei Ansätze sind die Suche statistisch signifikanter PWM-Treffer, die Untersuchung des Konservierungsgrads der DNA-Sequenz und die Berücksichtigung von TFBS-Häufungen in der Nachbarschaft einer Position. Sinngemäß wird an einer Sequenzposition dann eine TFBS vorhergesagt, wenn diese in einem zu mindestens 90% konservierten Bereich liegt (gemessen zwischen Mensch- und Maus-Sequenzen) und zudem eine bestimmte Signifikanzschranke unterschritten wird.

PWM-Modelle in Mischverteilungsansätzen. In einigen Fällen besitzen Transkriptionsfaktoren mehrere, verschiedenartige Bindungsmotive, etwa, wenn sie in verschiedenen Konformationen vorliegen können oder sie auf verschiedene Weise mit anderen Faktoren ko-agieren können und dies Einfluss auf die bevorzugte Bindungssequenz hat (siehe z.B. [Bil05]). PWM-Modelle, die auf einem lückenlosen Alignment aller möglichen Bindungssequenzen trainiert werden, würden in diesem Fall eine sehr schwache Leistung zeigen. In verschiedenen Arbeiten wurde deshalb ein Mischverteilungsansatz mit verschiedenen PWM-Modellen für die verschiedenen Bindungsmotive eines Faktors vorgeschlagen. Die Mischverteilungskoeffizienten sowie die Komponenten (die PWM-Modelle) werden dazu mittels eines EM-Verfahrens trainiert [Han05]. Georgi und Schliep [Geo06] wenden den Mischverteilungsansatz für einzelne PWM-Spalten anstatt für ganze Gewichtsmatrizen an. So können sie lokale Unterschiede in den verschiedenen Bindungsmotiven eines Faktors genauer modellieren und gemeinsame Teile traditionell in einem Modell belassen.

Abhängigkeiten zwischen Sequenzpositionen. Eine wichtige Annahme von PWM-Modellen ist die statistische Unabhängigkeit zwischen den Spalten des Lernalignments. Abgesehen davon, dass die statistische Unabhängigkeit von benachbarten DNA-Positionen auch für nicht funktionale DNA nicht vorliegt[5], konnten verschiedene Studien zeigen, dass diese Annahme für einige konkrete Transkriptionsfaktoren nicht zutreffend ist. Wolfe et al. [Wol99] zeigten, dass es einen zuvor proklamierten Protein-DNA-Erkennungscode

[5] Wie ließe sich das Phänomen der CpG-Inseln erklären?

für Zinkfinger-Transkriptionsfaktoren (siehe Unterabschnitt 2.2.3), der isoliert Aminosäuren einzelnen Nukleotiden zuordnet (also einen additiven Beitrag jeder TFBS-Position) nicht gibt. Die Annahme eines additiven, unabhängigen Beitrags sei eine Vereinfachung der biologischen Realität. Man und Stormo [Man01] wiesen nach, dass die Positionen 16 und 17 der TFBS für den Faktor *Mnt* nicht unabhängig zu der DNA-Protein-Bindung beitragen. Weitere Hinweise für nicht-additive Beiträge von TFBS-Positionen zur Bindung lieferten Bulyk et al. [Bul02] mit ihrer Untersuchung von Mutanten des Zinkfinger-Proteins EGR1 in Mäusen.

Diese neuen Erkenntnisse weckten Hoffnung, dass die häufig unbefriedigenden Fehlerraten von PWM-Modellen überwunden werden könnten, wenn Modelle eingesetzt würden, die Abhängigkeiten zwischen den Positionen berücksichtigen (siehe auch [Ben02a] als Beitrag der damaligen Diskussion). Eine Reihe von Arbeiten setzten direkt an der Erweiterung von PWM an, während andere neue Wege bei der Modellierung von TFBS einschlugen.

Schon in den frühen 90er Jahren hatten Zhang et al. [Zha93] das PWM-Modell auf das Alphabet $\Sigma_{DNA} \times \Sigma_{DNA}$ aller Dinukleotide erweitert und diese Dinukleotid-Gewichtsmatrizen zur Vorhersage von *Splice*-Stellen eingesetzt. Sie konnten zeigen, dass es auch bei *Splice*-Stellen häufig schwache Korrelationen zwischen benachbarten Spalten eines Alignments gibt, und die Berücksichtigung dieser Abhängigkeiten die Klassifikationsfehlerraten reduzieren kann. Gereshenzon et al. [Ger05] griffen diese einfache Erweiterung für die TFBS-Erkennung auf.

Ein Problem der Dinukleotidmatrizen ist die Robustheit der Parameterschätzung aus den gewöhnlich recht kleinen Alignments. Trotz der recht hohen Zahl an zu schätzenden Parametern ist das Dinukleotid-PWM-Modell nicht in der Lage, Abhängigkeiten zwischen beliebigen Spalten abzubilden. Eine flexiblere Handhabung des Sachverhaltes, der ebenfalls nicht weit von dem PWM-Ansatz entfernt liegt, wird in einer Arbeit von Zhou et al. [Zho04] vorgestellt. Im Falle einer Abhängigkeit zwischen zwei Spalten, die nicht zwingend benachbart sein müssen, wird anstelle der beiden Spaltenvektoren eine 4×4-Matrix eingesetzt, welche die gemeinsame Verteilung der beiden Spalten repräsentiert. Die übrigen Spalten werden herkömmlich modelliert. Das additive Bewertungsschema ändert sich nur dahingehend, dass vereinzelt der Bewertungsbeitrag eines Spaltenpaars auftritt.

Bayessche Netze, ein maßgeblicher Modellierungsansatz in dieser Dissertation, drängen sich für Anwendungen, in denen Abhängigkeiten berücksichtigt und gleichzeitig die Anzahl freier Parameter gering gehalten werden soll, geradezu auf. Im Bereich der DNA-Sequenzanalyse wurden sie erstmals von Cai et al. [Cai00] für das verwandte Problem der *Splice*-Stellenerkennung eingesetzt. Er verglich baumartige Bayessche Netze mit PWM-Modellen und Markovketten erster Ordnung. Barash et al. [Bar03] wendeten Bayessche Netze erstmals auf das TFBS-Erkennungsproblem an. Sie untersuchten verschiedene Klassen Bayesscher Netz-Klassifikatoren, darunter den im folgenden Kapitel in Abschnitt 4.3.2 auf Seite 75 vorgestellten TAN. Eine weitere Arbeit verwendet Bayessche

Netze variabler Ordnung, wobei sich *Ordnung* auf die maximale Anzahl von Sequenzpositionen bezieht, von denen eine bestimmte Position beeinflusst wird [BG05]. Diese Ordnung wird lokal und gemäß der Datenlage variiert.

Strukturelle Eigenschaften von DNA und der Bindung. Einige Arbeiten versuchen, direkt die Protein-DNA-Bindung zu modellieren, indem Kontakte zwischen bestimmten Aminosäuren der Bindungsdomäne ganz bestimmten Nukleotiden einer TFBS zugewiesen werden. Erste Untersuchungen diesbezüglich gehen auf Seeman et al. [See74] zurück. Dort wurde die Hoffnung geweckt, es gäbe einen deterministischen Code zwischen Aminosäuren und Nukleotiden. In der Folgezeit wurde die Existenz eines solch einfachen Codes u.a. durch Pabo et al. [Pab00] widerlegt, wobei gewisse Präferenzen einiger Aminosäuren für bestimmte Nukleotide eingeräumt wurden (siehe auch [Suz97]).

Mandel-Gutfreund et al. setzten darauf auf und verwenden kristallographische Daten, um statistisch zu untersuchen, welche Aminosäure mit welchem Nukleotid bei der Bindung eines Transkriptionsfaktors auf der DNA in Wechselwirkung treten [MG01]. Dazu stellten Sie eine Art Codetabelle auf, die jedem Aminosäure-Nukleotid-Paar ein Gewicht, ähnlich den *log-odds*-Bewertungen zuweist. Für die Suche von optimalen TFBS für einen Faktor muss dessen Bindungsdomäne im Detail vorliegen. Die Eignung einer Sequenz als TFBS des Faktors wird über die Summation der passenden Gewichte in der Codetabelle berechnet. Auch hier liegt also ein additives Bewertungsschema vor. Ein etwas komplexeres mathematisches Modell der Interaktion zwischen Transkriptionsfaktor und seiner Bindungsstelle wurde von Panayiotis et al. [Ben02b] aufgestellt. Dieses kommt ohne Unabhängigkeitsannahmen zwischen den Einzelkontakten aus. Von Kono et al. [Kon99] kommt eine weitere Arbeit, die direkte Wechselwirkungen zwischen Aminosäuren und Basen untersucht. Hier werden die Aufenthaltsorte von Aminosäuren in der Umgebung von Nukleotiden geometrisch und statistisch aus Kristallographiedaten analysiert und zur Erkennung von TFBS verwendet. Die gleiche Arbeitsgruppe verwendete diese Datenbasis dazu, mittels eines neuronalen Netzes Transkriptionsfaktoren vorherzusagen [Ahm04].

Ein Problem all dieser Ansätze ist, dass sie schlecht für die *alltägliche* Suche nach TFBS für wenig beforschte Transkriptionsfaktoren einsetzbar sind, da entsprechende strukturelle Informationen zu den meisten Faktoren fehlen. Einen alternativen Ansatzpunkt zur Berücksichtigung struktureller Eigenschaften stellen die sequenzabhängigen DNA-Strukturparameter dar, die in Unterabschnitt 2.1.1 auf Seite 9 vorgestellt wurden. Karas et al. [Kar96] arbeiteten auf sequenzabhängigen Parametern, die in jedoch auf Tetrameren, nicht auf Dinukleotiden, definiert sind, und entwickelten erstmals strukturelle Motive für die TFBS-Erkennung. Baldi et al. [Bal98] stellen mit Hilfe dieser Parameter Strang-invariante Codes zur Beschreibung von TFBS auf und verwenden diese in einem HMM zur Vorhersage von Promotoren. Zuvor untersuchten Ponomarenko et al. [Pon97] bekannte TFBS hinsichtlich struktureller Auffälligkeiten. Sie konnten dabei u.a. strukturelle Motive in der Nachbarschaft von TATA-Boxen feststellen und diese erfolgreich zur Verbesserung der TFBS-Vorhersage im Vergleich mit PWM-Modellen einsetzen. In einer

Folgearbeit [Pon99] stellen sie die verwendeten strukturellen DNA-Parameter in der Datenbank *BDNA-VIDEO* zusammen, die auch Grundlage für die in Kapitel 5 verwendeten strukturellen Eigenschaften ist. Zwar nicht zur Modellierung einzelner TFBS, jedoch im Zuge der Promotorenerkennung setzten auch Ohler et al. auf strukturelle Eigenschaften, zusätzlich zu Sequenzinformationen. Die Sequenzen modellierten sie in einem Markovmodell, die kontinuierlichen Strukturwerte mit Gaussverteilungen [Ohl01].

Verwenden heterogener Informationen zu TFBS. Kielbasa et al. verwenden in einem deterministischen Motivsuchansatz (siehe Unterabschnitt 3.2) Positionsverteilungen und signifikante Häufungen von TFBS zur Vorhersage [Kie01]. Positionsverteilungen versprechen jedoch nur in sehr einfach strukturierten Genomen, wie z.B. bei Bakterien, eine Verbesserung der Erkennungsleistung.

Zwar nicht auf dem Gebiet der TFBS-Erkennung, sondern zur Modellierung von Proteinfamilien, gingen Plotz et al. [Plo04] einen ähnlichen Schritt wie der Autor dieser Dissertation in Kapitel 5, der Abkehr von reinen Sequenzen zur Beschreibung von Motiven, hin zu einer allgemeineren Beschreibung durch eine Menge von Merkmalen. Im Falle der Arbeit von Plot et al. stammen diese Merkmale vor allem aus Homologieanalysen. Anstelle einer Aminosäurensequenz liegt zunächst ein mehrdimensionales Signal vor, dessen relevante Information über eine Wavlettransformation extrahiert wird (Dimensionsreduzierung). Das nach wie vor kontinuierliche Signal wird in kontinuierlichen HMM verarbeitet.

Jolly et al. [Jol05] verwendeten zur Vorhersage von TFBS des Hefe-Transkriptionsfaktors Ndt80 PWM-Modelle, die auf Grundlage von Bindungsaffinitäten konstruiert wurden, sowie die evolutionäre Konserviertheit und Positionsverteilungen. Die drei Arten der Information werden jedoch völlig unabhängig und teilweise manuell ausgewertet und erst in einer gemeinsamen Bewertungsfunktion vereint.

Ein Ansatz von Narlikar e al. [Nar06b], der in Richtung des in Kapitel 5 vorgestellten Modells geht, modelliert gemeinsame Eigenschaften von TFBS ganzer Transkriptionsfaktorfamilien. Dazu verwenden sie Wissen über die strukturellen Eigenschaften der jeweiligen Domäne sowie einige Sequenzmerkmale. Die trennschärfsten Merkmale werden in einem kombinierten Trainings- und Merkmalsauswahlverfahren ermittelt.

Gerade in jüngster Zeit wurden Arbeiten veröffentlicht, die eine kombinierte Analyse von Sequenzdaten und Genexpressionsdaten versuchen. Im Allgemeinen werden dabei TFBS bzw. TFBS-Module gesucht und gleichzeitig analysiert, wie gut diese Vorhersagen vorhandene Genexpressionsdaten erklären können, indem Gene mit gleichen Expressionsmuster ähnliche TFBS in ihrem Promotor ausweisen, und andere Gene diese TFBS nicht besitzen [Aer03, Bus00, Liu05, Kel06].

3.1.4 Datenbanken

Im Rahmen der Erforschung regulativer DNA-Sequenzen haben sich eine Reihe von Datenbanken etabliert, die entsprechende Sequenzen der Forschungsgemeinde bereit stellen.

TRANSFAC. Die umfangreichste Datenbank für eukaryontische Transkriptionsfaktoren und vor allem ihrer Bindungsstellen ist TRANSFAC, die von dem Privatunternehmen *BioBase GmbH* in Braunschweig entwickelt wurde und gepflegt wird [Mat06]. Ihre vier Haupttabellen beinhalten 1.) Individuelle TFBS, 2.) individuelle Transkriptionsfaktoren mit Einordnung in ein Klassifikationsschema, 3.) regulatorische Bereiche von Genen, in denen individuelle TFBS bekannt sind sowie 4.) PWM-Modelle, die aus Alignments bekannter TFBS konstruiert sind. In der Version 10.3 liegen insgesamt 19′114 TFBS für 9621 Faktoren und 821 PWM-Modelle vor. Daneben gibt es in den neueren Versionen einen Bereich für Sequenzen aus *ChIP on chip*-Experimenten. Alle Daten werden von den TRANSFAC-Mitarbeitern aus der Originalliteratur zusammengestellt. Dieser Aufwand verhindert jedoch nicht, dass es zu Redundanzen und Widersprüchen im Datenbestand kommt. Das wird auch dadurch ermöglicht, dass die Datenbank aus schlichten ASCII-Dateien besteht und nicht innerhalb eines relationalen Datenbanksystems läuft. Damit verbunden ist die etwas umständliche Recherche mittels der angebotenen Webbrowser-Schnittstelle. Ein weiteres Problem ist, dass TRANSFAC keine verlässliche Annotationskonventionen durchsetzt.

TRANSFAC ist eng verzahnt mit Schwesterndatenbanken, die anderen Aspekte der Genregulation dienen. Ein für diese Arbeit relevantes Schwesternprojekt ist TRANSCOMPEL, eine Datenbank speziell für TFBS-Paare kooperierender Transkriptionsfaktoren [Mat06]. Eine weitere wichtige Datenbank von BioBase ist TRANSPATH, die Informationen über genregulatorische Netzwerke enthält.

Die Datenbanken von BioBase sind nicht frei, jedoch sind ältere Versionen öffentlich zugänglich. Zusammen mit der Datenbank selbst sind einfache Software-Hilfsmittel verfügbar, etwa zur Suche von TFBS in DNA-Sequenzen mittels der TRANSFAC-PWM (MATCH [Kel03]).

JASPAR. Eine weitere Datenbank für eukaryontische Transkriptionsfaktoren und deren Bindungsstellen ist JASPER [San04]. Im Gegensatz zu TRANSFAC ist diese frei zugänglich, jedoch ebenfalls nur über ein Webinterface. JASPAR beschränkt sich zudem auf PWM-Modelle, wobei die Sequenzen der zugrunde liegenden Alignments von TRANSFAC übernommen wurden. Diese Modelle haben oft eine höhere Qualität als die TRANSFAC-Modelle und können auf der Webseite von JASPAR zur Suche in hochgeladenen Sequenzen verwendet werden.

EPD. Die Eukaryontische Promotoren Datenbank (EPD) ist eine Sammlung annotierter Promotoren für proteinkodierende Gene in Eukaryonten [Pra02]. In jeder einzelnen Sequenz liegt eine experimentell validierte TSS vor. Es können einfach Gruppen von Promotoren, z.B. einer bestimmten Spezies, heruntergeladen werden. Die Sequenzen enthalten Verweise auf andere Datenbanken, wie EMBL oder TRANSFAC. In der aktuellen Version (92) enthält die EPD 4809 Promotorsequenzen, davon 1871 menschliche, die als Datensatz in Kapitel 5 verwertet werden.

TRRD. Bei der *Transcription Regulatory Region Database* (TRRD) handelt es sich um eine Datenbank, die sich zum Ziel gesetzt hat, die Regulation von Genen umfassend zu erklären. Zentrale Objekte dieser Datenbank sind daher Gene, für die Informationen über regulatorische Regionen, einzelne TFBS oder Expressionsmuster vorliegen. Der Ansatz ähnelt dem von TRANSFAC, auch in der Hinsicht, dass Daten aus biologischen Veröffentlichungen akquiriert werden. Inzwischen arbeiten viele TRRD-Verantwortliche für TRANSFAC und eine Weiterentwicklung von TRRD ist nicht erkennbar. Zum Zeitpunkt der Veröffentlichung [Kol02] enthielt die TRRD 1167 Geneinträge mit insgesamt 5537 TFBS.

3.2 Motivsuche

Liegen ausreichend bekannte TFBS in Form eines lückenlosen Alignments vor, ist die Konstruktion bzw. das Lernen von TFBS-Modellen häufig recht einfach. Consensus-Sequenzen lassen sich spaltenweise durch Anwendung der Regeln auf Seite 36 bilden. Die Berechnung der Gewichte einer PWM beruht meist auf Häufigkeitsstatistiken der Nukleotide in den einzelnen Spalten des Alignments. Diese komfortable Ausstattung mit wohlgeformten Sequenzdaten ist jedoch nicht vorhanden, wenn TFBS-Modelle aus *ChIP on chip*-Daten oder SELEX-Daten gewonnen werden sollen. Daten dieser Art bestehen aus mehr (*ChIP on chip*) oder weniger (SELEX) langen Sequenzen, von denen ziemlich sicher angenommen werden kann, dass derselbe Transkriptionsfaktor jede von ihnen an einer unbekannten Position binden kann. Da sich derartige Untersuchungen häufig mit Transkriptionsfaktoren beschäftigen, für die noch kein TFBS-Modell vorliegt, kann man nicht auf ein Solches zurückgreifen, um die unbekannten TFBS-Positionen aufzuspüren.

Unter dem Begriff *Motivsuche*[6] werden Verfahren zusammengefasst, die bei Eingabe einer Menge von Sequenzen entweder eine Gruppe von Teilsequenzen extrahieren, die sich einander stark ähneln und in möglichst vielen der Eingabesequenzen vorkommen, und die zum Lernen eines Sequenzmodells verwendet werden, oder auf anderem Wege Sequenzmodelle finden, die in möglichst vielen Eingabesequenzen Treffer besitzen. Die geforderte Ähnlichkeit der mutmaßlichen Modellinstanzen ergibt sich aus dem zugrunde liegenden Modellierungsansatz. Die Elemente einer Menge von Modellinstanzen sind sich

[6] Im Englischen sind die Bezeichnungen *motif discovery* oder *de novo motif search* üblich.

einander um so ähnlicher, um so stärker das aus ihnen berechenbare Sequenzmodell ist. Im Falle einer Gewichtsmatrix könnte der Informationsgehalt als Maß der Stärke dienen. Die Forderung, das die Vorkommen der Modellinstanzen auf möglichst viele der Eingabesequenzen verteilt sein sollen, ergibt sich aus der Motivation, das Sequenzmodell als Erklärungsmodell für einen gemeinsamen biologischen Hintergrund der Eingabesequenzen zu verwenden. Im Falle von SELEX-Daten besteht dieser gemeinsame Hintergrund beispielsweise in der Fähigkeit aller Sequenzen, ein bestimmtes Protein zu binden. Sich ähnelnde, häufig auftretende Teilsequenzen sind die mutmaßliche Ursache dieser Fähigkeit — die gesuchten Bindungssequenzen.

Dieser Abschnitt enthält einen Überblick über gängige Motivsuchverfahren, ohne deren Leistung zu beurteilen. Für einen Überblick der Leistungsfähigkeit bekannter Methoden sei der Übersichtsartikel [Tom05] empfohlen.

Die zahlreichen Motivsuchverfahren lassen sich grob in zwei große Hauptgruppen gliedern: 1.) vollständige, deterministische Verfahren und 2.) stochastische Verfahren. Diese werden in den Unterabschnitten 3.2.1 und 3.2.2 vorgestellt.

3.2.1 Deterministische Verfahren

Für die Suche nach zeichenkettenbasierten Sequenzmotiven, wie die in Abschnitt 3.1.1 vorgestellten, werden deterministische Motivsuchverfahren eingesetzt. Nach Brazma et al. sollten innerhalb der deterministischen Verfahren *musterorientierte* und *sequenzorientierte* Aufzählalgorithmen unterschieden werden [Bra98].

Musterorientierte Ansätze. Verfahren dieser Klasse zählen einen Lösungsraum möglicher Sequenzmotive einer gewählten Klasse vollständig auf. Die Sequenzdaten werden verwendet, um für jedes der Muster eine Güte zu berechnen. Die Sequenzmotive mit dem höchsten Gütewert werden von den Verfahren als Lösung ausgegeben. Die bekannten Gütefunktion berücksichtigen in verschiedener Weise die Anzahlen der Treffer des Sequenzmotivs in den Eingabesequenzen bzw. die Anzahl der Eingabesequenzen, die solche Treffer besitzen. Der Vorteil von musterorientierten Verfahren ist, dass sie innerhalb ihres Lösungsraums garantiert die optimale Lösung, d.h. das stärkste Sequenzmotive finden. Ihr Nachteil ist die hohe Zeit- und Raumkomplexität, die sich aus dem Aufzählen des gesamten Lösungsraums ergibt.

Ein sehr frühes Verfahren [Wat84] arbeitet auf einem Lösungsraum, der alle konkreten Sequenzen der Länge W über dem Basisalphabet Σ_{DNA} enthält. Für jedes der Sequenzmotive wird die statistische Signifikanz bestimmt. Die Verfahren in [Sta89b] und [Wol96] arbeiten ebenfalls auf Sequenzmotiven, die nur Zeichen aus Σ_{DNA} enthalten, jedoch wird als Güte eines Motivs der Informationsgehalt eines lückenlosen Alignments bestimmt, dass jene Teilsequenzen enthält, die einen hohen Grad an Übereinstimmung mit dem Sequenzmuster zeigen.

Beim Übergang zu Sequenzmotivklassen höherer Stufen tritt zunehmend das Problem auf, dass der Lösungsraum in so großem Maße wächst, dass ein vollständiges Aufzählen und Evaluieren jedes Sequenzmotivs nicht praktikabel ist. Dennoch sind solche Sequenzmotive von Interesse, da sie die Variabilität von TFBS besser ausdrücken können. Einen Ausweg schlug Smith et al. vor, indem sie zwar Sequenzmotive einer höheren Stufe betrachten, sich andererseits auf eine stark eingeschränkte Teilmenge innerhalb dieser Stufe beschränkten [Smi90].

Anstatt den Lösungsraum über eine eingeschränkte Musterdefinition zu verkleinern, kann er auch während der Suche nach guten Mustern dynamisch beschnitten werden. Dabei wird ausgenutzt, dass sich Sequenzmuster häufig kompositorisch zu längeren Sequenzmotiven zusammensetzen lassen. Der Lösungsraum lässt sich dann in einer baumartigen Datenstruktur darstellen. Die Untersuchung eines Teilbaums kann abgebrochen werden, wenn zu seinem Wurzelknoten gehörende, kürzere Sequenzmuster schon nicht den Qualitätskriterien genügt [Sag95, Neu94, Jon95].

Eine Reihe von Verfahren operieren nicht direkt auf den Eingabesequenzen, sondern erzeugen zunächst *Suffixbäume*[7] oder verwandte Datenstrukturen aus der Eingabe, um anschließend bestimmte Operationen, wie die Suche einer Teilsequenz wesentlich effizienter durchführen zu können. Erstmalig vorgeschlagen wurde ein solches Vorgehen von Marie Sagot [Sag98]. Suffixbäume wurden bis dahin vor allem im Bereich des *Text mining* eingesetzt. Die zeitliche und räumliche Komplexität der Konstruktion eines Suffix-Baumes ist linear in der Länge der Eingabe. Anschließend können Teilsequenzen der Länge W in $\mathcal{O}(W)$ Zeit gesucht werden.

Suffixbäume gehören zu der Klasse der Indexdatenstukturen. Weitere Indexstrukturen wurden in ähnlicher Weise zur Motivsuche eingesetzt. Die Autoren des Algorithmus SMILE [Mar00] ermöglichen über zusätzliche Verknüpfungen die fehlertolerante Suche von Sequenzmotivinstanzen. Pavesi setzen auf der SMILE-Idee auf und verbessern die Laufzeit in ihrem Algorithmus *Weeder* gegenüber SMILE deutlich [Pav01]. Eskin et al. verwenden in ihrem MITRA-Algorithmus [Esk02] mit *Mismatch*-Baum eine weitere Indexstruktur.

Sequenzorientierte Ansätze. Bei musterorientierten Motivsuchverfahren fällt den Eingabedaten lediglich die Rolle als Evaluierungsdatensatz für Sequenzmotive des Lösungsraumes zu. Sequenzorientierte Verfahren dagegen generieren, einer bestimmten Strategie folgend, Mengen von Teilsequenzen aus der Eingabe, aus deren Alignment Kandidatenmuster erzeugt werden. Ziel ist es, ein Teilsequenz-Alignment zu finden, das ein möglichst starkes Sequenzmotiv impliziert. Da die Untersuchung aller möglichen Mengen

[7] Ein *Suffixbaum* einer Sequenz $\boldsymbol{X} = X_1 \cdots X_L$ ist ein gerichteter Baum mit L Blättern (beschriftet mit den Indizes $i = 1 \cdots L$), dessen Kanten mit Teilsequenzen aus $\boldsymbol{X}$ beschriftet sind. Die inneren Knoten haben mindestens zwei Kinder, wobei die Beschriftungen der ausgehenden Kanten mit verschiedenen Symbolen beginnen. Die verketteten Kantenbeschriftungen von der Wurzel zu einem Blatt i ergibt den Suffix von $\boldsymbol{X}$ ab Index i.

von Teilsequenzen einer Eingabe schon für kurze Eingabesequenzen und kleine Sequenzmotivlängen W kaum durchführbar ist, wurden eine Vielzahl von Strategien entwickelt, möglichst schnell gute Alignments zu erreichen.

Der bekannte Algorithmus WINNOWER, vorgestellt von Pevzner et al. [Pev89], setzt dazu auf eine graphenbasiertes *Clustering*-Methode. Die Knoten des zugrunde liegenden Graphen sind mit den Eingabeteilsequenzen der Länge l beschriftet. Die Knotenmenge wird partitioniert, so dass jede Partition die Teilsequenzen einer einzelnen Eingabesequenz enthält. Zwei Knoten des Graphen werden durch eine Kante verbunden, wenn sich ihre Sequenzen in höchstens d Zeichen unterscheiden. Das WINNOWER-Verfahren versucht nun, möglichst große Cliquen zu finden, die sich über möglichst viel Partitionen erstrecken. Weiterentwicklungen sind SP-STAR [Pev00], dass eine andere Bewertungsfunktion für die gefundenen Muster verwendet, und cWINNOWER [Pev89], das ein auf Consensus-Sequenzen basierendes Kriterium für das Ziehen einer Kante zwischen zwei Knoten verwendet[8].

Ein weiteres *Clustering*-Verfahren wurde von Styczynski [Sty04] vorgestellt. Es arbeitet auf einer Distanzmatrix, in der die Hammingdistanzen aller Paare der in der Eingabe vorkommenden Teilsequenzen eingetragen sind. Olman et al. [Olm03] bilden Teilsequenzen der Eingabe in einem Raum ab, in denen ähnliche Sequenzen eine kleine Hammingdistanz haben. *Cluster* dieses Raumes werden in Beziehung zu Teilbäumen eines *Minimal Spanning Trees* gesetzt, der anhand der Daten aufgebaut wird. In [Yad98] ist ebenfalls ein *Clustering* von Teilsequenzen der Eingabe vorgesehen. Aus den Sequenzen jedes identifizierten *Clusters* wird ein Hidden Markov Modell gelernt, das verwendet wird, um in iterative Weise die *Cluster* zu verfeinern. Dieses Verfahren steht zwischen den deterministischen und den stochastischen Verfahren.

Ein weiteres Verfahren der Pevzner-Gruppe [Kei02], *Motifprofiler*, versucht, die Vorzüge der sequenzmusterbasierten Verfahren, nämlich kein Motiv versehentlich zu übersehen, mit denen der sequenzbasierten Verfahren, nämlich nicht den gesamten Lösungsraum aufzählen zu müssen miteinander zu kombinieren. Der Algorithmus versucht, alle k-Nachbarschaften aller in der Eingabe vorkommenden Teilsequenzen zu bestimmen. Die k-Nachbarschaften einer Sequenz besteht aus allen Sequenzen, die sich von ihr in maximal k Positionen unterscheiden. Gesucht werden natürlich möglichst große k-Nachbarschaften. Vishnevsky et al [Vis05] verfolgen eine ähnliche Strategie in ihrem Verfahren namens *ARGO-Motifs*.

Eine weitere Gruppe von Motivsuchverfahren, die allesamt als *wörterbuchbasierte* Methoden bezeichnet werden, bildet vielversprechende Sequenz-Teilmengen in iterativer Weise durch Verkettung kürzerer Sequenzen, die zuvor das Gütekriterium passiert haben. Ein

[8] Pevzner lancierte mit dieser Arbeit gleichzeitig einen Wettbewerb in Form einer algorithmischen Herausforderung, auf die in der Folgezeit von vielen Arbeitsgruppen bei der Veröffentlichung ihrer Algorithmen Bezug genommen haben: Sei eine Eingabe bestehend aus 20 Sequenzen der Länge 600 bp gegeben. Jede der 20 Sequenzen enthalte einen Treffer für ein Sequenzmuster der Länge $L = 15$, wobei dieser Treffer in höchstens $k = 4$ Positionen vom Muster unterscheiden darf. Finde dieses Sequenzmuster!

bekannter Vertreter ist der Algorithmus *MobyDick* von Bussemaker et al [Bus00]. MobyDick legt zunächst ein Wörterbuch an, das lediglich aus den 4 Nukleotiden `A`, `C`, `G` und `T` besteht. Die Wörter (Sequenzen) des Wörterbuches sind zu jedem Zeitpunkt in Bezug auf die statistische Signifikanz ihrer Überrepräsentiertheit in den Daten sortiert (im initialen Wörterbuch schlicht nach der Häufigkeit der Nukleotide in den Eingabedaten). In den folgenden Iterationen werden durch Verkettung möglichst signifikanter Sequenzen längere Sequenzen gebildet. Sind auch diese längeren Sequenzen signifikant, werden sie in das Wörterbuch eingefügt. Der Algorithmus eignet sich hinsichtlich seines zeitlichen Aufwandes nicht für Sequenzmuster, die länger als 8 Zeichen sind, kann aber recht umfangreiche Eingabedaten, wie z.B. komplette Genome verarbeiten. Auf dem *MobyDick*-Prinzip basiert auch der VOCABULON-Algorithmus von Sabatti et al [Sab04]. Das Wörterbuch dieses Verfahrens verwaltet jedoch eine Variante von Gewichtsmatrizen, die zunächst einzelne Sequenzen repräsentieren (durch entsprechendes Setzen der Gewichte zugunsten eines bestimmten Zeichens in jeder Spalte). Der TEIRESIAS-Algorithmus von Rigoutsos et al [Rig98] agiert dagegen auf regulären Ausdrücken.

3.2.2 Stochastische Verfahren

Stochastischen Motivsuchverfahren liegt die Annahme zugrunde, dass die Eingabesequenzen, wie z.B. SELEX-Sequenzen von einem unbekannten stochastischen Modell erzeugt wurden. Positionen einer Sequenz, die einer TFBS angehören, wurden dabei von einer anderen Komponente des Modells erzeugt als nicht-gebundene Positionen. Die Aufgabe besteht nun darin, das Sequenzmodell zu finden, dass am Besten die Eingabesequenzen erzeugen kann. Insbesondere ist der Teil des Modells von Interesse, der die Erzeugung der vermuteten TFBS beschreibt.

Die Aufgabe, ein TFBS-Sequenzmodell zu identifizieren, wäre über eine einfache *Maximum-Likelihood*-Schätzung lösbar, wenn die Positionen der TFBS in den Eingabesequenzen bekannt wäre, was nicht der Fall ist. In dieser Situation unvollständigen Wissens bieten sich iterative Verfahren, schrittweise das erzeugende Sequenzmodell zu verbessern. Im Bereich der Motivsuche haben sich zwei Ansätze durchgesetzt: 1.) die Gibbs-*Sampling*-Methode und der 2.) Methoden, die auf dem *Expectation-Maximization*-Prinzip (EM) arbeiten.

Gibbs-Sampling-Verfahren. *Gibbs-Sampling* ist ein Algorithmus zum Erzeugen einer Sequenz von Stichprobenvektoren einer Menge von Zufallsvariablen, die gemäß einer gemeinsamen Verteilung verteilt sind. Die gemeinsame Verteilung ist unbekannt und soll mittels der Stichprobenvektorensequenz approximiert werden. Das Grundprinzip funktioniert gerade dann gut, wenn die gemeinsame Verteilung zwar unbekannt, die bedingten Verteilungen der Einzelvariablen (gegeben die aktuellen Werte der anderen Variablen) jedoch bekannt sind. In diesem Fall wird in jedem Schritt des Gibbs-Samplings zufällig eine Variable ausgewählt und von dieser zufällig ein Wert gemäß der bedingten Verteilung gezogen. Die sich ergebende Folge von Stichprobenvektoren kann durch eine

Markovkette beschrieben werden. Die stationäre Verteilung dieser Markovkette ist die gesuchte gemeinsame Verteilung der Variablen.

Übertragen auf das Problem der Motivsuche stellt sich das folgendermaßen dar [Law93]: Gegeben seien N DNA-Sequenzen $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N$ der Länge L, in denen jeweils genau eine Teilsequenz der Länge W gesucht ist, der TFBS. Eine PWM $M \in \mathbb{R}^{4\times W}$, dessen Gewichte m_{ij} (logarithmierte) Wahrscheinlichkeiten sind, beschreibt die Erzeugung dieser TFBS. Die restlichen Positionen werden von einer Hintergrundverteilung $p_{\mathtt{A}}$, $p_{\mathtt{C}}$, $p_{\mathtt{G}}$, $p_{\mathtt{T}}$ erzeugt. Das unvollständige Wissen sind die Startpositionen $z_n \in \{1, \ldots, L - W + 1\}$ für $1 \leq n \leq N$ der TFBS in den einzelnen Sequenzen.

Der Algorithmus wird mit zufälligen Werten für die z_n und mit zufälligen Gewichten m_{ij} initialisiert. Anschließend werden die folgenden Schritte fortwährend wiederholt, bis ein Abbruchkriterium erreicht ist:

1. Schätzschritt: Es wird eine zufällige Sequenz $\boldsymbol{s}_n$ ausgewählt. Die Gewichte m_{ij} und die Hintergrundwahrscheinlichkeiten werden unter Verwendung der aktuellen Startpositionen z_k für $1 \leq k \leq N$ und $k \neq n$ neu ML-geschätzt, ohne Berücksichtigung der Sequenz $\boldsymbol{s}_n$.
2. *Sampling*schritt: Die frisch geschätzte Matrix M wird verwendet, um in Sequenz $\boldsymbol{s}_n$ jede Position zu bewerten. Bezüglich dieser Bewertungen wird nun eine neue Startposition z_n gezogen.

Je stärker das PWM-Modell in Schritt 1 ist, desto sicherer ist das Ziehen einer günstigen TFBS-Position, und umgekehrt. Das *Gibbs-Sampling*-Verfahren wurde erstmals von Lawrence et al. [Law93] zur Motivsuche in Proteinen eingesetzt und seither in verschiedenen Arbeiten verfeinert und weiterentwickelt. Neuwald und Lawrence [Neu95] et al. passten die Originalmethode für DNA an. Roth et al. ermöglichen in ihrer Software *AignACE* mehrere TFBS in einzelnen Sequenzen [Rot98]. Die Möglichkeit, zwei oder mehrere Motive gleichzeitig in einer Menge von Sequenzen zu suchen, wurde auch von Stormo et al. unter dem Namen *Co-Bind* in [Guh01] sowie von Thijs et al. in [Thi02] vorgesehen.

Die Software *BioProspector* von Liu et al. [Liu01] kann auch mit unterbrochenen und palindromischen TFBS umgehen, und verwendet ein Hintergrundmodell höherer Ordnung. Ein weiterer Abkömmling ist *CompareProspector* [Liu04], der zusätzlich die Konserviertheit der mutmaßlichen TFBS bei dem *Sampling-Schritt* berücksichtigt.

Schließlich wurden von Narlikar et al. Gibbs-Sampling-Verfahren entwickelt, die a priori Informationen über den unbekannten Motivinstanzpositionen einsetzen, in ähnlicher Weise wie der in Kapitel 7 vorgestellte EM-basierte Ansatz [Nar06a, Nar07].

EM-Verfahren. Der einzige Unterschied zwischen dem *Gibbs-Sampling*-Prinzip und dem EM-Algorithmus ist, dass beim *Gibbs-Sampling* im ersten Schritt die unbekannten Daten von der aktuellen gemeinsamen Verteilung zufällig erzeugt werden (*sampling*), während beim EM-Algorithmus ein Erwartungswert für die unbekannten Daten

berechnet wird und dieser im zweiten Schritt zur Modellbildung verwendet wird. Da der EM-Algorithmus zentraler Gegenstand von Kapitel 7 ist, soll das Grundprinzip hier nur kurz erläutert werden.

Der EM-Algorithmus besteht aus der wiederholten Ausführung von zwei Schritten, dem E-Schritt und dem M-Schritt. Im E-Schritt (*expectation*) wird ein Erwartungswert der vollständigen Daten (Eingabedaten + unbekannte Daten) unter Verwendung des aktuellen Modells berechnet. Dieser Erwartungswert dient als Ersatz für die unbekannten vollständigen Daten und wird im M-Schritt (*maximization*) verwendet, um die Parameter des Modells zu finden, welche die erwarteten vollständigen Daten maximieren (klassische ML-Schätzung).

Übertragen auf das Motivsuchproblem und unter Verwendung der bei *Gibbs-Samplern* eingeführten Notation bedeutet dies, dass im t-ten E-Schritt Erwartungswerte für Zufallsvariablen Z_n berechnet werden, welche die TFBS-Anfangspositionen repräsentieren:

$$\boldsymbol{Z}^{(t)} = \underset{(Z \mid \boldsymbol{S}, M^{(t-1)})}{\mathbb{E}}[Z] \tag{3.9}$$

Für die Berechnung wird das vorherige PWM-Modell $M^{(t-1)}$ auf die Sequenzdaten $\boldsymbol{S}$ angewendet. Im M-Schritt wird das PWM-Modell $M^{(t)}$ gesucht, dass die gemeinsame Wahrscheinlichkeit $P(\boldsymbol{S}, \boldsymbol{Z}^{(t)})$ der vollständigen Daten maximiert:

$$M^{(t)} = \underset{M}{\operatorname{argmax}} \underset{(Z \mid \boldsymbol{S}, M^{(t-1)})}{\mathbb{E}}[\log P(X, Z \mid M)] \tag{3.10}$$

Lawrence et al., die auch bei der Anwendung des *Gibbs-Samplings* auf die Motivsuche federführend waren, stellten bereits 1990 einen ersten EM-Algorithmus zur Motivsuche vor [Law90], verlegten ihre Anstrengungen in der Folgezeit jedoch auf das *Gibbs-Sampling*. Die bekannteste Software, die EM-Motivsuche durchführt, ist *MEME*, dass von Bailey und Elkan 1995 veröffentlicht wurde und auch heute noch große Bedeutung in der Anwendungspraxis besitzt [Bai95b]. Es kombiniert den Originalansatz mit einer heuristischen Suche nach günstigen Anfangs-PWM-Modellen und ermöglicht die Suche von Motiven in drei Suchmodi (siehe Kapitel 7)

Aufbauend auf MEME wurden von Bailey und anderen Gruppen Anpassungen vorgeschlagen. ParaMEME von Grundy et al. ist eine parallelisierte Version von MEME samt einer Web-basierten Benutzeroberfläche [Gru96]. OrthoMEME [Pra04] von Prakash et al. nutzt als Verbesserungsmöglichkeit aus, dass den Eingabesequenzen zusätzlich ihre orthologen Sequenzen aus anderen Genomen hinzugefügt werden können.

3.3 Modellierung von TFBS-Modulen

Eine andere mögliche Folgerung aus der hohen Zahl vermeintlicher Falschvorhersagen von TFBS-Modellen ist, dass die Transkriptionsfaktoren eben tatsächlich rein zufällig

an allen Sequenzen binden, die entfernt ihren bevorzugten Bindungssequenzen ähneln. Die funktionstragende Bindung entsteht jedoch erst durch den Kontakt mit kooperierenden Faktoren bzw. mit dem RNA-Polymerase-II-Komplex. Der Eintritt dieser günstigen Konfiguration wäre dann ein Zufallsereignis, dessen Wahrscheinlichkeit sich einerseits durch entsprechend höhere Konzentrationen beteiligter Transkriptionsfaktoren und andererseits durch eine Nachbarschaft der TFBS der kooperierenden Faktoren erhöht. Eine Gruppe benachbarter und funktionell zusammenhängender TFBS heißt im folgenden *TFBS-Modul*. Neben akkurateren Beschreibungsansätzen der relevanten Eigenschaften einer einzelnen TFBS ist die Einbeziehung von Wissen über kooperierende Transkriptionsfaktoren und deren TFBS eine vielversprechende Möglichkeit, die wirklich biologisch aktiven TFBS eines Transkriptionsfaktors vorherzusagen. In diesem Abschnitt wird auf Arbeiten zu dieser Thematik eingegangen.

Bailey et al. unterteilen in ihrer Veröffentlichung [Bai03] die Palette der Verfahren zur Erkennung von TFBS-Modulen in drei Gruppen. Für den folgenden Überblick wird diese Kategorisierung aufgegriffen. Zunächst gibt es sogenannte *fensterbasierte* Verfahren, die in Unterabschnitt 3.3.1 vorgestellt werden. Eine weitere Gruppe von Verfahren setzt auf modular zusammengesetzte *Hidden Markov Modelle* und wird in Unterabschnitt 3.3.2 beleuchtet. Das in Abschnitt 5.5 eingeführte TFBS-Modul-Erkennungsverfahren gehört ebenfalls in diese Gruppe. Die dritte Gruppe, die so genannten *diskrimitativen* Verfahren basieren auf dem Lernen der Unterschiede von echten TFBS-Modulen und beliebigen anderen Sequenzen. Wichtige Vertreter dieser Gruppe werden in Unterabschnitt 3.3.3 genannt. Hierzu gehören auch Methoden, die regulative Sequenzen über Sequenzvergleiche mit orthologen Sequenzen zu identifizieren versuchen.

3.3.1 Fenster-basierte Verfahren

Das in [Ber02] vorgestellte Werkzeug *CIS-ANALYST* ist ein typischer Vertreter der fenster-basierten Suche von TFBS-Modulen. Zunächst durchsucht *CIS-ANALYST* eine Eingabesequenz nach einzelnen TFBS für eine gewünschte Menge von Transkriptionsfaktoren. Dafür wird die frei verfügbare Software *Patser* [Her99] verwendet, ein einfaches Hilfmittel zur Vorhersage von TFBS mit PWM-Modellen. Anschließend untersucht *CIS-ANALYST* alle Sequenzfenster einer vorgegebenen Größe, und markiert jene Fenster, die mindestens eine bestimmte Anzahl von TFBS-Vorhersagen enthalten. In einem letzten Schritt werden sich überlappende Fenster zusammengeführt und als Modulvorhersage ausgegeben.

Auch *MSCAN* [Joh03] durchsucht zunächst die Eingabesequenzen nach einzelnen TFBS, wobei jeweils ein p-Wert der Treffer berechnet bzw. geschätzt wird. Die p-Werte der Treffer eines Sequenzfensters werden nun zu einem Maß verrechnet, dass die statistische Signifikanz des Fensters widerspiegeln soll. Es ermöglicht eine Abschätzung, wie viele Sequenzabschnitte einer mindestens gleichhohen Bewertung in der Eingabe erwartet werden. Nur Fenster, deren Bewertung eine bestimmte Signifikanz erreicht, wird als

TFBS-Modul ausgegeben. Wie auch bei CIS-ANALYST werden überlappende Treffer vereinigt.

Während *CIS-ANALYST* und *MSCAN* für die Gültigkeit einer einzelnen TFBS-Vorhersage lediglich verlangen, dass diese in einem Sequenzfenster liegt, in dem gehäuft TFBS-Vorhersagen anzutreffen sind, können mit der Software *FastM* von Klingenhoff et al. konkretere Bedingungen für den Aufbau eines zu suchenden TFBS-Moduls festgelegt werden [Kli99]. Das schließt vor allem die relative Position und Orientierung bestimmter TFBS im Modul zueinander ein. Zudem kann das Vorhandensein einer bestimmten Teilmenge von TFBS gefordert werden. Die Abstände zwischen zwei benachbarten TFBS können als Intervall angegeben werden. Mit einem zweiten Programm, *ModelInspector* können nun TFBS-Module gesucht werden, die zuvor mit *FastM* definiert wurden. Da auch hier überlappende Fenster der gemäß Moduldefinition nötigen Breite nach einzelnen TFBS durchsucht werden und für diese Fenster die Erfülltheit der Modulbedingungen überprüft wird, kann es dennoch zu den rein fensterbasierten Ansätzen gezählt werden. Die Idee, komplexere Bedingungen an die Gestalt eines TFBS-Moduls zu stellen, wird auch in Kapitel 6 dieser Dissertation aufgegriffen, jedoch in probabilistischer Weise.

FlyEncancer ist eine speziell für die genomweite Suche von Sequenzmotivtreffern in *Drosophila* entwickelte Web-Anwendung und arbeitet mit einfachen Consensussequenzen [Mar02]. Ausgegeben werden alle Sequenzintervalle fester Breite, die mindestens eine bestimmte Anzahl von Treffern aufweisen.

Crowley et al. stellten in [Cro97] einen TFBS-Modulerkennungsansatz vor, der Gemeinsamkeiten mit dem in Kapitel 6 vorgestellten System zur Berücksichtigung von TFBS-Kontexten bei der Vorhersage besitzt. Zunächst werden auf fensterbasierte Weise TFBS-Vorhersagen auf einer DNA-Sequenz gemacht. Für jeden Einzeltreffer wird nun in einem Katalog nachgeschlagen, ob die Faktoren der benachbarten TFBS funktionell etwas mit dem Treffer zu tun haben könnten. In Bezug auf diese Auswertung werden nachträglich die TFBS-Vorhersagen in *falsche* oder *richtige* Treffer eingeteilt.

Einen interessanten Weg, Einzel-TFBS-Vorhersagen zu Modulen zusammenzufügen, stellten Pickert et al. vor [Pic98]. Auf dem durchsuchten Sequenzintervall werden *Cluster* gebildet, und jede Einzelvorhersage wird dem *Cluster* zugeordnet, dem es am Nächsten liegt (Abstandsmaß: Basenpaare). Dabei darf eine gewisse Entfernung nicht überschritten werden.

Schones et al. versuchen mit ihrer Software *MODSTORM* die statistische Signifikanz einer Häufung von TFBS-Vorhersagen zu bestimmen [Sch07]. Dazu nehmen sie ein stochastisches Modell zur Verteilung von TFBS innerhalb des Genoms an und berechnen, wie wahrscheinlich das Auftreten von k Treffern in einem gewissen Abstand in einer zufälligen Sequenz ist. Gemeinsam mit den statistischen Signifikanzen der Einzeltreffer kann jedem potentiellen Modul eine Sigifikanz zugewiesen werden. Die Forscher geben auch eine Strategie an, mit der maximale TFBS-Module aus signifikanten kleineren Modulen, die in dem maximalen enthalten sind, bestimmt werden können. Auch Wagner et al. stellen ein Verfahren zur Signifikanzbestimmung von TFBS-Modulen vor [Wag99]. In

ihrem Ansatz werden einzelne TFBS als *Poisson*-verteilt in einer Sequenz angenommen, TFBS-Module gemäß einer *Pearson*-Verteilung.

Die beiden Programme, *ModuleSearcher* und *ModuleScanner*, sind Teil des Softwarepakets *TOUCAN* von Aert et al. [Aer03]. Während sich hinter *ModuleScanner* trotz etwas komplexerer Bewertungsfunktion eine normale fensterbasierte Suche von TFBS-Modulen verbirgt, bietet *ModuleSearcher* einen Algorithmus zur Suche eines optimlen Modells für TFBS-Module auf Grundlage einer Menge koregulierter Gene an. Dazu wird der A^*-Algorithmus verwendet. Die Reihenfolge und Orientierung der einzelnen TFBS eines Moduls bleibt jedoch unberücksichtigt.

3.3.2 HMM-basierte Verfahren.

Frith et al. veröffentlichten 2001 die erste Arbeit, in der ein HMM zur Modellierung und Erkennung von TFBS-Modulen verwendet wurde [Fri01]. Dieses Modell, dass in der Software *Cister* verwendet wird, Es wird in Abschnitt 5.5 genauer beleuchtet, da die Struktur der dort eingeführten HMM mit Bayesschen-Netz-Zuständen in Anlehnung an dieses Modell entwickelt wurde.

Die Grundidee hierbei ist es, PWM-Modelle gemäß den Ausführungen auf Seite 39 als spezielle HMM-Modelle aufzufassen, und diese speziellen HMM in Parallelschaltung in einem größeren HMM zu integrieren, welches dann gültige TFBS-Module modelliert. Neben den PWM-Modellen enthält das modulare System Zustände für die Erzeugung des Sequenzbereiches zwischen zwei TFBS eines Moduls und für den Sequenzbereich zwischen zwei TFBS-Modulen. Die Suche nach TFBS-Modulen besteht in der Berechnung des optimalen Zustandspfades durch das HMM. In einem jüngeren Programm, *Comet*, vereinfachen Frith et al. die HMM-Struktur, um die Bestimmung einer statistischen Signifikanz für TFBS-Modulvorhersagen mathematisch handhabbar zu machen [Fri02]. Das mathematische Modell zur Signifikanzbestimmung ist an [Wag99] angelehnt. Sie können jedoch auf diese Weise die Ergebnisse von *Cister* nicht erreichen.

Bailey et al. verwenden in [Bai03] nahezu das gleiche HMM wie Frith et al. in [Fri01]. Ihr Verfahren ist jedoch für die Suche in einer Datenbank von Sequenzen optimiert, so dass es möglich ist, die Anzahl von Modul-Treffern statistisch zu bewerten und eine Rangliste bezüglich der Stärke der Treffer auszugeben. Sie verwenden mit dem Viterbi-Algorithmus ein anderes Dekodierungsverfahren und entwickeln einen auf p-Werten einzelner TFBS-Vorhersagen aufgebautes Bewertungsschema für TFBS-Module.

Im Jahre 2000 veröffentlichten Ohler et al. [Ohl00] einen Ansatz zur Erkennung von eukaryontischen Kernpromotoren mit Hilfe spezieller HMM, den *Stochastischen Segment-Modellen (SSM)*. Mit Hilfe dieser Modelle kann die Längenmodellierung der nichtfunktionalen Zwischenräume flexibler behandelt werden als mit gewöhnlichen HMM.

Eigentlich nicht als HMM entworfen, argumentieren Rajewki et al. in einem Zusatzdokument zu [Raj02], dass das Modell der Software *Ahab* zur Segmentierung einer DNA-Sequenz in Hintergrundsequenzstücke und in TFBS im Prinzip identisch mit einem HMM ist. Dessen Übergangswahrscheinlichkeiten werden jedoch mit einem Gradientenabstieg optimiert und müssen deshalb im Gegensatz zu Frith et al. und Bailey et al. nicht vom Anwender definiert werden. Sinha et al. [Sin03] setzen ebenfalls auf das Prinzip von *Ahab*, die Übergangswahrscheinlichkeiten eines HMM zu lernen. Sie nutzen jedoch zusätzliche Möglichkeiten, die ein HMM-Ansatz bildet, nämlich die Berücksichtigung von Reihenfolgen der TFBS-Vorhersagen oder die Berücksichtigung von Präferenzen eines TFBS-Typs, auf einem bestimmten DNA-Strang zu liegen.

3.3.3 Diskrimitative Verfahren

Die bekannteste Veröffentlichung bezüglich dieser Gruppe von Verfahren stammt von Wasserman und Fickett [Was98]. Darin verwenden sie ein logisches Regressionsmodell, um TFBS-Module aus muskelspezifischen Genen von nichtfunktionalen Sequenzen zu unterscheiden. Als Merkmale verwendet das System evolutionäre Konserviertheit sowie PWM-Bewertungsprofile von muskelspezifischen Transkriptionsfaktoren in der untersuchten Sequenz. Der Datensatz, bestehend aus 27 muskelspezifischen Enhancern gilt seither als wichtiger Benchmark für Verfahren zur Erkennung von TFBS-Modulen.

Phylogenetic Footprinting. Hierbei handelt es sich um eine Technik, TFBS oder TFBS-Module in einer DNA-Sequenz durch den Vergleich dieser Sequenz mit orthologen Sequenzen anderer Arten zu identifizieren. Die Grundidee dieser Technik ist, dass funktionale, regulative Sequenzen, genau wie codierende DNA-Bereiche, einem größeren evolutionären Druck ausgesetzt sind als nichtfunktionale Sequenzbereiche. Außerdem wird angenommen, dass die Eigenschaften einer TFBS einschließlich des Mechanismus der Protein-DNA-Bindung ebenfalls stark evolutionär konserviert sind [UV03].

Im Allgemeinen müssen beim *Phylogenetic Footprinting* drei Aufgaben bearbeitet werden, die jede für sich genommen einige Schwierigkeiten bereiten kann. Zunächst müssen Arten ausgewählt werden, deren orthologe Sequenzen mit dem zu untersuchenden Genom verglichen werden können. Dabei ist der evolutionäre Abstand zwischen zwei Arten von Bedeutung. Die Sequenzbereiche zwischen zwei Genen, also potentielle regulierende Sequenzen, können insgesamt noch zu stark konserviert sein, dass die wichtigen TFBS-Bereiche sich nicht genügend abheben. Bei zu weit entfernten Arten besteht häufig das Problem, dass die Promotorregionen von orthologen Genen so unähnlich sind, dass gewöhnliche Alignmentverfahren nicht mehr anwendbar sind. Im Idealfall stehen gleich mehrere Vergleichsarten verschiedener evolutionärer Abstände zur Verfügung, wie z.B. im Falle der zwölf komplett sequenzierten *Drosophila*-Genome [Heg07].

Die zweite Aufgabe besteht darin, in den ausgewählten Vergleichsgenomen die orthologen Sequenzen zu der Sequenz zu finden, in der TFBS-Module gesucht werden sollen.

In selten günstigen Fällen sind alle zu vergleichenden Genome hochgradig annotiert, im Normalfall müssen zunächst die orthologen codierenden Sequenzen in einer Datenbannk mit *BLAST* gesucht werden, um anschließend die Bereiche oberhalb der codierenden Bereiche miteinander zu alignieren. Da es im Laufe der Evolution häufig zu Genduplikationen kommt, ist die Eindeutigkeit einer solchen Suche nicht immer garantiert und die Frage, welche regulativen Bereiche miteinander verglichen werden sollen, nicht einfach zu beantworten [Pra05].

Konnten orthologe Sequenzen für alle zu vergleichenden Genome identifiziert werden, müssen die nicht-codierenden Sequenzen oberhalb dieser Gene miteinander verglichen werden. Dazu kommen Verfahren für multiple Aligments zum Einsatz, angefangen von globalen multiplen Aligmentalgorithmen wie *ClustalW* [Che03], über globale Alignmentverfahren, die Ankerpunkte (die hochkonservierten Bereiche der Gene oberhalb und unterhalb der relierenden Bereiche) verwenden wie *MAVID* [Bra04] bis hin zu ankergestützten *lokalen* Aligmentalgorithmen wie *DIALIGN* [Mor98]. Die meisten dieser Verfahren sind jedoch für den Vergleich hochkonservierter codierender Bereiche entwickelt worden und nicht besonders gut geeignet, die wenigen und sehr kurzen konservierten TFBS-Module in ansonsten komplett unähnlichen Sequenzen zu finden. So sind einige der Verfahren problemlos in der Lage, auch in nichtorthologen Promotorbereichen konservierte TFBS-Module zu finden [Pra05]. In wenigen Fällen liegen bereits globale, genomweite Alignments zwischen zwei oder mehreren Arten vor, z.B. ein Mensch-Maus-Alignment, ein Mensch-Maus-Ratte-Alignment oder ein Alignment von zwölf Drosophila-Arten [UV03]. Diese können manuell in grafischen Web-Oberflächen wie dem UCSC-Browser durchsucht werden

Besonders die soeben beschriebene Aufgabe erschwert die völlig automatisierte Suche noch konservierten Bereichen orthologer, nicht-kodierender Sequenzbereiche, denn viele Genome sind nur unzureichend annotiert und die Suche der orthologen Gene mit BLAST nicht immer zuverlässig. Dennoch gibt es eine Reihe von Softwarepaketen, die sich allen Teilbereichen des *Phylogenetic Footprinting* widmen.

Bei Meng et al. [Men06] wird der phylogenetische Vergleich auf Bindingsite-Ebene durchgeführt. Zunächst werden in zwei orthologen Sequenzen TFBS mit Hilfe von TRANSFAC-PWM gesucht. Für jede Position und jede PWM wird ein p-Wert bestimmt. Die beiden p-Wert-Kurven jeweils einer PWM werden aschließend geglättet und über ein inneres Produkt miteinander verrechnet. Die entstandene Kurve korreliert mit der evolutionären Konserviertheit der mutmaßlichen TFBS. Bei *rVista* handelt es sich um ein einfaches Werkzeug zur Filterung von TFBS-Vorhersagen auf Grundlage eines globalen Alignments der entsprechenden orthologen Sequenzen [Loo02]. Das Alignment muss dem Anwender bereits bekannt sein.

Blanco et al. [Bla06] gehen bei der komparativen Analyse regulativer Sequenzen einen etwas anderen Weg. Sie verwenden zunächst einfache Sequenzmodelle, um in beiden Orthologen TFBS vorherzusagen. Es ensteht jeweils eine Liste von TFBS-Vorhersagen. Anschließend bilden sie ein Alignment dieser Listen von TFBS-Vorhersagen. Auf diese Weise abstrahieren sie von üblichen Alignments, bei denen die zueinander passenden

TFBS auch in relativer Nähe zueinander liegen müssen. Auf der anderen Seite kann dieser Ansatz Vertauschungen zweier TFBS im Vergleich zur orthologen Sequenz nicht abbilden.

Kapitel 4

Maschinelles Lernen und Schließen mit Bayesschen Netzen

In den Modellierungsansätzen, die in Kapitel 5 und Kapitel 6 entwickelt werden, werden Bayessche Netze eingesetzt, um Transkriptionsfaktorbindungsstellen und Module dieser Bindungsstellen zu modellieren und zu detektieren. Zuvor soll dieses Kapitel eine Einführung in dieses für die Dissertation zentrale Konzept bieten.

Bayessche Netze sind grafische Repräsentationen einer gemeinsamen Wahrscheinlichkeitsverteilung einer Menge von Zufallsvariablen. Eine Reihe vorteilhafter Eigenschaften haben sie zu einem bedeutenden Modellierungsansatz in vielen Bereichen der Künstlichen Intelligenz gemacht. So werden sie erfolgreich in Expertensystemen, Diagnosesystemen, Entscheidungssystemen sowie bei der Modellierung und Klassifikation von Mustern eingesetzt.

Die Zufallsvariablen eines Bayesschen Netzes können verschiedenartige Wertemengen haben. Bayessche Netze eignen sich deshalb besonders zur Modellierung und Verarbeitung heterogener Daten. Eine wesentliche Eigenschaft Bayesscher Netze ist die Fähigkeit, stochastische Abhängigkeiten zwischen Variablen durch bedingte Wahrscheinlichkeitsverteilungen darzustellen. Diese Abhängigkeiten ergeben sich häufig aufgrund kausaler Zusammenhänge zwischen zwei Variablen oder aufgrund statistischer Analysen einer Menge von Variableninstantiierungen in einem Lernprozess. Die gemeinsame Verteilung der Zufallsvariablen hält ein Bayessches Netz in faktorisierter Form bereit, seine Werte ergeben sich aus dem Produkt der (bedingten) Wahrscheinlichkeiten für die Belegungen der Einzelvariablen. Vor allem diese Eigenschaft hat weitreichende Konsequenzen für das robuste Lernen Bayesscher Netze aus Daten begrenzten Umfangs und für ihre Anwendung zum probabilistischen Schließen in Expertensystemen. Die Anzahl der Parameter, die zur Beschreibung einer gemeinsamen Verteilung benötigt werden, ist verschwindend klein gegenüber einer tabellenartigen Repräsentation dieser Verteilung. Die faktorisierte Darstellung der gemeinsamen Verteilung gestattet zusammen mit den vorhandenen effizienten Algorithmen die Beantwortung beliebiger probabilistischer Anfragen in Form bedingter Randverteilungen einer Teilmenge von Variablen. Aufgrund dieser Möglichkeiten probabilistischen Schließens empfehlen sich Bayessche Netze außerdem für den Umgang mit unvollständigen und ungewissen Daten.

Zu guter Letzt haben die "*Soft skills*" der Bayesschen Netze, nämlich ihr intuitiv einfacher Aufbau, der es dem Menschen ermöglicht, die Semantik der im Netz dargestellten Zusammenhänge zwischen den Variablen leicht zu erfassen, zu ihrem Erfolg beigetragen.

Dieses Kapitel ist wie folgt aufgebaut. In Abschnitt 4.1 werden Bayessche Netze formal definiert. Abschnitt 4.2 beleuchtet die Anwendung von Bayesschen Netzen als Datenstrukturen für probabilistisches Schließen. Abschnitt 4.3 beschäftigt sich mit speziellen Bayesschen Netzen für die Klassifikation von Mustern. Abschnitt 4.4 stellt Lernverfahren für Bayessche Netze vor.

4.1 Bayessche Netze

Es werden im Folgenden Elemente einer endlichen Menge $\boldsymbol{U}$ diskreter Zufallsvariablen mit endlichen Wertemengen betrachtet. Einfache Großbuchstaben (z.B. X, Y, Z) werden für einzelne Variablen und Kleinbuchstaben (z.B. x, y, z) für spezifische Werte dieser Variablen verwendet. Der Wertebereich einer Variable X heißt D_X. Teilmengen von $\boldsymbol{U}$ werden als fettgedruckte Großbuchstaben dargestellt (z.B. $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$), fettgedruckte Kleinbuchstaben stehen abkürzend für Belegungen der Variablen einer Menge $\boldsymbol{X} \in \boldsymbol{U}$. Weiterhin wird für eine Variablenmenge $\boldsymbol{Z} \subseteq \boldsymbol{U}$ die vereinfachende Schreibweise $P(\boldsymbol{z})$ für die Wahrscheinlichkeit des Zufallsereignisses $P(\boldsymbol{Z} = \boldsymbol{z})$ verwendet.

DEFINITION 4.1: *Ein* **Bayessches Netz** $\mathcal{B}$ *ist ein Paar* (G, Θ)*, für das gilt:*

1. $G = (\boldsymbol{U}, \boldsymbol{E})$ *ist ein gerichteter, azyklischer Graph (DAG für engl.* directed acyclic graph*). Die Knoten* $X \in \boldsymbol{U}$ *sind diskrete Zufallsvariablen mit endlichen Wertemengen* D_X*. Die Kantenmenge* $\boldsymbol{E}$ *heißt* **Struktur** *von* $\mathcal{B}$*. Für eine Variable* $X \in \boldsymbol{U}$ *bezeichnet* $\boldsymbol{\Pi}_X$ *die Menge ihrer Elternvariablen.*
2. $\boldsymbol{\Theta}$ *bezeichnet die Gesamtheit aller Parameter in* $\mathcal{B}$*. Diese Gesamtheit gliedert sich in je einen Parametersatz* Θ_i *für jede Zufallsvariable* $X_i \in \boldsymbol{U}$*. Ein Parametersatz* Θ_i *enthält bedingte Wahrscheinlichkeiten*

$$\theta_{x_i|\boldsymbol{\pi}_{X_i}} = P(X_i = x_i \,|\, \boldsymbol{\Pi}_{X_i} = \boldsymbol{\pi}_{X_i}) \tag{4.1}$$

   für jeden möglichen Wert $x_i \in D_{X_i}$ *einer Variable* X_i *und jede mögliche Belegung* $\boldsymbol{\pi}_{X_i}$ *der Elternvariablen in* $\boldsymbol{\Pi}_{X_i}$*.*

Der Graph eines Bayesschen Netzes definiert über seine Struktur Abhängigkeiten zwischen den Zufallsvariablen $X \in \boldsymbol{U}$. Diese Abhängigkeiten werden durch die bedingten Wahrscheinlichkeiten $\theta_{x_i|\boldsymbol{\pi}_{X_i}}$ quantifiziert. Hat eine Zufallsvariable X_i keine Eltern in G, so enthält Θ_i unbedingte Wahrscheinlichkeiten θ_{x_i} für alle Werte $x_i \in D_{X_i}$.

Implizit werden durch die Struktur und die dadurch gegebene Dimensionierung der Parametergesamtheit Unabhängigkeitsannahmen getroffen. So ist eine Variable X_i bedingt unabhängig von allen seinen nicht-Eltern $\boldsymbol{U} \setminus \boldsymbol{\Pi}_{X_i}$ unter gegebenen Eltern in $\boldsymbol{\Pi}_{X_i}$:

$$\begin{aligned} P(x_i \,|\, \boldsymbol{\pi}_{X_i},\, y) &= P(x_i \,|\, \boldsymbol{\pi}_{X_i}) && (4.2)\\ &= \theta_{x_i|\boldsymbol{\pi}_{X_i}} && (4.3) \end{aligned}$$

für alle Variablen $Y \in \boldsymbol{U} \setminus \boldsymbol{\Pi}_{X_i}$ und ihren Werten $y \in D_Y$. Diese wichtige Eigenschaft von Bayesschen Netzen heißt *Markov-Eigenschaft*.

Ein Bayessches Netz $\mathcal{B}$ repräsentiert eine gemeinsame Wahrscheinlichkeitsverteilung $P_{\mathcal{B}}(\,\cdot\,)$ über seinen Zufallsvariablen $\boldsymbol{U}$. Durch Anwendung der impliziten Unabhängigkeitsannahmen auf die Kettenregel für gemeinsame Wahrscheinlichkeiten,

$$P(x_1,\, \ldots,\, x_n) = P(x_1) \prod_{i=1}^{n} P(x_i \,|\, x_1,\, \ldots,\, x_{i-1}) \tag{4.4}$$

bei entsprechender Ordnung der Variablen [Pea88] ergibt sich eine faktorisierte Darstellung dieser Verteilung:

$$\begin{aligned} P_{\mathcal{B}}(x_1,\, \ldots,\, x_n) &= \prod_{i=1}^{n} P_{\mathcal{B}}(x_i \,|\, \boldsymbol{\pi}_{X_i}) && (4.5)\\ &= \prod_{i=1}^{n} \theta_{x_i \,|\, \boldsymbol{\pi}_{X_i}}. && (4.6) \end{aligned}$$

Abbildung 4.1 zeigt ein Bayessches Netz, das in der Fachliteratur häufig als einführendes Beispiel verwendet wird.

An dieser Stelle sei darauf verwiesen, dass Bayessche Netze in der Fachliteratur meist über die Erfülltheit der Markoveigenschaft einer gemeinsamen Verteilung P und eines DAG G definiert werden [Nea03, Pea88]. In [Nea03] ist beispielsweise ein Bayessches Netz ein geordnetes Paar (G, P), bestehend aus einem DAG G und einer gemeinsamen Wahrscheinlichkeitsverteilung P, für dass die Markov-Eigenschaft gilt. Die obige Definition betont dagegen die Konstruktion eines Bayesschen Netzes und leitet anschließend die Markov-Eigenschaft daraus ab.

4.2 Schließen mit Bayesschen Netzen

Bayessche Netze werden häufig zum *probabilistischen Schließen* verwendet. Im Falle von zwei Variablen X und Y wird darunter die Anwendung des Bayes-Theorems

$$P(x \,|\, y) = \frac{P(y \,|\, x) \cdot P(x)}{P(y)} \tag{4.7}$$

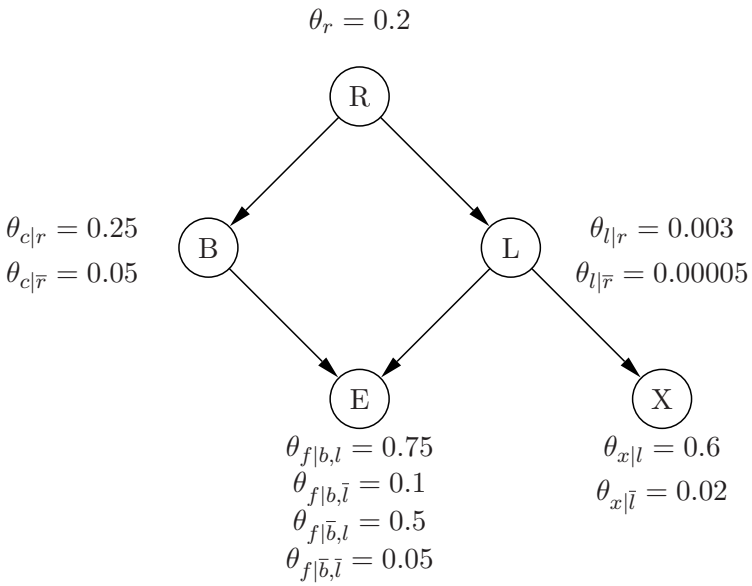


Abbildung 4.1: Beispielhaftes Bayessches Netz zur Differentialdiagnose von Lungenkrebs. Die Variablen stehen für Raucher (R), Lungenkrebs (L), Bronchitis (B), ständige Erschöpfung (E) und positiver Röntgenbefund (X). Alle Variablen besitzen je zwei Werte, z.B. r =Raucher und $\overline{r}$ =Nichtraucher. Eine Anwendungsmöglichkeit ist die Berechnung der Wahrscheinlichkeit dafür, dass eine Person Lungenkrebs hat, wenn diese Raucher ist und einen Röntgenbefund hat: $P(L = l \,|\, r, x)$.

verstanden, also die Berechnung der möglicherweise unbekannten Wahrscheinlichkeit $P(X = x \,|\, Y = y)$ aus möglicherweise vorhandenen Werten für $P(Y = y \,|\, X = x)$ (siehe auch das Beispiel in Abbildung 4.1).

Übertragen auf ein Bayesches Netz $\mathcal{B}$ mit Zufallsvariablen $\boldsymbol{U}$ besteht die Aufgabe darin, Anfragen der Form $P_{\mathcal{B}}(\boldsymbol{X} = \boldsymbol{x} \,|\, \boldsymbol{Y} = \boldsymbol{y})$ für disjunkte Variablenmengen $\boldsymbol{X}, \boldsymbol{Y} \subseteq \boldsymbol{U}$ zu beantworten. Die Belegung $\boldsymbol{y}$ der Variablenmenge $\boldsymbol{Y}$ sind dabei *Beobachtungen* bzw. Fakten, die verwendet werden sollen, um die Wahrscheinlichkeit einer konkreten Belegung der restlichen Variablen $X \in \boldsymbol{X}$ abzuleiten.

Die triviale Herangehensweise über die Formel für bedingte Wahrscheinlichkeiten,

$$P_{\mathcal{B}}(\boldsymbol{x} \,|\, \boldsymbol{y}) = \frac{P_{\mathcal{B}}(\boldsymbol{x}, \boldsymbol{y})}{P_{\mathcal{B}}(\boldsymbol{y})} \tag{4.8}$$

$$= \sum_{\boldsymbol{V} = \boldsymbol{U} \setminus (\boldsymbol{X} \cup \boldsymbol{Y})} \frac{P_{\mathcal{B}}(\boldsymbol{v})}{\sum_{\boldsymbol{Z} = \boldsymbol{U} \setminus \boldsymbol{X}} P_{\mathcal{B}}(\boldsymbol{z})}, \tag{4.9}$$

ist nicht praktizierbar, da der Aufwand für die Marginalisierung über mehrere Variablen exponentiell in der Anzahl der Variablen wächst. Obwohl das Problem des probabilistischen Schließens NP-vollständig ist [Coo90], ermöglichen die sich aus der Erfülltheit der Markoveigenschaft ergebenden Unabhängigkeitsannahmen zwischen den Variablen

eines Bayesschen Netzes meist ein wesentlich schnelleres Vorgehen zur Beantwortung dieser Anfragen. Besonders für bestimmte Klassen von Bayesschen Netzen, für deren Struktur gewissen Einschränkungen gelten, gibt es exakte und gleichzeitig akzeptabel schnelle und Speicherplatz schonende Algorithmen. Für den allgemeinen Fall kann zudem auf approximierende Algorithmen zurückgegriffen werden, deren Genauigkeit meist völlig ausreichend im jeweiligen Anwendungsfall ist. Im Folgenden sollen die wichtigsten Algorithmen beider Gruppen kurz angesprochen werden.

Exakte Algorithmen. Ein wichtiger Vertreter dieser Algorithmenklasse ist Pearls *message-passing*-Algorithmus [Pea88]. In seiner ursprünglichen Form ist dieser Algorithmus für Bayessche Netze geeignet, deren Graphen *Wurzelbäume* sind. Für die durch $\mathcal{B}$ repräsentierte gemeinsame Verteilung $P_{\mathcal{B}}$ der Variablen bedeutet das zum Einen, dass die Wurzelvariable bedingt unabhängig von allen anderen Variablen ist und zum Anderen, dass alle übrigen Variablen bedingt abhängig von nur einer weiteren Variablen, der *Elternvariablen*, bezüglich der Graphstruktur sind.

Der Algorithmus von Pearl geht zunächst von einem Bayesschen Netz aus, dessen Variablen alle unbelegt sind. Da noch keine Beobachtungen vorliegen, wird für jede Variable X die a priori Verteilung $P(X = x)$ berechnet. Jede Variable X benötigt hierfür lediglich die Information über die a priori Verteilung ihrer Elternvariablen, die als Nachricht an X gesandt wird.

Da das Prinzip der Nachrichtenübermittlung zwischen den Variablen zentral für eine Reihe weiterer Schließalgorithmen ist, soll es an dieser Stelle anhand des sehr einfachen Bayesschen Netzes auf Abbildung 4.2 nachvollzogen werden. Das dort abgebildete Netz ist ein Wurzelbaum mit Wurzel W. Jede Variable A hat in diesem Beispiel zwei mögliche Werte, die jeweils mit a_1 und a_2 bezeichnet werden. Da ohne Elternknoten, liegt für die Wurzel W bereits die a priori Verteilung vor. Die beiden Werte $P(w_1)$ und $P(w_2)$ werden als Nachricht an die beiden Variablen X und Y gesandt, um mittels der Formel der *totalen Wahrscheinlichkeit* die a priori Verteilungen dieser Variablen berechnen zu können. Die Wahrscheinlichkeit $P(y_1)$ berechnet sich beispielsweise vermöge

$$P(y_1) = P(y_1 \mid w_1)P(w_1) + P(y_1 \mid w_2)P(w_2). \tag{4.10}$$

Mit den anderen Variablen wird unter Verwendung der a priori Wahrscheinlichkeiten der Vorgängervariablen analog verfahren.

Um die a priori-Verteilung aller Variablen zu berechnen, müssen ausgehend von der Wurzel W die schon berechneten Wahrscheinlichkeiten einer jeden Variablen an ihre Nachfolgervariablen übertragen werden. Diese Richtung der Nachrichtenübertragung heißt *Vorwärtspropagation*. Für die Initialisierungsphase des Pearl'schen Alogorithmus in Wurzelbäumen wird lediglich die Vorwärtspropagation verwendet.

Das Instantiieren der Variablen U mit dem Wert u_2, d.h., das Beobachten des Wertes u_2, ist gleichbedeutend mit der Veränderung der a priori Wahrscheinlichkeiten von U

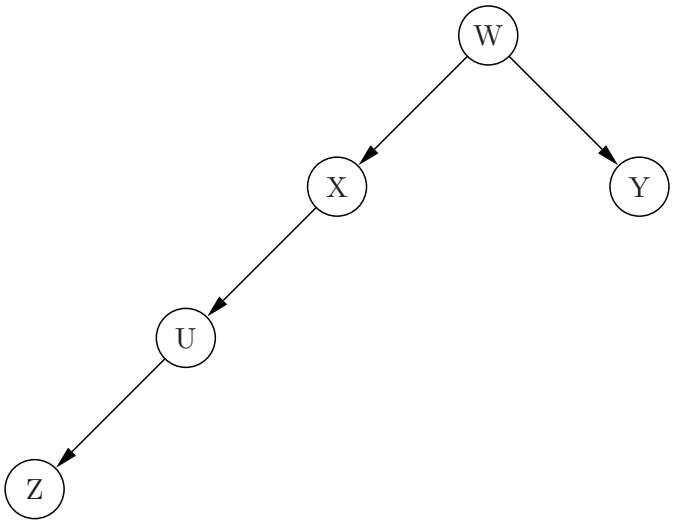


Abbildung 4.2: Beispiel für probabilistisches Schließen in Bayesschen Netzen. Erläuterung im Text.

hin zum sicheren Eintritt von u_2:

$$P(u_2) = 1. \tag{4.11}$$

Diese Anpassung muss nun allen anderen Variablen bekannt gemacht werden, um anschließend für jede Variable X die bedingten Wahrscheinlichkeiten $P(x_i \mid u_2)$ ihrer Werte bei Beobachtung von u_2 zu kennen.

Für die Nachfolgervariablen von U wird dafür, wie bei der Initialisierung, Vorwärtspropagation und die Formel der totalen Wahrscheinlichkeit verwendet. Für die Berechnung der bedingten Wahrscheinlichkeit des Elternknoten $P(x_1 \mid u_2)$ und $P(x_2 \mid u_2)$ wird die Beobachtung $P(u_2) = 1$ via *Rückwärtspropagation* an X gesendet und anschließend das Bayestheorem angewendet:

$$P(x_1 \mid u_2) \quad = \quad \frac{P(u_2 \mid x_1)P(x_1)}{P(u_2)} \tag{4.12}$$

Analog wird das Bayestheorem auch für W, die Elternvariable von X, angewendet, um die bedingte Wahrscheinlichkeit gegeben der Beobachtung u_2 zu erhalten:

$$P(w_1 \mid u_2) \quad = \quad \frac{P(u_2 \mid w_1)P(w_1)}{P(u_2)} \tag{4.13}$$

Die Berechnung kann zunächst nicht durchgeführt werden, da die Wahrscheinlichkeit $P(u_2 \mid w_1)$ nicht bekannt ist. Die fehlende Wahrscheinlichkeit kann jedoch wieder mit Hilfe der Vorwärtspropagation ermittelt werden:

$$P(u_2 \mid w_1) \quad = \quad P(u_2 \mid x_1)P(x_1 \mid w_1) + P(u_2 \mid x_2)P(x_2 \mid w_1). \tag{4.14}$$

Nach diesem Prinzip des Austausches von Nachrichten zwischen den Variablen des Bayesschen Netzes kann jede Anfrage nach der Wahrscheinlichkeit eines beliebigen Wertes einer beliebigen Variablen bei einer gegebenen Menge von Beobachtungen beantwortet werden. Der Algorithmus von Pearl gibt ein möglichst effektives Schema für die Reihenfolge der Berechnungen vor.

Durch einfache Anpassungen des Algorithmus ist auch probabilistisches Schließen in *einfach verbundenen* Bayesschen Netzen möglich. Ein Netz heißt einfach verbunden, falls es zwischen je zwei Knoten höchstens einen gerichteten Pfad gibt. Im Unterschied zu Bäumen können Variablen in einfach verbundenen Netzen mehrere Elternvariablen besitzen.

Für mehrfach verbundene Bayessche Netze kann der Pear'sche Algorithmus unter Verwendung einer Methode eingesetzt werden, die *conditioning* genannt wird. Auch hier soll das Grundprinzip anhand eines einfachen Beispiels erläutert werden. Das dazugehörige Bayessche Netz ist in Abbildung 4.3 dargestellt. Dieses Netz ist nicht einfach verbunden, da es zwischen den Variablen X und U zwei verschiedene Pfade gibt. Würde jedoch X entfernt werden, wäre das resultierende Netz einfach verbunden. Die Idee des *conditioning* besteht darin, durch Löschen von X zwei neue Bayessche Netze herzustellen, die sich dadurch unterscheiden, dass die a priori Verteilungen für die entstehenden Wurzeln Y und Z jeweils unter Annahme eines der Werte von X gestellt werden. Demnach gilt für das erste abgeleitete Netz:

$$P(y_i) := P(y_i \,|\, x_1) \tag{4.15}$$
$$P(z_i) := P(z_i \,|\, x_1), \tag{4.16}$$

für das zweite Netz dagegen

$$P(y_i) := P(y_i \,|\, x_2) \tag{4.17}$$
$$P(z_i) := P(z_i \,|\, x_2). \tag{4.18}$$

Das Schließen im ursprünglichen Netz läuft nun stellvertretend in den beiden daraus abgeleiteten Netzen. In größeren Netzen ist unter Umständen das Löschen mehrerer Variablen nötig. Während das Vorgehen im Beispiel dadurch vereinfacht wird, dass die gelöschte Variable X eine Wurzel ist, können solche Variablen nicht immer gefunden werden, um ein Anfrageproblem für ein Bayessches Netz zu lösen. Suermondt und Cooper vergleichen dazu in [Sue90] verschiedene Kriterien, die für zu löschende Variablen gelten müssen. Sie beweisen zudem, dass das Problem der Bestimmung einer minimalen Menge zu löschender Knoten NP-vollständig ist. Müssen in großen Netzen, deren Variablen zwei verschiedene Werte haben können, k Variablen gelöscht werden, bedeutet dies, dass stellvertretend in $\Theta(2^k)$ abgeleiteten Netzen der Pearl'sche Algorithmus angewendet werden muss, um jede beliebige Anfrage zu bearbeiten. In solchen Fällen ist die Technik des *conditioning* selbst bei bekannter minimaler zu löschender Variablenmenge nicht mehr mit annehmbarem Zeit- und Speicheraufwand durchführbar.

Ein weiterer Algorithmus, *PPTC* (englisch: *probabilistic propagation in trees of clusters*), zum Schließen in beliebigen Bayesschen Netzen wurde auf Grundlage einer Idee

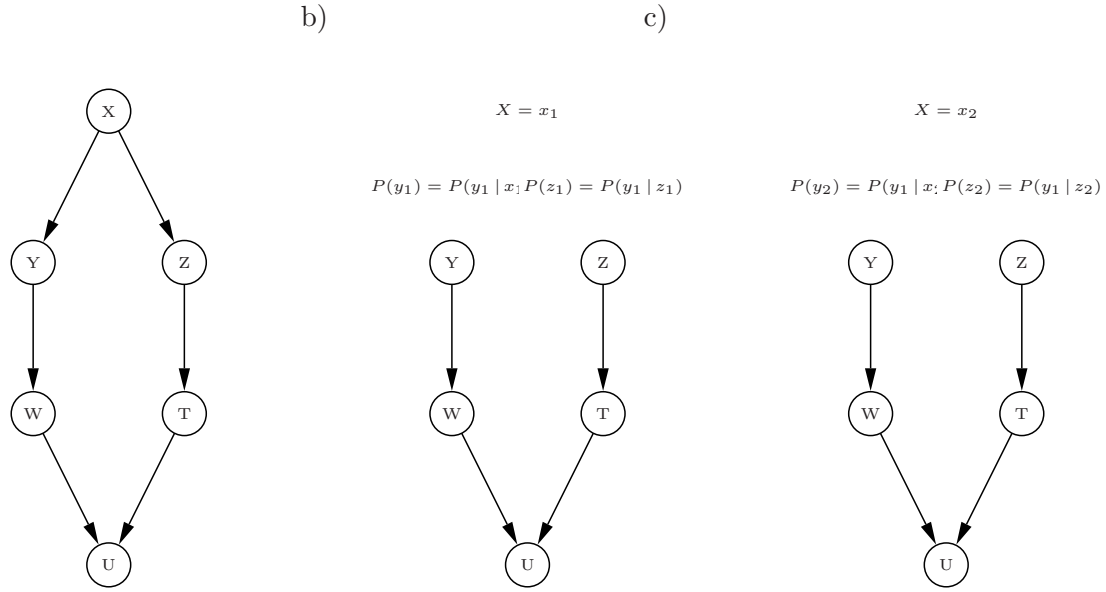


Abbildung 4.3: Beispiel für probabilistisches Schließen in mehrfach verbundenen Bayesschen Netzen. Durch Entfernen des Knoten X in a) entstehen beim *conditioning* zwei verschiedene Netze mit jeweils anderen a priori-Wahrscheinlichkeiten für Y und Z.

von Lauritzen und Spiegelhalter [Lau88] von Jensen et al. [Jen90] entwickelt. Dabei wird aus aus dem DAG des Bayesschen Netzes ein ungerichteter, triangulierter Graph erzeugt. Anschließend wird ein Baum abgeleitet, dessen Knoten für Cliquen des triangulierten Graphen stehen. Dieser Baum heißt *junction tree*. Anfragen in Form von bedingten Wahrscheinlichkeiten werden dann ebenfalls durch die Propagation bestimmter Wahrscheinlichkeiten zwischen den Knoten des Baumes berechnet. Der Algorithmus von Jensen et al. wurde unter Anderem in der Software für Bayessche Netze, HUGIN, implementiert. Einen pragmatischen Zugang zu ihrem Algorithmus bieten Huang et al. in [Hua96] an.

Alle bisher vorgestellten Schließalgorithmen nutzen die bedingten Unabhängigkeiten, die der Graph eines Bayesschen Netzes vorgibt, aus, um erfragte bedingte Wahrscheinlichkeiten zu berechnen. Darüber hinaus gibt es eine Gruppe von Algorithmen, die mit der faktorisierten Form der gemeinsamen Verteilung der Variablen arbeiten. Ein bekannter Vertreter dieser Algorithmen ist die *Variableneliminierung* (VE) von Zhang und Poole [Zha96]. Eine Anfrage $P(\boldsymbol{X} \mid \boldsymbol{Y})$ zerlegt die Variablenmenge $\boldsymbol{U}$ in drei Mengen: die beobachteten Variablen $Y \in \boldsymbol{Y}$, die befragten Variablen $X \in \boldsymbol{X}$ sowie jene, die für diese Anfrage irrelevant sind $(\boldsymbol{U} \setminus \boldsymbol{X} \cup \boldsymbol{Y})$. Durch Ausnutzung der faktorisierenden Zusammensetzung der gemeinsamen Verteilung über $\boldsymbol{U}$ gelingt es VE, die für die Berechnung der gewünschten Wahrscheinlichkeit $P(\boldsymbol{X} \mid \boldsymbol{Y})$ benötigte Marginalisierung $P(\boldsymbol{X}, \boldsymbol{Y})$ in den meisten Fällen wesentlich schneller zu bestimmen als über den in Glei-

chung 4.9 angedeuteten allgemeinen Weg. VE verfolgt damit eine ähnliche Strategie wie der SPI-Algorithmus (*symbolic probabilistic inference*) von D'Ambosio [Li94] und der Algorithmus *BucketTreeElimination* von Dechter et al. [Kas01, Dec99].

Approximierende Algorithmen. Obwohl die exakten Algorithmen PPTC und VE in den meisten Fällen schnell die gesuchte Wahrscheinlichkeit bestimmen können, ist sowohl ihre Laufzeit als auch ihr Speicherbedarf im ungünstigsten Fall exponentiell bezüglich der Größe des Bayesschen Netzes. Da für die meisten Anwendungsfälle für Bayessche Netze, z.B. dem Ableiten von Entscheidungen, die numerische Exaktheit des Ergebnisses keine hohe Priorität hat, wurden Algorithmen entwickelt, welche die gesuchten Wahrscheinlichkeiten in ausreichender Genauigkeit approximieren. Der Vorteil gegenüber den exakten Algorithmen ist ihre höhere Geschwindigkeit und ihr niedriger Speicherbedarf.

Verschiedene Verfahren setzen *Sampling*-Methoden ein, um bedingte Wahrscheinlichkeiten zu approximieren. Ein bekanntes Verfahren dieser Art ist das *Logic Sampling* von Henricon [Hen88]. Dabei werden die Variablen $\boldsymbol{U}$ des Netzes zunächst bezüglich der Nachfolgerrelation sortiert[1]. Für das Beispiel in Abbildung 4.2 wäre W, X, Y, U, Z, V eine gültige Sortierung. Um eine beliebige bedingte Wahrscheinlichkeit als Antwort auf eine Anfrage näherungsweise zu bestimmen, führt der Algorithmus Zufallsexperimente für bestimmte Zufallsvariablen des Bayesschen Netzes unter Verwendung der lokalen bedingten Verteilung durch. Um beispielsweise die Wahrscheinlichkeit $P(z_1 \mid w_1)$ zu berechnen ($W = w_1$ ist die Beobachtung), würde zunächst eine Stichprobe der Variablen X bezüglich der Verteilung $P(X \mid w_1)$ gezogen werden. Dies geschieht stellvertretend für die Berechnung der beiden Wahrscheinlichkeiten $P(x_1 \mid w_1)$ und $P(x_1 \mid w_1)$, die zur Berechnung von $P(z_1 \mid w_1)$ zur exakten Berechnung benötigt werden (vergleiche Gleichung 4.10). Anstelle korrekt berechneter bedingter Wahrscheinlichkeiten werden in der Folge aus den Stichproben-Statistiken abgeleitete Wahrscheinlichkeiten verwendet.

Ein weiteres *sampling*-basiertes Verfahren, dass erfolgreich zum approximierten Schließen eingesetzt wird, ist das *Gibbs-sampling*, dessen Grundprinzip im Abschnitt 3.2 über unüberwachte Lernverfahren für Sequenzmotive vorgestellt wurde. Eine bekannte Anwendung im Bereich Bayesscher Netze ist die Software *BUGS* (*Bayesian inference using Gibbs sampling*), die von Andrew Thomas et al. [Tho06] an der Universität in York entwickelt wurde.

4.3 Klassifikation mit Bayesschen Netzen

Im Rahmen dieser Dissertation werden Bayessche Netze vorwiegend zur Klassifikation von Mustern eingesetzt. Dazu werden speziell strukturierte Bayessche Netze eingesetzt,

[1] Nachfolgeordnung einer Knotenmenge eines DAG: Ist X ein Nachfolger von Y, so steht Y vor X in der Sortierung.

die in Unterabschnitt 4.3.2 eingeführt werden. Zuvor führt Unterabschnitt 4.3.1 in die Begrifflichkeiten der Muster und Merkmale ein.

4.3.1 Muster und Merkmale

In der Mustererkennung wird unter Klassifikation eine Zuordnungsvorschrift verstanden, die Elemente eines *Problemkreises* Ω einer von K möglichen Klassenbereichen Ω_1, Ω_2, ..., $\Omega_K \subseteq \Omega$ zuordnet. Ein Problemkreis könnte die Menge aller Pflanzenblüten sein, die Menge aller Menschen zum Zweck einer medizinischen Diagnose oder die Menge aller DNA-Sequenzen. Die Elemente $\boldsymbol{\omega} \in \Omega$ heißen *Muster* und bilden alle messbaren Eigenschaften des Teils der Welt ab, der für den betrachteten Problemkreis relevant sind. Für den Problemkreis der Pflanzenblüten könnten das unter anderem Fotografien, chemische Analysen oder Informationen über die Art der Fortpflanzung sein. Als solche sind Muster Vektoren aller dieser messbaren Größen [2].

In der Regel werden die Muster nicht direkt für die Klassenzuordnung verwendet, sondern es werden *Merkmale* generiert, die einen Klassenbereich des Problemkreises möglichst charakteristisch beschreiben. Die Generierung von Merkmalen kann durch Kombination von Musterkomponenten oder deren Transformation geschehen. So könnte bei der Klassifikation von Pflanzenblüten nicht eine Fotografie einer Blüte verwertbar sein, aber bestimmte Eigenschaften wie Anzahl der Blütenblätter oder deren Farbe, die sich anhand der Fotografie ermitteln lassen. Ein *Merkmal* X ist demnach eine Funktion, die Muster $\boldsymbol{\omega} \in \Omega$ in ein Element x des Wertebereiches D_X von X übersetzt [3]:

$$X : \left\{ \begin{array}{ccc} \Omega & \rightarrow & D_X \\ \boldsymbol{\omega} & \mapsto & x = X(\boldsymbol{\omega}) \end{array} \right. , \tag{4.19}$$

Zur Beschreibung eines Klassenbereiches werden meist verschiedene Merkmale berechnet. Ein *Klassifikator* δ operiert dann auf Vektoren der Merkmalsausprägungen $(x_1, x_2, \ldots, x_d)$ als Ersatz für die Muster:

$$\delta : D_{X_1} \times \ldots \times D_{X_d} \longrightarrow \{1, \ldots, K\} \tag{4.20}$$

Der Problemkreis Ω wird auch *Musterraum* genannt. Der Raum, der durch die Wertebereiche der verwendeten Merkmale aufgespannt wird, also das karthesische Produkt $\mathsf{D} = D_{X_1} \times \cdots \times D_{X_d}$, heißt *Merkmalsraum* der Klassifikationsaufgabe.

4.3.2 Bayessche Netz-Klassifikatoren

Bayessche Netze werden in dieser Arbeit als stochastische Modelle für die Musterklassifikation eingesetzt, wobei die Merkmale mit den diskreten Zufallsvariablen der Bayesschen

[2] Im eigentlichen Sinne sind Muster Vektoren von Funktionen, wobei jede dieser Funktion das Verhalten einer Meßaparatur, eines Sensors usw. beschreibt.

[3] Es wird für Merkmale dieselbe Notation verwendet wie für Zufallsvariablen eines Bayesschen Netzes, da diese in Kürze zusammengeführt werden.

Netze gleichgesetzt werden. Das Bayessche Netz definiert über die lokalen Parameter seiner Zufallsvariablen Verteilungen über den Wertebereichen der Merkmale. In günstigen Fällen belegen die Klassenbereiche des Musterraumes auch abgegrenzte Bereiche im Merkmalsraum D.

Bayessche Netze eignen sich in besonderer Weise für die Musterklassifikation. So können Bayessche Netze leicht Merkmale X mit verschiedenen Wertemengen verarbeiten. Ein zu klassifizierendes Muster wird als Beobachtung bzw. Evidenz dem Netz bekanntgegeben. Mittels der vorgestellten Algorithmen lassen sich anschließend Wahrscheinlichkeiten für die Belegung der unbeobachteten Variablen berechnen. Prinzipiell lässt sich jedes Bayessche Netz zur Klassifikation einsetzen. Unter einem Bayesschen Netz-Klassifikator wird in der Fachliteratur aber ein speziell strukturiertes Netz verstanden, dass dieser Aufgabe besonders entgegenkommt.

DEFINITION 4.2: *Sei* $\mathsf{D} = \{D_{X_1} \times \cdots \times D_{X_d}\}$ *ein Merkmalsraum. Ein* **Bayesscher Netz-Klassifikator** $\mathcal{C}$ *(BN-Klassifikator) für* K *Klassen über* D *ist ein Paar* $(\mathcal{B}, \delta)$ *mit folgenden Eigenschaften:*

1. $\mathcal{B}$ *ist ein Bayessches Netz mit der Variablenmenge* $X_1, \ldots, X_d, C$.
2. *Die Zufallsvariablen* X_i *heißen* **Merkmale** *und haben jeweils die Wertemenge* D_{X_i}.
3. *Die zusätzliche Variable* C *heißt* Klassenvariable. *Sie kann Werte* $\kappa \in \{1, \ldots, K\}$ *annehmen: den Klassenindizes.*
4. *Die Struktur von* $\mathcal{B}$ *ist der Art, dass jedes Merkmal* X_i *die Klassenvariable* C *als Elternknoten hat:* $C \in \mathbf{\Pi}_{X_i}$. *Die Klassenvariable* C *hat keine Elternknoten und ist damit bedingt unabhängig von den Merkmalsvariablen.*
5. *Die Klassifikationsvorschrift* δ *weißt jedem Muster* $\boldsymbol{x}$ *einen Klassenindex* κ *vermöge*

$$\delta(\boldsymbol{x}) = \operatorname*{argmax}_{\kappa'} P(C = \kappa' \,|\, \boldsymbol{x}). \tag{4.21}$$

zu.

Abbildung 4.4 zeigt einen typischen Bayesschen Netz-Klassifikator. Wie aus der Definition hervorgeht, erfolgt die Klassifikation eines Merkmalvektors $\boldsymbol{x} = (x_1, \ldots, x_d)$ nicht über die gemeinsame Verteilung $P_{\mathcal{B}}(\boldsymbol{x}, C = \kappa)$, sondern über Anfragen an die Klassenvariable C, wobei für die Merkmale $\boldsymbol{X}$ die Ausprägungen $\boldsymbol{x}$ als beobachtet vorliegen. Da die Klassenvariable keine Elternknoten besitzt, kann diese Anfrage in dem Fall, dass jedes Merkmal tatsächlich beobachtet wurde, also vollständiges Wissen vorliegt, auch leicht ohne aufwändige Schließalgorithmen gewonnen werden, indem über die Klassenvariable marginalisiert wird:

$$P(C = \kappa \,|\, \boldsymbol{x}) = \frac{P_{\mathcal{B}}(\boldsymbol{x}, C = \kappa)}{P(\boldsymbol{x})} \tag{4.22}$$

$$= \frac{P_{\mathcal{B}}(\boldsymbol{x}, C = \kappa)}{\sum_{\kappa'} P_{\mathcal{B}}(\boldsymbol{x}, C = \kappa')}. \tag{4.23}$$

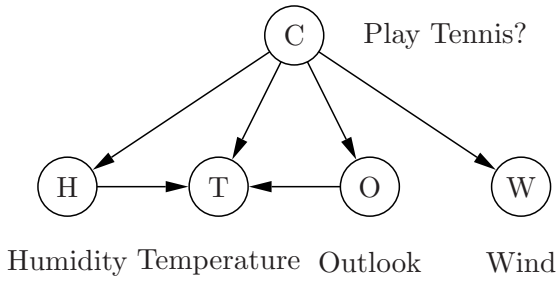


Abbildung 4.4: BN-Klassifikator zur Entscheidung, ob Tennis gespielt werden sollte? Die beobachteten Merkmale sind Luftfeuchtigkeit, Temperatur, Windverhältnisse und Aussicht. Hierbei handelt es sich um ein beliebtes Beispiel in der Literatur für Maschinelles Lernen.

Jedoch gewährleistet die besondere Struktur eines Klassifikatornetzes auch bei unvollständigem Wissen ein relativ kostengünstiges probabilistisches Schließen.

Die Struktur und die lokalen Wahrscheinlichkeitsverteilungen der Bayesschen Netz-Klassifikatoren werden in den meisten Fällen mittels statistischer Lernverfahren aus Beispieldaten des Merkmalsraumes gelernt (siehe dazu den folgenden Abschnitt 4.4). Für viele Mustererkennungsaufgaben liegen dafür nur Daten in begrenztem Umfang vor, so dass eine robuste Parameterschätzung beliebig strukturierter Netze, in denen eine Variable viele Elternvariablen und demzufolge eine enorme Anzahl von Parametern haben kann, nicht möglich ist. Aus diesem Grund wurden spezielle Formen Bayesscher Netz-Klassifikatoren entwickelt, für die jeweils Beschränkungen der Netzstruktur vereinbart sind. Ein günstiger Nebeneffekt ist dabei, dass für diese eingeschränkten Strukturen effiziente Lernalgorithmen existieren.

Das Ziel, die Anzahl der Parameter überschaubar zu halten, muss gegen das primäre Ziel, eine möglichst adäquate Beschreibung des Musterraumes für die Klassifikation zu erhalten, abgewogen werden. Komplexe Abhängigkeitsbeziehungen zwischen einer Gruppe von Merkmalen kann bei stark eingeschränkten Strukturen möglicherweise nicht mehr abgebildet werden. Im Folgenden sollen zwei Spezialisierungen des Bayesschen Netz-Klassifikators vorgestellt werden.

Naiver Bayes-Klassifikator. Den größten strukturellen Einschränkungen unterliegt der *Naïve Bayes-Klassifikator* (NB). Außer den obligatorischen Kanten von der Klassenvariable zu den Merkmalsvariablen gibt es gar keine Struktur. Jedes Merkmal X_i wird als

bedingt unabhängig von jedem anderen Merkmal X_j bei gegebener Klasse κ angenommen:

$$P(X_i \mid X_j, C = \kappa) = P(X_i \mid C = \kappa). \tag{4.24}$$

Da keine Abhängigkeiten unter den Merkmalen beachtet werden, kommt der NB unter allen Bayesschen Netz-Klassifikatoren mit der kleinsten Anzahl an Wahrscheinlichkeitsparametern aus. Bei einem NB mit d Merkmalen, von denen jedes v verschiedene Werte annehmen kann, beträgt die Anzahl freier Parameter $Kd(v-1)$.

Trotz dieser Unabhängigkeitsannahme zeigt der NB häufig erstaunlich gute Erkennungsleistungen. Seinen Erfolg, gemessen in Häufigkeit seiner Anwendung, verdankt der NB jedoch seinem konkurrenzlos niedrigen Lernaufwand, da sich das Lernen dank fest vorgegebener Netzstruktur auf die Maximum-Likeli-Schätzung der lokalen Wahrscheinlichkeitsparameter $\theta_{x_i|\kappa}$ beschränkt.

Baumartiger Bayes-Klassifikator. Der NB-Ansatz stößt an seine Grenzen, wenn zwischen den Merkmalen statistische Abhängigkeiten bestehen, die er nicht darstellen kann. Merkmale könnten, als unabhängig betrachtet, schlechte Prädiktoren der verschiedenen Klassen sein, jedoch konditioniert auf ein mit ihm korrelierenden Merkmal klare Unterschiede zwischen den Klassen zeigen.

Ein häufig verwendeter Kompromiss zwischen dem Zulassen beliebiger Abhängigkeiten und dem Ignorieren sämtlicher Abhängkeiten stellt der *baumartige* Bayes-Klassifikator dar. Die geläufige Bezeichnung, *TAN* (englisch: *Tree-augmented network*), soll auch in dieser Arbeit verwendet werden.

DEFINITION 4.3: *Ein Bayesscher Netz-Klassifikator $\mathcal{C} = (\mathcal{B}, \delta)$ ist ein* TAN, *wenn jedes Merkmal X_i bedingt abhängig von höchstens einem weiteren Merkmal gegeben C ist:*

$$\forall_{X_i} |\mathbf{\Pi}_{X_i} \setminus \{C\}| \leq 1. \tag{4.25}$$

Der TAN-Klassifikator vereinigt eine Reihe günstiger Eigenschaften, die ihn zu einem vielverwendeten Ansatz machen. Im Gegensatz zum NB kann er Abhängigkeiten zwischen Merkmalen modellieren und damit die gemeinsame Verteilung des Musterraumes adäquater abbilden. Die Einschränkungen hinsichtlich erlaubter Abhängigkeiten wirken sich in vielen Anwendungsbeispielen nicht negativ aus. Im Vergleich zu beliebig strukturierten Bayesschen Netzen bleibt die Anzahl der Parameter überschaubar und nach oben begrenzt. Ein TAN für K Klassen und d Merkmale, die jeweils v verschiedene Werte annehmen können, besitzt maximal Kd^2 Wahrscheinlichkeitsparameter. Ein weiterer Vorteil gegenüber beliebigen Bayesschen Netzen ist, dass es für TAN-Klassifikatoren einen effizienten Algorithmus gibt, die optimale Struktur in Bezug auf eine Lernstichprobe zu bestimmen (siehe Unterabschnitt 4.4.2).

4.4 Lernen Bayesscher Netze

In einigen Fällen ist es möglich, Bayessches Netze anhand kausaler Zusammenhänge des Anwendungsbereiches und einer bewussten Abwägung möglicher Ereignisse von Hand zu konstruieren. Das gilt insbesondere für Diagnosesysteme wie in dem Beispiel in Abbildung 4.1.

In den meisten Fällen sollen Bayessche Netze jedoch automatisch aus vorhandenen Daten konstruiert werden. Unter *Daten* sei im Folgenden eine Stichprobe gemeint, die eine bestimmte Anzahl von Variablenkonfigurationen für eine gegebene Menge von Variablen enthält.

DEFINITION 4.4: *Sei $\boldsymbol{X} = (X_1, \ldots, X_d)$ ein Vektor von d Zufallsvariablen bzw. Merkmalen eines Merkmalsraums. Eine* Stichprobe d *von $\boldsymbol{X}$ der Größe N ist eine Menge*

$$\mathsf{d} = \left\{ \begin{array}{llll} (x_1^{(1)}, & x_2^{(1)}, & \ldots & x_d^{(1)}), \\ (x_1^{(2)}, & x_2^{(2)}, & \ldots & x_d^{(2)}), \\ \ldots & \ldots & \ldots & \ldots \\ (x_1^{(N)}, & x_2^{(N)}, & \ldots & x_d^{(N)}) \end{array} \right\}, x_i^{(n)} \in D_{X_i} \tag{4.26}$$

von N Vektoren, die Realisierungen der Zufallsvariablen (bzw. beobachtete Ausprägungen der Merkmale) in $\boldsymbol{X}$ enthalten. Diese Vektoren $\boldsymbol{x}^{(n)}$ heißen Stichprobenelemente.

Wird eine Stichprobe verwendet, um einen Bayesschen Netz-Klassifikator zu lernen, so werden *etikettierte* Stichproben und *unetikettierte* unterschieden. Bei etikettierten Stichproben besitzt der Variablenvektor $\boldsymbol{X}$ eine zusätzliche Komponente, die Klassenvariable C. Für jedes Stichprobenelement ist demnach die Klassenzugehörigkeit bekannt.

Eine Stichprobe d ist eine endliche Teilmenge aller Merkmalsvektoren, die aus Mustern des Musterraumes Ω gewonnen wurden, also eine endliche Teilmenge des Merkmalsraums. Bayessche Netze werden anhand einer solchen Stichprobe trainiert, da angenommen werden muss, dass sich die gemeinsame Verteilung der Merkmale im Merkmalsraum D und die empirische Verteilung der Merkmalsvektoren in der Stichprobe d nur geringfügig unterscheiden.

Ziel ist es, ein Bayessches Netz zu finden, dass möglichst gut die gemeinsame Verteilung des Merkmalsraumes beschreibt. Dies ist nicht identisch mit dem Ziel, ein Bayessches Netz zu finden, dass möglichst gut die empirische Verteilung der Stichprobe beschreibt, denn das wäre trivialerweise mit einem vollverbundenen Bayesschen Netz möglich. Viele der dabei gemessenen Abhängigkeiten sind jedoch nur Folge einer unzureichenden Abdeckung des Merkmalsraums durch die Stichprobe, bzw. gibt es für die meisten Kombinationen gar keine Stichprobenelemente. Diese Problematik wird auch als *Fluch der Dimension* bezeichnet. Die Folge wäre eine Überanpassung des Bayesschen Netzes an die Lernstichprobe.

Die Lernaufgabe bei Bayesschen Netzen vereint also zwei Ziele: 1.) Das resultierende Netz soll die gemeinsame Verteilung des Merkmalsraumes repräsentieren und 2.) Das Netz soll dafür möglichst wenige freie Parameter benötigen. Es ist üblich, beide Ziele in einem gemeinsamen Qualitätsmaß zu vereinen und dieses durch einen Lernprozess zu optimieren. Ein bekanntes Qualitätsmaß ist der sogenannte *MDL* (englisch: *minimum description length*, siehe Seite 84).

Die Aufgabe, ein Bayessches Netz zu bestimmen, dass hinsichtlich eines Qualitätsmaßes für eine Stichprobe optimal ist, ist *NP-vollständig* [Chi96], kann jedoch in zwei Teilaufgaben zerlegt werden: 1.) *Strukturlernen:* Finde einen gerichteten, azyklischen Graphen, der optimal hinsichtlich eines Qualitätsmaßes ist. 2.) *Parameterlernen:* Für eine gegebene Struktur des Netzes führe eine Parameterschätzung für alle Wahrscheinlichkeitsparameter des Netzes durch. In diesem Abschnitt wird im Unterabschnitt 4.4.1 zunächst das Problem der Parameterschätzung behandelt. Anschließend werden in Unterabschnitt 4.4.2 Verfahren zur Bestimmung der optimalen Struktur vorgestellt. Zwei weitere Unterabschnitte widmen sich weiteren Aspekten des Lernens von Klassifikatoren, die nur indirekt Bayessche Netze betreffen aber von Relevanz für das im folgenden Kapitel vorgestellte TFBS-Erkennungssystem sind: Unterabschnitt 4.4.3 beschäftigt sich mit der Auswahl geeigneter Merkmale zur Konstruktion eines Klassifikators, Unterabschnitt 4.4.4 mit der Diskretisierung ursprünglich kontinuierlicher Merkmale, um sie in diskreten Bayesschen Netzen einsetzen zu können.

4.4.1 Lernen der Parameter

Dieser Unterabschnitt erläutert, wie die Wahrscheinlichkeitsparameter eines Bayesschen Netzes mit festgelegter Struktur mit Hilfe einer Lernstichprobe d geschätzt werden. Zunächst beschränken sich die Darstellungen losgelöst von Bayesschen Netzen auf eine einzelne Zufallsvariable. Anschließend wird argumentiert, dass die Parameter der Zufallsvariablen eines Bayesschen Netzes unabhängig voneinander geschätzt werden können, um die globale Zielfunktion für die Daten d zu maximieren.

Sei also zunächst eine diskrete Zufallsvariable X mit Wertebereich $D_X = \{1, \ldots, \nu\}$ gegeben. Für diese liegt eine Stichprobe

$$\mathsf{d} = \{x^{(1)}, \ldots, x^{(N)}\} \tag{4.27}$$

der Größe N vor.

Ein bekannter Schätzer dieser Wahrscheinlichkeiten ist der *Maximum-Likelihood-Schätzer* (ML-Schätzer). Für eine Stichprobe d versucht dieser, jene Wahrscheinlichkeitsparameter $\boldsymbol{\theta}^{(ML)}$ zu bestimmen, welche die logarithmierte *Stichprobenwahrscheinlichkeit*, auch ML-Zielfunktion genannt, maximiert:

$$\boldsymbol{\theta}^{(ML)} \quad =_{\text{def}} \quad \operatorname*{argmax}_{\boldsymbol{\theta}} \ell_{\boldsymbol{\theta}}(\mathsf{d}) \tag{4.28}$$

$$= \quad \operatorname*{argmax}_{\boldsymbol{\theta}} \sum_{n=1}^{N} \log P(x^{(n)} \,|\, \boldsymbol{\theta}) \tag{4.29}$$

$$= \quad \operatorname*{argmax}_{\boldsymbol{\theta}} \sum_{k=1}^{\nu} \underbrace{\#k}_{|\{(x)\in \mathsf{d}: x=k\}|} \log P(k \,|\, \theta) \tag{4.30}$$

$$= \quad \operatorname*{argmax}_{\boldsymbol{\theta}} \sum_{k=1}^{\nu} \#k \log \theta_k. \tag{4.31}$$

Dabei ist θ_k die Wahrscheinlichkeit $P(X = k)$ und $\#k$ die absolute Häufigkeit des Wertes k in d für $k \in \{1, \ldots, \nu\}$[4]. Es leuchtet ein, dass zur Maximierung der Summe in Ausdruck 4.31 die voneinander unabhängigen Summanden $\#k \log \theta_k$ einzeln maximiert werden müssen. Die ML-Schätzung für θ_k ergibt sich durch die relativen Häufigkeiten $P_{\mathsf{d}}(k) = \frac{\#k}{N}$:

$$\theta_k^{(ML)} = P_{\mathsf{d}}(k). \tag{4.32}$$

Wäre X Teile eines Bayesschen Netzes und hätte X die Menge Elternvariablen $\boldsymbol{\Pi}_X$, dann müsste diese Schätzung für jede Konfiguration $\boldsymbol{\pi}_X$ der Elternvariablen durchgeführt werden, um die jeweiligen Parameter $\boldsymbol{\theta}_{X \,|\, \pi_X}$ zu erhalten. Insbesondere dann, wenn die gegebene Struktur eines Bayesschen Netzes einen oder mehrere Elternvariablen für X vorsieht, wird es häufig vorkommen, dass für eine konkrete Elternkonfiguration nicht jeder Wert $x \in D_X$ in der Stichprobe vorkommt. Die Folge wären Nullwahrscheinlichkeiten, die jedoch in Bayesschen Netzen nicht auftreten dürfen. Ein Grund hierfür ist die Funktionsweise der Schließalgorithmen. Ein anderer Grund ist die ungewollte Überanpassung des Modells an die Stichprobe, denn eine lokale Nullwahrscheinlichkeit würde die gemeinsame Wahrscheinlichkeit bisher ungesehener Merkmalsvektoren auslöschen.

Aus diesen Gründen wird für die Schätzung der lokalen Parameter einer Variablen X ein anderes Verfahren angewendet, dass im Folgenden beschrieben wird. Zunächst wird der Fall betrachtet, dass keine Stichprobe für X vorliegt, d.h., dass bisher keine Information über die Verteilung von X vorliegt. In diesem Fall wird davon ausgegangen, dass die Parameter $\theta_1, \ldots, \theta_\nu$ dieser Verteilung gemäß einer *Dirichlet*-Verteilung verteilt sind.

DEFINITION 4.5: *Die* **Dirichlet-Dichte-Funktion** *der Ordnung ν mit Parametern $\alpha_1, \alpha_2, \ldots, \alpha_\nu \in \mathbb{R}$ mit $M = \sum_{k=1}^{\nu} \alpha_k$ ist für einen $\nu - 1$-dimensionalen euklidischen Raum definiert durch:*

$$f(\theta_1, \theta_2, \ldots, \theta_{\nu-1}) = \frac{\Gamma(M)}{\prod_{k=1}^{\nu} \Gamma(\alpha_k)} \prod_{k=1}^{\nu} \theta_k^{\alpha_k - 1} \tag{4.33}$$

[4]Mit $\ell_\cdot(\,\cdot\,)$ werden in dieser Arbeit stets ML-Zielfunktionen bezeichnet. Der Index variiert mit der Art des Modells, dass mit der jeweiligen ML-Schätzung optimiert werden soll. Teilweise werden als Index Bezeichner für Modelle verwendet, teilweise auch Parametersätze eines Modells.

wobei θ_ν eine Abkürzung für $1 - \sum_{k=1}^{\nu-1}$ ist und gefordert wird, dass $\sum_{k=1}^{\nu} p_k = 1$ und $0 \leq p_k \leq 1$ für alle k gilt. Zufallsvariablen $\Theta_1, \ldots, \Theta_\nu$, die diese Dichtefunktion besitzen, sind gemäß einer **Dirichlet-Verteilung** *$Dir(\theta_1, \theta_2, \ldots, \theta_{\nu-1}\,;\, \alpha_1, \alpha_2, \ldots, \alpha_\nu)$ verteilt[5].*

Mit der Dirichlet-Verteilung über Zufallsvariablen Θ_k wird also das *a priori*-Wissen über die Wahrscheinlichkeitsparameter θ_k der Variablen X ausgedrückt, indem festgelegt wird:

$$P(X = k \,|\, \theta_k) = \theta_k. \tag{4.34}$$

Es gilt weiterhin $P(X = k) = \mathcal{E}[\Theta_k]$, denn

$$\begin{aligned} P(X = k) &= \int_0^1 P(X = k \,|\, \theta_k) P(\theta_k) d\theta_k & (4.35)\\ &= \int_0^1 \theta_k P(\theta_k) d\theta_k = \mathcal{E}[\Theta_k]. & (4.36) \end{aligned}$$

Für den Erwartungswert $\mathcal{E}[\Theta_k]$ einer Komponente von Dirichlet-verteilten Zufallsvariablen $\Theta_1, \ldots, \Theta_\nu$ gilt aber

$$\mathcal{E}[\Theta_k] = \frac{\alpha_k}{M}, \tag{4.37}$$

woraus unmittelbar folgt:

$$\Longrightarrow P(X = k) = \frac{\alpha_k}{M}. \tag{4.38}$$

Als Schätzwert des Parameters θ_k wird der Erwartungswert der Zufallsvariablen Θ_k, also $\frac{\alpha_k}{M}$ verwendet.

Beispiel: *Das einmalige Würfeln mit einem sechseitigen Würfel sei durch die Zufallsvariable W mit der unbekannten Verteilung $p_1, \ldots, p_6$ beschrieben. Soll davon ausgegangen werden, dass der Würfel fair ist, dann ist der Würfel beispielsweise gemäß $Dir(p_1, \ldots, p_5\,;\, 2, 2, 2, 2, 2, 2)$ verteilt. Dann gilt:*

$$\begin{aligned} Dir(p_1, \ldots, p_5\,;\, 2, 2, 2, 2, 2, 2) &= \frac{\Gamma(12)}{\prod_{k=1}^{6} \Gamma(2)} p_1 \cdot p_2 \cdot p_3 \cdot p_4 \cdot p_5 \cdot p_6 & (4.39)\\ &= 11.25\,(p_1 \cdot p_2 \cdot p_3 \cdot p_4 \cdot p_5 \cdot p_6). & (4.40) \end{aligned}$$

Die Annahme über die Wahrscheinlichkeit, eine 3 zu würfeln ist gemäß Gleichung 4.38 wie für einen fairen Würfel erwartet

$$P(X = 3) = \frac{\alpha_3}{M} = \frac{1}{6}. \tag{4.41}$$

Sollte der Würfel jedoch äußerlich den Verdacht erwecken (z.B. durch ungleich schwere Materialen auf den Flächen), dass eine 1 bei ihm besonders wahrscheinlich ist, dann wäre eine Dirichlet-Verteilung angeraten, bei dem der Parameter α_1 größer ist, als die

[5] $\Gamma(\cdot)$ heißt Gammefunktion und entspricht der Fakultätsfunktion für reelle Zahlen.

anderen, etwa $Dir(p_1, \ldots, p_5\,;\, 5, 1, 1, 1, 1, 1)$. *Die Wahrscheinlichkeit für dieses Ereignis wäre dann:*

$$P(X = 1) = \frac{\alpha_1}{M} = \frac{5}{10} = \frac{1}{2}. \tag{4.42}$$

Mit Hilfe einer geeigneten Dirichlet-Verteilung kann eine a priori Verteilung einer Variablen X des Netzes für jede Konfiguration der Elternknoten festgelegt werden. Eine Stichprobe d für X, also die mehrfache Realisierung des durch X beschriebenen Zufallsexperiments erweitert das Wissen darüber, wie groß die Wahrscheinlichkeiten $\theta_1, \ldots, \theta_\nu$ tatsächlich sind. Als Nächstes wird erläutert, wie dieses zusätzliche Wissen in den eben eingeführten Ansatz einzubringen ist.

Dazu seien, wie schon zuvor, Θ_k für $1 \leq k \leq \nu$ Dirichlet-verteilte Zufallsvariablen mit Parametern α_k. Dann gilt:

$$f(\theta_1, \ldots, \theta_{\nu-1} \,|\, \mathsf{d}) = \frac{\left(\prod_{k=1}^{\nu} \theta_k^{\#k}\right) f(\theta_1, \ldots, \theta_{\nu-1})}{\mathcal{E}\left[\prod_{k=1}^{\nu} F_k^{\#k}\right]} \tag{4.43}$$

$$= Dir(\theta_1, \ldots, \theta_{\nu-1}\,;\, \alpha_1 + \#1, \ldots, \alpha_\nu + \#\nu). \tag{4.44}$$

Dabei bezeichnet $f(\theta_1, \ldots, \theta_{\nu-1} \,|\, \mathsf{d})$ die bedingte Dirichletdichte bei gegebener Stichprobe d. Die Stichprobenstatistiken $\#k$ werden also zu den ursprünglichen Dirichletparametern α_k addiert, um eine neue Dirichletverteilung mit Parametern $\alpha'_k = \alpha_k + \#k$ der Zufallsvariablen Θ_k zu erhalten. Um den Einfluss der Stichprobe zu notieren, erscheint d als Bedingungsteil in der Dirichletdichte. Nach dieser Anpassung gilt für einen neu beobachteten Wert $x^{(N+1)}$ von X:

$$P(x^{(N+1)} = k) = \mathcal{E}[\Theta_k \,|\, \mathsf{d}] = \frac{\alpha_k + \#k}{M + N}, \tag{4.45}$$

wobei N die Größe der Stichprobe ist.

Beispiel: *Sei erneut ein Würfel gegeben, der zunächst als fair angenommen wird, d.h., seine Wahrscheinlichkeiten* $p_1, \ldots, p_5$ *sind z.B. gemäß* $Dir(p_1, \ldots, p_5\,;\, 1, 1, 1, 1, 1, 1)$ *verteilt. Die Wahrscheinlichkeit für jedes Würfelergebnis liegt also bei* $\frac{1}{6}$. *Nach* 10*-maligem Würfeln ergibt sich die Stichprobe* $\{1, 2, 4, 1, 1, 3, 1, 5, 1, 5\}$. *Unter Berücksichtigung dieser Daten wird nun angenommen, dass die Würfelverteilung gemäß* $Dir(p_1, \ldots, p_5\,;\, 1 + 5, 1 + 1, 1 + 1, 1 + 1, 1 + 5, 1 + 0)$ *verteilt ist. Die Wahrscheinlichkeit für eine* 1 *wird entsprechend größer.*

Das Beispiel deutet auch an, wie die Wahl der Dirichlet-Parameter α_k die Schätzwerte für die Parameter θ_k der Zufallsvariablen beeinflusst:

- $\alpha_1 = \alpha_2 = \cdots = \alpha_\nu = 1$: Hier wird als Vorwissen angenommen, dass jede Kombination von Parametern θ_k für $1 \leq k \leq \nu$ gleichwahrscheinlich sind. Solch eine Wahl würde getroffen werden, wenn kein besonderes Wissen über die wahre Verteilung vorliegt.

- $\alpha_1 = \alpha_2 = \cdots = \alpha_\nu > 1$: Diese Werte drücken aus, dass es als wahrscheinlicher angesehen wird, dass die Wahrscheinlichkeit für das Ereignis $X = k$ bei ungefähr $\frac{\alpha_k}{M}$ liegt und der Stichprobe nicht so viel Glauben geschenkt wird.
- $\alpha_1 = \alpha_2 = \cdots = \alpha_\nu < 1$: In diesem Fall wird sich eher auf die aus der Stichprobe berechneten empirischen Wahrscheinlichkeiten verlassen. Die a priori-Annahmen gehen in geringerem Maße in die Schätzung ein.
- $\alpha_1 \neq \alpha_2 \neq \cdots \neq \alpha_\nu > 1$: wenn detailiertes Vorwissen vorhanden ist, dass einige der Ereignisse weniger oder mehr wahrscheinlich sind.

Nachdem die Schätzung der Wahrscheinlichkeitsparameter θ_k einer Variablen X geklärt ist, werden die gemachten Beobachtungen auf ein Bayessches Netz $\mathcal{B}$ mit d Zufallsvariablen angewendet. Dafür ist etwas Notation nötig. Jede der Variablen X_i hat im Weiteren einen Wertebereich $D_{X_i} = \{1, \ldots, \nu_i\}$. Die Menge der Elternvariablen von X_i wird abkürzend mit $\mathbf{\Pi}_i$ bezeichnet, die Menge aller möglichen Konfigurationen von $\mathbf{\Pi}_i$ wird durchnumeriert: $\{\boldsymbol{\pi}_{i1}, \ldots, \boldsymbol{\pi}_{iq_i}\}$, wobei q_i die Anzahl dieser Konfigurationen ist. Die Zufallsvariablen der Parameter $\boldsymbol{\theta}$ gliedern sich wie folgt. Für jede Variable X_i gibt es einen Satz Zufallsvariablen $\mathbf{\Theta}_i$ für den Parametersatz $\boldsymbol{\theta}_i$, so dass

$$\mathbf{\Theta} = \mathbf{\Theta}_1 \cup \ldots \mathbf{\Theta}_d \tag{4.46}$$

gilt. Jeder Variablensatz $\mathbf{\Theta}_i$ besteht wiederum aus Vektoren

$$\mathbf{\Theta}_{ij} = (\Theta_{ij1}, \ldots, \Theta_{ij\nu_i}) \tag{4.47}$$

für jede Elternkonfiguration ($1 \leq j \leq q_i$), deren Komponenten Θ_{ijk} schließlich Zufallsvariablen der einzelnen Parameter sind, welche die Wahrscheinlichkeit dafür, dass $X_i = k$ ist, unter der gegebenen Elternkonfiguration $\boldsymbol{\pi}_j$ repräsentieren. Die Vektoren $\mathbf{\Theta}_{ij}$ sind Dirichlet-verteilt gemäß

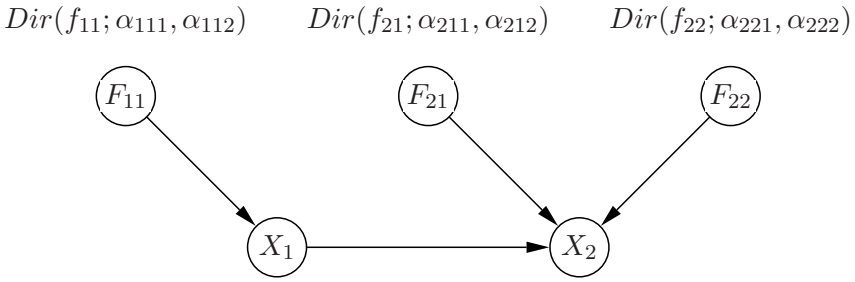
$$Dir(\theta_{ij1}, \ldots, \theta_{ij\nu_{i-1}}; \alpha_{ij1}, \ldots, \alpha_{ij\nu_i}). \tag{4.48}$$

Hilfsweise kann die gesamte vereinbarte Notation in einem sogenannten *erweiterten* Bayesschen Netz ($\mathcal{B}$, $\mathbf{\Theta}$, $\boldsymbol{f}$) dargestellt werden[6], wobei $\boldsymbol{f}$ die Gesamtheit aller Dirichlet-Dichten für die Parameter-Zufallsvariablen $\mathbf{\Theta}$ ist. Dieses erweiterte Netz ist so aufgebaut, dass es zunächst die Variablen X_i aus $\mathcal{B}$ in der durch $\mathcal{B}$ definierten Struktur enthält, und zusätzlich für jedes $i \in \{1, \ldots, d\}$ Zufallsvariablen Θ_{ijk} mit $1 \leq j \leq q_i$ und $1 \leq k \leq \nu_i$, die ausgehende Kanten nach X_i haben. Zusätzlich wird gefordert, dass jeweils

$$P(X_i = k \,|\, \boldsymbol{\pi}_{ij}, \boldsymbol{\theta}_{i1}, \ldots, \boldsymbol{\theta}_{iq_i}) = \theta_{ijk} \tag{4.49}$$

ist. Abbildung 4.5 zeigt ein einfaches erweitertes Bayessches Netz. Da alle zusätzlichen Variablen Θ_{ijk} Wurzeln im erweiterten Netz sind, sind sie bedingt unabhängig. Unter

[6] Dieses Bayessche Netz wird auch kontinuierliche Variablen enthalten.



$$P(X_1 = 1) = \frac{\alpha_{111}}{\alpha_{111}+\alpha_{112}}$$

$$P(X_1 = 2) = \frac{\alpha_{112}}{\alpha_{111}+\alpha_{112}}$$

Abbildung 4.5: Erweitertes Bayessches Netz für zwei "*echte* " Variablen mit jeweils zwei Werten.

Verwendung etwas informeller Dichtefunktionswerte heißt das:

$$P(\boldsymbol{\theta}) = \prod_{i=1}^{d} P(\boldsymbol{\theta}_i) \tag{4.50}$$

$$= \prod_{i=1}^{d} \prod_{j=1}^{q_i} f(\boldsymbol{\theta}_{ij};\, \boldsymbol{\alpha}_{ij}). \tag{4.51}$$

Diese Parameterunabhängigkeit sichert zu, dass analog zu der Betrachtung einer einzelnen Variable gilt:

$$P(X_i = k|\boldsymbol{\pi}_{ij}) = \frac{\alpha_{ijk}}{M_{ij}}, \tag{4.52}$$

mit $M_{ij} = \sum_{k=1}^{\nu_i} \alpha_{ijk}$.

Nun sei erneut eine Stichprobe d der Größe N gegeben. Auch hier ist etwas zusätzliche Notation nötig. Mit N_{ij} wird die Anzahl der Stichprobenelemente bezeichnet, in denen für X_i die $j$te Elternkonfiguration vorliegt, und mit $\#k_{ij}$ der Anteil dieser Fälle, in denen $X_i = k$ ist.

Aufgrund der Parameterunabhängigkeit kann die Stichprobe unabhängig auf jede Dirichlet-Verteilung angewendet werden und es gilt für $1 \leq i \leq d$ und $1 \leq j \leq q_i$:

$$P(\theta_{ij1}, \ldots, \theta_{ij\nu_i} \,|\, \mathsf{d}) \sim Dir(\theta_{ij1}, \ldots, \theta_{ij\nu_i-1};\, \alpha_{ij1} + \#1_{ij}, \ldots, \alpha_{ijk} + \#k_{ij}). \tag{4.53}$$

Wie zuvor gilt deshalb für einen neu beobachteten Merkmalsvektor $\boldsymbol{x}^{(M+1)}$

$$P(x_i = k \,|\, \boldsymbol{\pi}_{ij}) = \frac{\alpha_{ijk} + \#k_{ij}}{M_{ij} + N_{ij}}. \tag{4.54}$$

Als Ergebnis der Betrachtung von a priori Dirichletverteilungen bei der Parameterschätzung von Bayesschen Netzen ergibt sich demnach, dass die Hyperparameter α_{ijk} als Glättungskoeffizienten (im englischen häufig: *pseudo counts*) verwendet werden, um eine Überanpassung der Netzparameter an die Lernstichprobe zu vermeiden.

4.4.2 Strukturen lernen

Im vorhergehenden Abschnitt wurde angenommen, dass die Struktur des zu trainierenden Bayesschen Netzes bekannt ist und sich auf die Schätzung der Parameter Θ beschränkt. Dieser Unterschnitt wird sich mit Verfahren beschäftigen, die mit Hilfe einer Stichprobe d jene Struktur des Bayesschen Netzes ermitteln, die adäquat und sparsam die statistischen Abhängigkeiten zwischen den Merkmalen des Merkmalsraums abbildet.

Bei der Formalisierung der Aufgabe des Strukturlernens wird sich häufig der Bayesschen Statistik bedient. Dabei gibt es Hypothesen G^h über die Struktur eines Bayesschen Netzes und Wahrscheinlichkeiten $P(G^h)$ für das Zutreffen dieser Hypothesen. Die Lernaufgabe besteht darin, jene Hypothese mit maximaler a posteriori Wahrscheinlichkeit $P(G^h \mid \mathsf{d})$ zu finden. Die a posteriori Wahrscheinlichkeiten werden unter Verwendung des Bayes-Theorems berechnet:

$$P(G^h \mid \mathsf{d}) = \frac{P(\mathsf{d} \mid G^h)P(G^h)}{P(\mathsf{d})}. \tag{4.55}$$

Wollte man sich nicht für eine Struktur entscheiden, so könnte die Wahrscheinlichkeit eines bisher unbeobachteten Stichprobenelements $\boldsymbol{x}^{(N+1)}$ theoretisch über die Summation über alle Strukturhypothesen und über alle dazu passenden Modellparametersätze $\boldsymbol{\theta}_G$ berechnet werden:

$$P(\boldsymbol{x}^{(N+1)} \mid \mathsf{d}) = \sum_{G^h} P(G^h \mid \mathsf{d}) \int P(\boldsymbol{x}^{(N+1)} \mid G^h, \boldsymbol{\theta}_G) P(\boldsymbol{\theta}_G \mid \mathsf{d}, G^h) d\boldsymbol{\theta}_G). \tag{4.56}$$

Diese Summe ist aufgrund der großen Anzahl verschiedener Hypothesen G^h nur aufwendig zu berechnen, jedoch approximierbar. Beispielsweise wird unter der Annahme, dass es eine einzige (sehr) wahrscheinliche Hypothese gibt, obige Gleichung auf eine Hypothese G^h reduziert. Das Integral wird ebenfalls durch die optimalen Parameter zu dieser Hypothese ersetzt, die gemäß dem in Abschnitt 4.4.1 beschriebenen Verfahren geschätzt werden können.

Es können zwei Hauptstrategien zur Modellauswahl unterschieden werden. Die eine Strategie besteht aus einem Suchverfahren im Raum aller Strukturen und einer Bewertungsfunktion. Die andere Strategie versucht, Abhängigkeiten (d.h. Kanten) durch lokale Abhängigkeitsanalysen, etwa über einen χ^2-Test oder die Transinformation, zu identifizieren, und die Struktur möglichst in Einklang mit diesen Abhängigkeiten aufzubauen. Daneben gibt es Methoden, mehrere wahrscheinliche Strukturhypothesen gleichzeitig zu

zulassen, und über diese zu mitteln. Besonders gut funktioniert das, wenn möglichst verschiedene Strukturen ausgewählt werden, um das ganze Spektrum des Strukturraumes abzudecken.

Modellsuche. Verschiedene Qualitätskriterien werden für die Modellsuche eingesetzt. Ein naheliegendes Kriterium ist die logarithmische a posteriori Wahrscheinlichkeit einer Strukturhypothese G^h

$$\log P(G^h \,|\, \mathsf{d}) = \log P(G^h) + \log P(\mathsf{d} \,|\, G^h). \tag{4.57}$$

Dieses Maß hat zwei Komponenten: die a priori Wahrscheinlichkeit der Hypothese und die Stichprobenwahrscheinlichkeit.

Einige Qualitätsmaße zur Bestimmung einer optimalen Struktur berücksichtigen den Interessenkonflikt, einerseits die gemeinsame Verteilung des Merkmalsraumes detailgetreu darstellen zu können, und andererseits die Anzahl der Parameter des Bayesschen Netzes gering zu halten. Zwei der bekanntesten Qualitätsmaße sind 1.) der MDL (für englisch *minimum description length*) [Fri97] und 2.) das *Bayessche Bewertungs Kriterium* (BSC für englisch *Bayesian scoring criterion*) [Nea03].

Der schon eingangs dieses Abschnittes erwähnte MDL berücksichtigt diese beiden Ziele. Modelle werden dabei anhand ihrer Fähigkeit beurteilt, Daten zu komprimieren. Belohnt werden soll das Modell, dass die Verteilung der Stichprobe mit möglichst wenigen Parametern abzubilden vermag. Das Prinzip des MDL basiert auf der Erstellung eines optimalen Codes. Die häufigsten Stichprobenelemente sollen dabei möglichst kurze Wörter des Codes belegen. Eine natürliche Wahl, dies zu beschreiben, ist die negative logarithmische Stichprobenwahrscheinlichkeit $-\ell_{\mathcal{B}}(\mathsf{d})$ in Bezug auf die Verteilung $P_{\mathcal{B}}(\,\cdot\,)$ des Netzes. Diese hat günstigerweise auch eine statistische Interpretation: Je höher $\ell_{\mathcal{B}}(\mathsf{d})$ ist, desto besser beschreibt $\mathcal{B}$ die in den Daten steckende gemeinsame Verteilung der Variablen in $\boldsymbol{X}$. Der *MDL*-Wert wird also folgendermaßen berechnet:

$$MDL(\mathcal{B} \,|\, \mathsf{d}) \underset{\text{def}}{=} \frac{\log d}{2}|\mathcal{B}| - \ell_{\mathcal{B}}(\mathsf{d}), \tag{4.58}$$

wobei $|\mathcal{B}|$ die Anzahl der Parameter eines Bayesschen Netzes $\mathcal{B}$ bezeichnet. Wie im ersten Term auf der rechten Seite zu sehen, sind $\frac{1}{2}\log d$ Bits zur Beschreibung eines Parameters in Θ vorgesehen. Ein ähnliches Maß ist das *Bayessche Informationskriterium* (BIC für englisch *Bayesian information criterion*). Beide Maße sind assymptotisch korrekt. Bei steigender Anzahl von Stichprobenelementen konvergiert die gemeinsame Verteilung des Bayesschen Netzes, dass bezüglich der Maße optimal ist gegen die wahre Verteilung des Merkmalsraums.

Friedman et al. [Fri97] weisen darauf hin, das der MDL zum Lernen eines Bayesschen Netz-Klassifikators nicht besonders geeignet ist. Dies hängt vor allem mit dem zweiten Term des MDL zusammen, der beschreibt, wie gut die Stichprobe durch das Netz beschrieben wird. Für einen Klassifikator ist dies jedoch nur ein sekundäres Ziel. Für ihn

ist es entscheidender, die betrachteten Klassen möglichst gut voneinander trennen zu können. Wird der zweite Teil des MDL vermöge

$$\ell_{\mathcal{B}}(\mathsf{d}) =_{\text{def}} \sum_{n=1}^{N} \log P_{\mathcal{B}}(\boldsymbol{x}^{(n)}, \kappa^{(n)}) \tag{4.59}$$

$$= \sum_{n=1}^{N} \log P_{\mathcal{B}}(\kappa^{(n)} \mid \boldsymbol{x}^{(n)}) + \sum_{n=1}^{N} \log P_{\mathcal{B}}(\boldsymbol{x}^{(n)}) \tag{4.60}$$

aufgetrennt, so ist ersichtlich, dass nur das vordere Spaltprodukt die Güte der Klassifikation beschreibt. Das hintere Spaltprodukt beschreibt die für die Klassifikation nebensächliche Anpassung des Bayesschen Netzes an die gemeinsame Verteilung der Merkmale, dominiert jedoch den wichtigen vorderen Teil. Eine Berücksichtigung dieser Beobachtung in einem angepassten Maß, dass sich auf die vorderen Terme beschränkt, ist problematisch, da dieses Maß nicht mehr die für die meisten Lernverfahren essentielle Parameterunabhängigkeit garantieren kann.

Unabhängig von der Wahl des Qualitätsmaßes ist die Suche einer optimalen Strukur NP-vollständig [Chi96]. Im Suchraum aller Strukturen schaffen Algorithmen Abhilfe, die das Optimum zumindest innerhalb einer beschränkten Klasse von Strukturen in polynomialem Aufwand finden[1].

Cooper und Herskovits [Coo92] entwickelten den *K2-Algorithmus*, einen *greedy* Suchalgorithmus zur Suche eines DAG G, der eine nahe dem Maximum gelegene Stichprobenwahrscheinlichkeit erreicht. Der Suchraum dieses Algorithmus ist die Menge aller DAG mit d Knoten. Der elementare Suchschritt besteht darin, einem Knoten einen Elternknoten durch Hinzunahme der entsprechenden Kante hinzuzufügen.

Dabei nutzt der Algorithmus die Tatsache aus, dass die Bewertungsfunktion in das Produkt

$$\ell_{\boldsymbol{\theta}|G^h}(\mathsf{d}) = \prod_{i=1}^{d} \ell_{\boldsymbol{\theta}_i|\Pi^h_{X_i}}(\mathsf{d}_i) \tag{4.61}$$

zerlegbar ist, wobei $\mathbf{\Pi}^h_{X_i}$ die Menge der Elternvariablen von X_i in der Struktur G^h und $\ell_{\boldsymbol{\theta}_i|\Pi^h_{X_i}}(\mathsf{d}_i)$ die logarithmierte Stichprobenwahrscheinlichkeit reduziert auf die bedingten Wahrscheinlichkeiten $P(x_i^{(n)} \mid \boldsymbol{\theta}_i, \Pi^h_{X_i})$ bezeichnen. Die Folge ist, dass zur Maximierung von $\ell_{\boldsymbol{\theta}|G^h}(\mathsf{d})$ die Faktoren des obigen Produktes getrennt maximiert werden können. Der K2-Algorithmus versucht deshalb, für jede Variable X_i eine Elternvariablenmenge zu bestimmen, die den Faktor $\ell_{\boldsymbol{\theta}_i|\Pi^h_{X_i}}(\mathsf{d}_i)$ maximiert.

Der K2-Algorithmus setzt eine spezielle Ordnung der d Variablen als gegeben voraus. Falls bezüglich dieser Ordnung X_i ein Vorgänger von X_j ist, so darf es keine Kante von X_j nach X_i geben, d.h., in einem Ergebnis-DAG des Algorithmus wäre X_i bedingt unabhängig von X_j gegeben der Elternvariablen $\mathbf{\Pi}_{X_i}$. Das ist auch der Grund dafür, dass

[1] Polynomial in der Anzahl d der Zufallsvariablen.

der K2-Algorithmus in der Regel kein globales Maximum der Bewertungsfunktion erreicht. Aufgrund dieser Konsequenz wird deutlich, dass die Konstruktion einer sinnvollen Ordnung keine triviale Aufgabe ist sondern mögliche Unabhängigkeitsannahmen, die sich aus den Daten ergeben könnten, berücksichtigen muss, schon bevor der K2-Algorithmus ausgeführt wird. Die Menge aller Vorgänger von X_i bezüglich der geforderten Ordnung heißt $Pred(X_i)$.

Der Algorithmus arbeitet wie folgt: Zunächst sind die Mengen der Elternvariablen $\mathbf{\Pi}_{X_i}$ leer und für jede Variable X_i wird die dazugehörige Bewertung $\ell_{\boldsymbol{\theta}_i|\emptyset}(\mathsf{d}_i)$ berechnet. Anschließend werden die einzelnen Variablen X_i der Reihe nach bezüglich der zuvor eingerichteten Ordnung durchsucht. Aus der Menge $Pred(X_i)$ wird jene Variable der Elternmenge $\mathbf{\Pi}^h_{X_i}$ hinzugefügt, welche die Bewertung $\ell_{\boldsymbol{\theta}_i|\Pi^h_{X_i}}(\mathsf{d}_i)$ am stärksten verbessert. Dieser Schritt wird wiederholt, so lange sich dadurch eine Verbesserung der Bewertung ergibt. Es folgt der Pseudo-Code des K2-Algorithmus:

ALGORITHMUS: K2

Input:	$X_1, \ldots, X_d$	geordnete Merkmalsmenge
	u	maximale Anzahl von Elternknoten
Output:	$\{\mathbf{\Pi}_{X_i} : i \in \{1, \ldots, d\}\}$	Menge von Elternvariablen für alle X_i

Pseudocode:

1. **FOR** $i = 1$ **TO** d **DO**
   - **a)** $\mathbf{\Pi}_{X_i} = \emptyset$
   - **b)** $score_{old} = \ell_{\boldsymbol{\theta}_i|\emptyset}(\mathsf{d}_i)$
   - **c)** $findmore = \text{true}$
   - **d)** **WHILE** $findmore \wedge |\mathbf{\Pi}_{X_i}| < u$ **DO**
     - **i.** $Y = \text{argmax}_{Z \in Pred(X_i)} \ell_{\boldsymbol{\theta}_i|\mathbf{\Pi}_{X_i} \cup \{Z\}}(\mathsf{d}_i)$
     - **ii.** $score_{new} = \ell_{\boldsymbol{\theta}_i|\mathbf{\Pi}_{X_i} \cup \{Y\}}(\mathsf{d}_i)$
     - **iii.** **IF** $score_{new} > score_{old}$ **THEN**
       - $score_{old} = score_{new}$
       - $\mathbf{\Pi}_{X_i} = \mathbf{\Pi}_{X_i} \cup \{Y\}$
     - **iv.** **ELSE**
       - $findmore = \text{false}$

Die Laufzeit des K2-Algorithmus hat eine obere Schranke von $\mathcal{O}(d^4 N r)$, wobei $N = |\mathsf{d}|$ die Anzahl der Stichprobenelemente und $r = \max_i |D_{X_i}|$ die maximale Anzahl möglicher Werte einer Variablen X_i ist. In der obigen Version wurde zusätzlich eine Schranke u für die maximale Anzahl von Elternknoten einer Variablen eingeführt.

Strukturlernen über lokale Abhängigkeitsuntersuchungen. Wie schon bei der Schätzung der Wahrscheinlichkeitsparameter ist für die Stichprobe d die einzige Information über die *wahre* gemeinsame Verteilung der Merkmale bekannt. Abhängigkeiten zwischen zwei Merkmalen werden nur erkannt, wenn diese durch ausreichend hohe Korrelation der Ausprägungen in den Stichprobenelementen hervortreten. Die Messung der Stärke der

Abhängigkeiten mit geeigneten Korrelationsmaßen oder anhand der Transinformation ist umso genauer, desto mehr Stichprobenelemente vorliegen. Bei kleinen Stichproben besteht die Gefahr, dass Pseudo-Abhängigkeiten erkannt werden, die im Merkmalsraum nicht vorhanden sind.

Eine weitere Möglichkeit, effizient eine *gute* Struktur für ein Bayessches Netz zu bestimmen, besteht darin, als Lösungen nur DAG mit einer eingeschränkten Struktur zu zulassen. Innerhalb dieser Teilmenge aller DAG könnte dann in nichtexponentieller Laufzeit die optimale Struktur bestimmt werden.

Für die baumartigen Bayessche Netz-Klassifikatoren, TAN, die in Kapitel 5 angewendet werden, gibt es ein entsprechendes Verfahren zur Ableitung der optimalen TAN-Struktur für eine gegebene Stichprobe $\mathtt{d}$. Aufgrund seiner Bedeutung in dieser Arbeit soll dieser Algorithmus etwas genauer betrachtet werden.

Der Lernalgorithmus für TAN-Modelle beruht auf einem Algorithmus von Chow und Liu [Cho68] zur Bestimmung der optimalen Struktur von Bayesschen Netzen, die eine baumartige Struktur haben. Dieser Algorithmus wiederrum basiert auf dem *Kruskal-Algorithmus* zur Bestimmung eines minimalen Spannbaums, wobei die Transinformation

$$\mathcal{J}_P(X;\, Y) = \sum_{x,\, y} P(x,\, y) \log \frac{P(x,\, y)}{P(x)P(y)} \tag{4.62}$$

zwischen je zwei Zufallsvariablen X und Y als Kantengewichte verwendet werden. Die Transinformation zwischen X und Y drückt aus, wieviel Information über X in Y steckt und kann daher als Maß der Stärke der Abhängigkeit zwischen X und Y verwendet werden.

Friedman et al. [Fri97] erweiterten dieses Verfahren für TAN-Klassifikatoren (siehe Abschnitt 4.3.2). Da in TAN jede Variable X_i zusätzlich zu einer optionaler Elternvariablen stets von der Klassenvariablen C abhängt, verwenden sie in ihrem Algorithmus *construct-TAN* die *bedingte* Transinformation zwischen je zwei Variablen, gegeben die Klassenvariable, als Kantengewicht. Die bedingte Transinformation zwischen zwei Variablen X und Y gegeben einer dritten Variablen Z ist definiert vermöge

$$\mathcal{I}_P(X;\, Y \mid Z) = \sum_{x,y,z} P(x,\, y,\, z) \log \frac{P(x,\, y \mid z)}{P(x \mid z)P(y \mid z)}. \tag{4.63}$$

Dieser Wert drückt den Gehalt an Information in Y über X aus, wenn der Wert von Z bekannt ist. Die Methode *construct-TAN* hat das folgende Ablaufschema:

ALGORITHMUS: Construct-TAN

Input:	$X_1, \ldots, X_d$	Variablenmenge
	$\mathtt{d}$	Stichprobe
Output:	$\mathcal{C}$	TAN mit maximaler Stichprobenwahrscheinlichkeit $\ell_{\mathcal{B}}\mathtt{d}$

Pseudocode:

1. Berechne $\mathcal{I}_{P_\mathsf{d}}(X_i;\, X_j \,|\, C)$ für alle Variablenpaare mit $i \neq j$.
2. Bilde einen vollständigen, ungerichteten Graphen, in dem die Knoten die Variablen $X_1, \ldots, X_d$ sind, und die Kante zwischen X_i und X_j das Gewicht $\mathcal{I}_{P_\mathsf{d}}(X_i;\, X_j \,|\, C)$ trägt.
3. Berechne den maximalen Spannbaum mit dem Algorithmus von Kruskal.
4. Forme den resultierenden ungerichteten Baum in einen gerichteten Baum um, indem eine beliebige Variable als Wurzel markiert wird, und alle ungerichteten Kanten in einer Weise gerichtet werden, so dass sie von der Wurzel weg zeigen.
5. Konstruiere ein TAN, indem die Klassenvariable C dem Graphen hinzugefügt wird, je eine Kante von C zu den Variablen X_i gezogen wird und für die resultierende Netzstruktur eine Parameterschätzung durchgeführt wird.

SATZ: Sei d eine Stichprobe einer Variablenmenge $\{X_1, \ldots, X_d\}$ mit $N = |\mathsf{d}|$ und $\mathcal{C}$ ein TAN-Klassifikator, der mittels der Methode *Construct-TAN* gewonnen wurde. Sei weiterhin $B_{TAN,d}$ die Menge aller TAN für d Variablen. Dann gilt:

$$\mathcal{C} = \operatorname*{argmax}_{\mathcal{B} \in B_{TAN,d}} \ell_\mathcal{B}(\mathsf{d}). \tag{4.64}$$

Beweis: Zunächst muss darauf hingewiesen werden, dass die verwendete Transinformation $\mathcal{J}_P(X;\, Y)$ über die Differenz aus der Entropie $\mathcal{H}_P(X)$ von X und der bedingten Entropie $\mathcal{H}_P(X \,|\, Y)$ von X gegeben Y berechnet werden kann:

$$\mathcal{J}_P(X;\, Y) = \mathcal{H}_P(X) - \mathcal{H}_P(X \,|\, Y) = \sum_{x,\, y} P(x,\, y) \log \frac{P(x,\, y)}{P(x)P(y)}. \tag{4.65}$$

Sei P_d die empirische Verteilung der Daten d, d.h., $P_\mathsf{d}(\boldsymbol{x}) = \frac{1}{N} \sum_{\boldsymbol{y}} 1_{\boldsymbol{x}}(\boldsymbol{y})$[7]. Die logarithmische Stichprobenwahrscheinlichkeit mit Hilfe der empirischen Verteilung und bei Vertauschung der Summationen umformuliert werden zu:

$$\begin{aligned}
\ell_\mathcal{B}(\mathsf{d}) &= N \sum_{i=1}^{N} \sum_{\substack{x_i \in D_{X_i} \\ \boldsymbol{\pi}_i \in D_{\boldsymbol{\Pi}_i}}} P_\mathsf{d}(x_i,\, \boldsymbol{\pi}_i) \log P_\mathcal{B}(x_i \,|\, \boldsymbol{\pi}_i) && (4.66)\\
&= -N \sum_i \mathcal{H}_{P_\mathsf{d}}(X_i \,|\, \boldsymbol{\Pi}_{X_i}) && (4.67)\\
&= N \sum_i \mathcal{I}_{P_\mathsf{d}}(X_i;\, \boldsymbol{\Pi}_{X_i}) - N \sum_i \mathcal{H}_{P_\mathsf{d}}(X_i). && (4.68)
\end{aligned}$$

[7] Die Funktion $1_x(y)$ heißt *charakteristische Funktion* und hat entweder den Wert Eins, wenn $x = y$ gilt oder den Wert Null in allen anderen Fällen.

Der hintere Teil ist unabhängig von einem konkreten $\mathcal{B}$, so dass die Maximierung von $\ell_{\mathcal{B}}(\mathsf{d})$ gleichbedeutend mit der Maximierung von

$$\sum_i \mathcal{I}_{P_{\mathsf{d}}}(X_i;\, \mathbf{\Pi}_{X_i}) \tag{4.69}$$

ist.

Für den weiteren Beweis wird die Kurzschreibweise $X_{\pi(i)}$ für die einzig mögliche Elternvariable einer Variable X_i abgesehen von der Klassenvariable eingeführt. Der Index $\pi(i)$ ist also eine Abbildung aus $\{1, \ldots, d\}$ nach $\{0, \ldots, d\}$, wobei $\pi(i) = 0$ bedeutet, dass die betreffende Variable keine Elternvariable außer C hat.

Für TAN gilt, dass $\mathbf{\Pi}_{X_i}$ genau zwei Variablen enthält. Gemeinsam mit der eingeführten Kurzschreibweise kann der Ausdruck 4.69 für TAN etwas detailierter aufgeschrieben werden:

$$\sum_{i:\pi(i)>0} \mathcal{I}_{P_{\mathsf{d}}}(X_i;\, \{C,\, X_{\pi(i)}\}) + \sum_{i:\pi(i)=0} \mathcal{I}_{P_{\mathsf{d}}}(X_i;\, \{C\}). \tag{4.70}$$

Mit der Kettenregel

$$\mathcal{I}_P(\boldsymbol{X};\, \boldsymbol{Y},\, \boldsymbol{Z}) = \mathcal{I}_P(\boldsymbol{X};\, \boldsymbol{Z}) + \mathcal{I}_P(\boldsymbol{X},\, \boldsymbol{Y};\, \boldsymbol{Z}) \tag{4.71}$$

für die Transinformation kann Ausdruck 4.70 wiederrum zu

$$\sum_i \mathcal{I}_{P_{\mathsf{d}}}(X_i;\, \{C\}) + \sum_{i:\pi(i)>0} \mathcal{I}_{P_{\mathsf{d}}}(X_i,\, X_{\pi(i)};\, \{C,\, \}) \tag{4.72}$$

vereinfacht werden. Der erste Summand hängt nicht mehr von der Wahl der Funktion π ab, insofern bleibt die Maximierung des zweiten Teils von 4.72. Es ist leicht zu sehen, dass dies genau die Zielfunktion des Kruksal-Algorithmus ist.

4.4.3 Merkmalauswahlverfahren

Ein BN-Klassifikator modelliert die gemeinsame Verteilung von d Zufallsvariablen für jedes der K Klassengebiete. Diese Zufallsvariablen repräsentieren Merkmale, die von den zu klassifizierenden Objekten gemessen oder berechnet werden. Die Stichprobenelemente von d sind beispielsweise Vektoren von Ausprägungen solcher Merkmale und werden *Merkmalsvektoren* genannt. Als gültige Eingaben für den Klassifikator kann für solch einen Vektor eine Anfrage bezüglich der Klassenvariable gestellt werden.

In vielen Anwendungsfällen ist es möglich, eine schier unbegrenzte Anzahl von Merkmalen zu berechnen. Es ist weder sinnvoll noch möglich, all diese Merkmale als Zufallsvariablen in einem Bayessches Netz zu berücksichtigen. Viele der berechenbaren Merkmale werden irrelevant für die Klassifikationsaufgabe sein. Andere Merkmale könnten gegenseitig redundant sein, so dass sie keinen zusätzlichen Nutzen für die Klassifikation haben.

In Einzelfällen jedoch widersprüchlich, was einen negativen Einfluss auf die Klassifikationsleistung haben würde. Des Weiteren liegen in vielen Fällen nicht genügend Stichprobenelemente vor, um die gemeinsame Verteilung einer so großen Anzahl von Zufallsvariablen zu lernen. Weiterhin sprechen auch trotz immer leistungsfähigeren Rechnern Effizienzprobleme gegen diese Maximallösung.

Merkmalauswahlverfahren beschäftigen sich mit der Frage, welche Teilmenge von möglichen Merkmalen bei der Konstruktion eines Klassifikators günstigerweise zu verwenden ist. Diese Aufgabe lässt sich wie folgt formalisieren:

DEFINITION 4.6: *Sei $\boldsymbol{X}$ eine Menge von Merkmalen mit $|\boldsymbol{X}| = D$ und $\mathcal{J} : \boldsymbol{Y} \mapsto \mathbb{R}$ mit $\boldsymbol{Y} \subseteq \boldsymbol{X}$ eine Gütefunktion. Das* Merkmalauswahlproblem *ist die Suche nach jener Teilmenge $\boldsymbol{Y}^* \subseteq \boldsymbol{X}$, so dass gilt:*

$$\boldsymbol{Y}^* = \underset{\boldsymbol{Y} \subseteq \boldsymbol{X}}{\operatorname{argmax}} \mathcal{J}(\boldsymbol{Y}). \tag{4.73}$$

Merkmalauswahlverfahren sind Strategien, Merkmalauswahlprobleme zu lösen bzw. um Merkmalsteilmengen $\boldsymbol{Y}$ zu bestimmen, die einer optimalen Lösung möglichst nahe kommen. In den meisten Fällen handelt es sich um klassische Suchverfahren, deren Suchraum die Menge aller nichtleeren Merkmalteilmengen von $\boldsymbol{X}$ (vergleiche Abbildung 4.6) ist. Der Suchraum lässt sich theoretisch vollständig durchsuchen, wobei die Güte $\mathcal{J}(\,\cdot\,)$ von

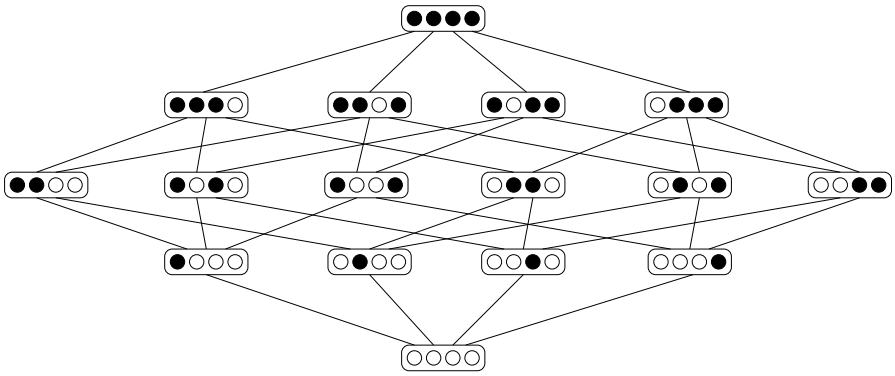


Abbildung 4.6: Suchraum für Merkmalauswahlverfahren. Die Menge $\boldsymbol{X}$ aller Merkmale besteht hier aus 4 Elementen. Jeder Punkt im Suchraum stellt eine nichtleere Teilmenge von $\boldsymbol{X}$ dar, wobei ausgewählte Merkmale durch schwarze Punkte markiert sind.

2^D Merkmalteilmengen bestimmt werden muss. Je nach Gestalt der Gütefunktion und der Anzahl der Merkmale stößt dieses Vorgehen schnell an die Grenzen des Machbaren. Auswege bieten heuristische Suchverfahren und lokale Bewertungsverfahren der einzelnen Merkmale.

In der Fachliteratur werden grundsätzlich zwei Gruppen von Merkmalsauswahlverfahren unterschieden: 1.) Filtermethoden und 2.) *Wrapper*methoden [Rok07, Lan94]. Filter arbeiten unabhängig von einem bestimmten Klassifikationsansatz, indem Merkmale bzw. Merkmalsteilmengen über eine bestimmte Gütefunktion bewertet werden [Yu03]. Wrapper hingegen lernen in jedem Suchschritt genau die Klassifikatoren, die durch die Merkmalsauswahl mit Merkmalen versehen werden sollen, und bewerten Merkmalsteilmengen über eine Kreuzvalidierung und einem daraus abgeleiteten Maß. Filter sind in der Regel wesentlich schneller als Wrappermethoden, jedoch sagen Wrapper viel treffsicherer die "wahre" Güte einer Merkmalsteilmenge in der späteren Anwendung voraus [Uch07]. [8]

Obwohl bestechend schnell, sind Filter nicht geeignet, die richtigen Merkmale für einen Bayesschen Netz-Klassifikator zu bestimmen, da diese vorwiegend dann eingesetzt werden, wenn die Merkmale des Musterraumes in vielfältiger Weise voneinander abhängen und eine Beurteilung der Güte einzelner Merkmale nicht möglich ist. Nach der Auffassung des Autors lässt sich die Güte einer Merkmalteilmenge für die Klassifikation mit Bayesschen Netzen nur dann adäquat bestimmen, wenn die Gütefunktion selbst die Klassifikationsleistung des Bayesschen Netzes direkt oder indirekt misst. Deshalb wird sich im Folgenden auf Wrappermethoden beschränkt.

Wrappermethoden [Das97] sind Suchverfahren im Raum aller Merkmalsteilmengen. Sie unterscheiden sich in ihrer Anfangsbedingung, einer Nachbarschaftsdefinition im Suchraum und einem Qualitätsmaß. Da das Qualitätsmaß $\mathcal{J}(\cdot)$ bei Wrappern in jedem Fall eine zeitintensive Kreuzvalidierung mit einer Stichprobe d durchführt, leuchtet es ein, dass angesichts des Aufwandes für die Berechnung der Güte die Anzahl der besuchten Suchraumpunkte möglichst gering gehalten werden sollte. Deswegen werden meist heuristische oder randomisierte Suchalgorithmen verwendet. Unabhängig davon, welcher Suchalgorithmus verwendet wird, kann dieser als Startpunkt entweder die leere Menge bzw. eine zufällige Menge weniger Merkmale haben, und dieser fortwährend Merkmale hinzufügen, oder aber zunächst die gesamte Menge $\boldsymbol{X}$ bewerten, und aus dieser Merkmale entfernen, um die Güte zu verbessern. Im ersten Fall wird von *Vorwärtsauswahl* (englisch *forward selection*), im zweiten Fall von *Rückwärtsauswahl* (englisch *backward selection*) gesprochen. Da in dem Anwendungsfall in dieser Dissertation lediglich die Vorwärtsstrategie in Frage kommt, werden im Folgenden nur solche Algorithmen vorgestellt. Für diese Algorithmen gibt es entsprechende Rückwärtsvarianten [Das97].

Für einige bekannte heuristische Suchalgorithmen, wie den A^*-Algorithmus oder *Branch and Bound* ist es jedoch zwingend notwendig, dass die Gütefunktion $\mathcal{J}(\cdot)$ *monoton* ist:

DEFINITION 4.7: *Sei* $\mathcal{J} : \boldsymbol{Y} \mapsto \mathbb{R}$ *eine Gütefunktion für Teilmengen der Merkmalmenge* $\boldsymbol{X}$. $\mathcal{J}(\cdot)$ *heißt* monoton*, wenn für beliebige Teilmengen* $\boldsymbol{Y}, \boldsymbol{Z} \in \boldsymbol{X}$ *gilt:*

$$\boldsymbol{Y} \subseteq \boldsymbol{Z} \Longrightarrow \mathcal{J}(\boldsymbol{Y}) \leq \mathcal{J}(\boldsymbol{Z}). \tag{4.74}$$

[8] Die einfachsten Filtermethoden bestehen lediglich darin, alle Merkmale bezüglich eines solchen Maßes zu sortieren und die besten als Ergebnisteilmenge auszugeben.

Nur eine monotone Gütefunktion ermöglicht eine adäquate Restkostenabschätzung zum optimalen Ergebnis beim A^*-Algorithmus. Wie spätestens in Kapitel 5 offensichtlich wird, kann im vorliegenden Anwendungsfall nicht von einer monotonen Gütefunktion ausgegangen werden. Diese würde bedingen, dass die Berücksichtigung aller D Kandidatenmerkmale den besten Klassifikator zur Folge hat. Da viele der verwendeten Kandidatenmerkmale redundant sind, indem sie ähnliche Eigenschaften in leicht abgewandelter Form beschreiben, aber dennoch eine Unschärfe zwischen ihnen besteht, so dass kein sicherer kausaler Zusammenhang besteht, und sie sich so in Einzelfällen widersprechen, ist es der Normalfall, dass nur wenige der D möglichen Merkmale in einer optimalen Merkmalteilmenge vertreten sind.

Ohne den Anspruch, die tatsächlich global optimale Lösung zu finden, bieten sich *sequentielle* Suchverfahren an, schnell (im Mittel) zu einer guten Lösung im Suchraum aller Teilmengen der Größe d zu geraten. Diese Verfahren haben einen Tiefensuche-Charakter, durchsuchen jedoch nicht den gesamten Suchraum, sondern besuchen, ausgehend von der aktuellen Merkmalteilmenge jene, die unter allen benachbarten Merkmalteilmengen die beste Güte besitzt. Hierbei gelten zwei Teilmengen als benachbart, wenn die eine durch Hinzufügen eines Merkmals in die Andere gewonnen werden kann.

Das denkbar einfachste sequentielle Merkmalmengensuchverfahren ist die sequentielle Vorwärtsauswahl, *SFS* (nach seinem englischen Namen *Sequential Forward Selection*) [Whi71]. Ausgehend von der leeren Menge wird in jedem Suchschritt dasjenige Merkmal hinzugefügt, dass die bisherige Güte am stärksten erhöht (siehe Abbildung 4.7).

ALGORITHMUS: SFS

Input:	$\boldsymbol{X}$	Merkmalmenge
	d	maximale Anzahl auszuwählender Merkmale
Output:	$\boldsymbol{Y}$	ausgewählte Merkmalteilmenge

Pseudocode:

1. $\boldsymbol{Y}_0 = \emptyset$
2. **FOR** $k = 1$ **TO** d
   a) $x^+ := \text{argmax}_{x \in \boldsymbol{X} \setminus \boldsymbol{Y}_{k-1}} \mathcal{J}(\boldsymbol{Y}_{k-1} \cup \{x\})$
   b) $\boldsymbol{Y}_k := \boldsymbol{Y}_{k-1} \cup \{x^+\}$
3. **NEXT** k
4. **RETURN** $\text{argmax}_{\boldsymbol{Y}_i, i \in \{1, \ldots, d\}} \mathcal{J}(\boldsymbol{Y}_i)$

Abbildung 4.7: Der SFS-Algorithmus.

Das für den SFS-Algorithmus verwendete Abbruchkriterium hat vor allem historische Gründe. So ließen sich aufgrund der Rechentechnik zu jener Zeit nicht in unbegrenztem Maße Merkmale berücksichtigen. Die Beschränkung auf eine fest vorgegebene Anzahl

von Merkmalen war üblich. Spielen solche Beschränkungen keine Rolle, so lässt sich das Abbruchkriterium dahingehend ändern, dass so lange Merkmale hinzugefügt werden, so lange dies zu einer Verbesserung der Güte führt.

Der SFS-Algorithmus funktioniert nur dann zufriedenstellend, wenn die Merkmale in $\boldsymbol{X}$ gegenseitig unabhängig sind, und demnach jedes Merkmal einen bestimmten, von den anderen Merkmalen unabhängigen Beitrag zur Klassifikationsleistung liefert. Gerade für Bayessche Netz-Klassifikatoren, die hauptsächlich in Fällen hochgradig abhängiger Merkmale eingesetzt werden, leidet der SFS-Algorithmus darunter, zu schnell in lokale Optima zu geraten, ohne Mittel zu besitzen, sich aus diesen zu befreien.

Um die Tendenz, in lokale Optima zu laufen, zu vermindern, wurden mehrere Anpassungen des SFS-Algorithmus vorgeschlagen. Eine *allgemeine* sequentielle Vorwärtsauswahl (GSFS-Algorithmus) [Kit78] fügt in jedem Schritt g Merkmale gleichzeitig der aktuellen Menge zu, und zwar jene Teilmenge der Größe g, die hinzugefügt eine maximale Steigerung der Güte ermöglicht. Den Teil an Vorhersagekraft, der sich aus der Modellierung von Abhängigkeiten zwischen jeweils g Merkmalen ergibt, wird in diesem Verfahren berücksichtigt. Im Mittel kann deshalb erwartet werden, dass die aus dem GSFS-Algorithmus resultierenden Merkmalteilmengen eine höhere Güte besitzen als die SFS-Lösungen. Jedoch erhöht sich der Rechenaufwand bereits für kleine g gewaltig. Statt $\mathcal{O}(dD)$ benötigt der GSFS bereits $\mathcal{O}(\frac{d}{g}D^g)$ Suchschritte.

Eine weitere Möglichkeit, zu frühe und schlechte lokale Gütemaxima zu vermeiden, ist das Zulassen des Löschens von Merkmalen aus der aktuellen Merkmalteilmenge, die zu einem früheren Zeitpunkt eingefügt wurden. Das dieses Vorgehen tatsächlich eine Verbesserung verspricht, zeigt folgendes Gedankenspiel: Seien zwei Merkmale A und B zu einem beliebigen vergangenen Zeitpunkt ausgewählt worden, wobei beide in etwa die gleiche Eigenschaft beschreiben und A etwas besser als B zwischen den Klassen diskriminiert (und deshalb vor B eingefügt wurde). Im aktuellen Schritt wird Merkmal C eingefügt, und es ergibt sich, dass zwischen B und C eine Abhängigkeit besteht. Diese beiden Merkmale beschreiben nun die bestimmte Eigenschaft zusammen besser als A. Darüber hinaus widersprechen sich die Werte von A auf der einen und B und C auf der anderen Seite in Einzelfällen, was die Erkennungsleistung eines Klassifikators, der alle drei Merkmale berücksichtigt, reduziert. Das Löschen von A wäre eine Möglichkeit, bereits getroffene Entscheidungen rückgängig zu machen und so einen Ausweg aus lokalen Optima zu finden.

Ein früher Algorithmus, der das Löschen von Merkmalen zulässt, ist der $PTA(l,r)$-*Algorithmus* (für englisch: *Plus l, take away r*) [Ste76]. Wie der Name bereits vorgibt, werden in jedem Schritt zunächst nacheinander die l besten Einzelmerkmale eingefügt und anschließend genau r Einzelmerkmale entfernt, wobei jeweils das Merkmal gelöscht wird, dessen Löschen zu einer maximalen Güte führt. Obwohl das Risiko schlechter lokaler Optima reduziert wird, durchstreift diese Methode den Suchraum recht starr in immer gleichen Schrittlängen, wie durch die Parameter l und r vorgegeben.

Die Weiterentwicklung dieses Prinzips wurde 1994 von Pudil et. al [Pud94] unter dem Namen *Sequential Forward Floating Selection* (kurz: *SFFS-Algorithmus*) vorgestellt. Da dieser Algorithmus im Rahmen dieser Dissertation eingesetzt wird, soll das Funktionsprinzip etwas genauer anhand des Pseudocodes illustriert werden (siehe Abbildung 4.8): Der SFFS-Algorithmus besteht im Wesentlichen aus zwei sich wiederholenden Schrit-

---

ALGORITHMUS: SFFS

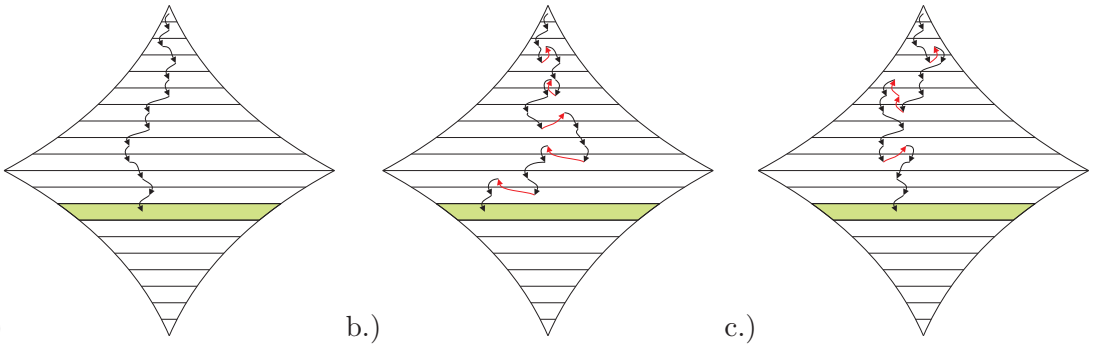
Input:	$\boldsymbol{X}$	Merkmalmenge
	d	maximale Anzahl auszuwählender Merkmale
Output:	$\boldsymbol{Y}$	ausgewählte Merkmalteilmenge

Pseudocode:

1. $\boldsymbol{Y}_0 = \emptyset$; $k = 1$
2. **Add:**
   a) **IF** $k = d + 1$ **THEN GOTO** Stop
   b) $x^+ := \text{argmax}_{x \in \boldsymbol{X} \setminus \boldsymbol{Y}_{k-1}} \mathcal{J}(\boldsymbol{Y}_{k-1} \cup \{x\})$
   c) $\boldsymbol{Y}_k := \boldsymbol{Y}_{k-1} \cup \{x^+\}$
3. **Remove:**
   a) $x^- := \text{argmax}_{x \in \boldsymbol{Y}_k} \mathcal{J}(\boldsymbol{Y}_k \setminus \{x\})$
   b) **IF** $\mathcal{J}(\boldsymbol{Y}_k \setminus \{x^-\}) > \mathcal{J}(\boldsymbol{Y}_(k-1))$ **THEN**
      i. $\boldsymbol{Y}_{k-1} := \boldsymbol{Y}_k \setminus \{x^-\}$
      ii. **GOTO** Remove
   c) **ELSE GOTO** Add
4. **RETURN** $\text{argmax}_{\boldsymbol{Y}_k : k=1,\ldots,d} \mathcal{J}(\boldsymbol{Y}_k)$

Abbildung 4.8: Der SFFS-Algorithmus.

ten. Im ersten Schritt wird analog zum SFS-Algorithmus das Merkmal der aktuellen Teilmenge $\boldsymbol{Y}_{k-1}$ hinzugefügt, dass deren Güte maximal verbessert. Im zweiten Schritt werden so lange Merkmale aus der aktuellen Teilmenge $\boldsymbol{Y}_k$ entfernt, solange die resultierende Güte höher ist als die der bisher besten Teilmenge gleicher Größe. Der SFFS-Algorithmus löscht also im Gegensatz zum PTA(l, r)-Algrithmus nur bei Bedarf Merkmale und wird so flexibel auf vielversprechende Punkte im Suchraum gelenkt (Abbildung 4.9 zeigt schematisch die unterschiedlichen Suchpfade des SFS-, des PTA(l, r)- und des SFFS-Algorithmus). Diesem Vorteil steht der theoretisch hohe Rechenaufwand entgegen. Im schlimmsten Fall durchsucht der SFFS-Algorithmus alle Teilmengen von $\boldsymbol{X}$ bis zur maximalen Kardinalität d ($\mathcal{O}(2^d)$). In der Praxis wird dies nur sehr selten auftreten, da auch der SFFS-Algorithmus nicht völlig vor dem Gefangensein in lokalen Optima geschützt ist. In einer vergleichenden Studie von Kudil und Sklansky [Kud00] werden dem SFFS-Algorithmus überdurchschnittliche Leistungen bescheinigt.



a.) b.) c.)

Abbildung 4.9: Schematische Darstellung der Suchpfade für die Merkmalauswahlverfahren: a.) SFS, b.) PTA(l, r) und c.) SFFS

Gütefunktion mit Kreuzvalidierung. Während einer k-fachen Kreuzvalidierung wird die Stichprobe k mal in Paare $(\mathsf{d}^{(i)}_{lern}, \mathsf{d}^{(i)}_{test})$ mit $i = 1 \ldots k$ aufgeteilt. Es gilt jeweils

$$\mathsf{d} = \mathsf{d}^{(i)}_{lern} \uplus \mathsf{d}^{(i)}_{test} \tag{4.75}$$

und

$$\mathsf{d} = \biguplus_i \mathsf{d}^{(i)}_{lern} = \biguplus_i \mathsf{d}^{(i)}_{test}. \tag{4.76}$$

Außerdem ist die Kardinalität eines jeden $\mathsf{d}^{(i)}_{test}$ ungefähr $\frac{1}{k}$ [9].

Ein geeignete Gütefunktion ist die Summe $LSAP$ der logarititthmierten a posteriori Wahrscheinlichkeiten für eine *bekannte* Klasse $\kappa^{(n)}$ eines jeden Stichprobenelements $\boldsymbol{m}^{(n)}$. Soll also die Qualität einer Merkmalsteilmenge $\boldsymbol{Y}$ bestimmt werden, so wird zunächst die gesamte Stichprobe d auf die Merkmale aus $\boldsymbol{Y}$ reduziert. Die reduzierte Stichprobe heißt ${}_{\boldsymbol{Y}}\mathsf{d}$. Anschließend wird, wie oben beschrieben, die Aufteilung der reduzierten Stichprobe vorgenommen, und für jedes $i = 1 \ldots k$ ein BN-Klassifikator $\mathcal{C}^{(i)}$ unter Verwendung von ${}_{\boldsymbol{Y}}\mathsf{d}^{(i)}_{lern}$ trainiert. Mit der Datenaufteilung und den entsprechenden Modellen wird die folgende Summe berechnet:

$$\mathcal{J}(\boldsymbol{Y} \mid \mathsf{d}) = \sum_{i=1}^{k} \sum_{j=1}^{|{}_{\boldsymbol{Y}}\mathsf{d}^i_{test}|} P_{\mathcal{C}^{(i)}}(\kappa^{(j)} \mid \boldsymbol{m}^{(j)}). \tag{4.77}$$

Dabei bezeichnet $P_{\mathcal{C}^{(i)}}(\cdot)$ die durch $\mathcal{C}^{(i)}$ definierte Wahrscheinlichkeitsverteilug. Ein optimaler Wert dieses Qualitätsmaßes ist Null, denn dann wird in jedem Fall der *richtigen* Klasse eines Merkmalsvektors die a posteriori Wahrscheinlichkeit 1 zugewiesen. Die Gütefunktion drückt also das Vermögen des Klassifikators aus, mit den Merkmalen aus $\boldsymbol{Y}$ die Stichprobenelemente einer Klasse κ zu erkennen bzw. Elemente anderer Klassen in Bezug auf κ abzuweisen.

[9] Bei Stichproben, deren Größe nicht durch k teilbar ist, weicht genau eine Teilteststichprobe von $\frac{1}{k}$ ab.

Im Anwendungsfall dieser Arbeit [Pud08], der Klassifikation von Transkriptionsfaktorbindungsstellen, enthalten die Stichproben im Allgemeinen sehr wenige Beispiele der TFBS-Klasse und sehr viele Beispiele beliebiger anderer Merkmalsvektoren (der Negativklasse). Es wurde in den Versuchen, die in Abschnitt 5.3 vorgestellt werden, beobachtet, dass in diesem Fall der Suchalgorithmus dazu tendiert, Merkmalsteilmengen weiterzuverfolgen, die nahezu perfekt die Elemente der Negativklasse erkennen, dafür weniger gut die Elemente der TFBS. Der wahrscheinliche Grund dafür ist, dass die Summe hauptsächlich durch die große Anzahl der Negativklasse dominiert wird, und die wenigen *schlechteren* Summanden der TFBS-Klasse nicht störend sind. Dieses Verhalten ist natürlich unbefriedigend, denn gerade für die Modellierung der Bindungsstellen sind charakteristische Merkmale gesucht.

In Abschnitt 5.3 wird aus diesem Grund eine gewichtete Variante obiger Summe verwendet. Seien dazu $\#\kappa$ die Anzahl von Elementen der Teilstichprobe d, die der Klasse κ angehören. Dann lautet die verwendete Variante des Qualitätsmaßes:

$$\mathcal{J}(\boldsymbol{Y} \mid \mathsf{d}) = \sum_{\kappa=1}^{K} \frac{1}{\#\kappa} \sum_{i=1}^{k} \sum_{\boldsymbol{m} \in \boldsymbol{Y}\, \mathsf{d}_{test}^{(i)})} 1_{\kappa}(\kappa(\boldsymbol{m})) P_{\mathcal{C}^{(i)}}(\kappa(\boldsymbol{m}) \mid \boldsymbol{m}^{(i)}), \tag{4.78}$$

wobei $1(\kappa(\boldsymbol{m}) = \kappa)$ genau dann den Wert 1 hat, wenn $\kappa(\boldsymbol{m}) = \kappa$ zutrifft. Durch die Wichtung mit $\frac{1}{\#\kappa}$ wird die Bedeutung einer seltenen Klasse bei der Gütebestimmung von $\boldsymbol{Y}$ erhöht.

4.4.4 Diskretisierung kontinuierlicher Merkmale

Das gesamte Kapitel betrachtete bisher ausschließlich Bayessche Netze mit diskreten Zufallsvariablen. Diese können selbstverständlich auch nur Merkmale mit endlichen und diskreten Wertemengen modellieren. In vielen Anwendungen, insbesondere auch diejenige, die im folgenden Kapitel vorgestellt wird, müssen jedoch Eigenschaften der Muster in Merkmalen berücksichtigt sein, die eigentlich einen kontinuierlichen Wertebereich haben.

Diskretisierungsverfahren zerlegen den kontinuierlichen Wertebereich eines solchen Merkmals in eine endliche Anzahl von zusammenhängenden Intervallen. Unter Verwendung dieser Zerlegung weist ein *Diskretisierer* jeder reellen Merkmalsausprägung $m \in \mathbb{R}$ den Index jenes Intervalls zu, in dem dieser Wert liegt.

DEFINITION 4.8: *Sei $\boldsymbol{t} = (t_1, \ldots, t_D)$ mit $t_i \in \mathbb{R}$ und $t_1 < \ldots < t_D$ ein Vektor von Intervallgrenzen. Ein* Diskretisierer $\mathrm{discr}_{\boldsymbol{t}}$ *für $\boldsymbol{t}$ ist eine Abbildung*

$$\mathrm{discr}_{\boldsymbol{t}} \;=\; \left\{ \begin{array}{lcl} \mathbb{R} & \rightarrow & \{0, 1, \ldots, D\} \\ a & \mapsto & \mathrm{discr}_{\boldsymbol{t}}(a) \end{array} \right. \quad \text{mit} \tag{4.79}$$

$$\text{discr}_{\boldsymbol{t}}(a) \quad = \quad \begin{cases} 0 & , falls \quad a \in (-\infty, t_1] \\ & \cdots \\ d & , falls \quad a \in (t_d, t_{d+1}] \\ & \cdots \\ D & , falls \quad a \in (t_D, \infty) \end{cases} . \tag{4.80}$$

Der Diskretisierung von kontinuierlichen Merkmalen sollte bei der Konstruktion von Bayesschen Netz-Klassifikatoren einige Aufmerksamkeit geschenkt werden, da die Wahl der Intervallgrenzen im hohen Maße den Nutzen des diskretisierten Merkmals beeinflussen kann.

Abgesehen von einfachsten Methoden, die den kontinuierlichen Wertebereich in äquidistante Intervalle zerlegen, verwenden die meisten Diskretisierungsverfahren zur Bestimmung günstiger Intervallgrenzen eine etikettierte Stichprobe

$$\mathsf{d} = \{(a^{(1)}, \kappa^{(1)}), (a^{(2)}, \kappa^{(2)}), \ldots, (a^{(N)}, \kappa^{(N)})\}$$

des zu diskretisierenden Merkmals A, mit der Besonderheit, dass die Stichprobenelemente bezüglich der Merkmalswerte $a^{(n)}$ aufsteigend sortiert sind.

Ein einfaches stichproben-basiertes Diskretisierungsverfahren ist unter seinem englischen Namen *Equal Frequency Bin discretizer* bekannt. Die Stichprobenelemente werden dabei bezüglich der kontinuierlichen Merkmalsausprägung sortiert. Intervallgrenzen werden so festgelegt, dass in jedem der resultierenden Intervalle die gleiche Anzahl an Stichprobenelementen liegt. Ein Nachteil dieses Verfahrens ist es, dass sich zunächst für eine Anzahl von Intervallen entschieden werden muss. Eine solche Entscheidung beeinflusst die Güte eines Merkmals nachhaltig und kann nicht auf andere kontinuierliche Merkmale übertragen werden.

Im Rahmen dieser Dissertation wurde das Diskretisierungsverfahren von von Fayyad und Irani [Fay93] eingesetzt. Dieses Verfahren arbeitet rekursiv auf sortierten Elementen einer etikettierten Stichprobe und bestimmt jeweils die Intervallgrenze, welche die Entropie maximal reduziert. Es bricht ab, wenn die weitere Unterteilung eines Teilintervalls keinen Gewinn in Bezug auf ein Abbruchkriterium verspricht.

Zur Erläuterung des Verfahrens sei A ein kontinuierliches Merkmal und d eine Stichprobe wie zuvor definiert. Jeder der Merkmalsausprägungen $a^{(n)}$ in der Stichprobe wird zunächst auf seine Eignung als Intervallgrenze zur Unterteilung eines kontinuierlichen Wertebereiches hin untersucht. Ausgewählt wird schließlich jene Grenze t_{max} welche die Entropie der Stichprobe maximal reduziert, d.h., jene Grenze, welche die Differenz $\Delta_{\mathcal{H}}$ der Entropie ohne Unterteilung und mit Unterteilung maximiert:

$$t_{max} \quad = \quad \underset{t=a^{(1)},\ldots,a^{(N)}}{\text{argmax}} \; \Delta_{\mathcal{H}} \tag{4.81}$$

$$= \quad \underset{t=a^{(1)},\ldots,a^{(N)}}{\text{argmax}} \; \mathcal{H}(\mathsf{d}) - (\frac{|\mathsf{d}_{\leq t}|}{|\mathsf{d}|}\mathcal{H}(\mathsf{d}_{\leq t}) + \frac{|\mathsf{d}_{>t}|}{|\mathsf{d}|}\mathcal{H}(\mathsf{d}_{>t}), \tag{4.82}$$

wobei $\mathsf{d}_{\leq t}$ der Teil der Stichprobe ist, für dessen Merkmalsausprägungen $a^{(n)} \leq t$ gilt, und entsprechend $\mathsf{d}_{>t}$ die Teilstichprobe mit Merkmalsausprägungen, die größer sind als t. Die Entropie einer (Teil)stichprobe d berechnet sich dabei durch

$$\mathcal{H}(\mathsf{d}) \quad = \quad -\sum_{\kappa=1}^{K} p_\kappa \log_2 p_\kappa, \tag{4.83}$$

wobei p_κ die relative Häufigkeit für Lernbeispiele der κ-ten Klasse in der Stichprobe d ist. Nachdem mittels der ersten Intervallschranke t eine Zerlegung der gesamten Stichprobe durchgeführt wurde, wird in derselben Weise mit den entstandenen Teilstichproben $\mathsf{d}_{\leq t}$ und $\mathsf{d}_{>t}$ verfahren. Der Algorithmus bricht für eine Teilstichprobe d ab, falls die Entropiereduktion $\Delta_\mathcal{H}$ der optimalen Intervallgrenze den folgenden Grenzwert unterschreitet:

$$\Delta_\mathcal{H} \quad < \quad \frac{\log_2(|\mathsf{d}|-1) + \log_2(3^K-2) - (K\mathcal{H}(\mathsf{d}) - K_{\leq t}\mathcal{H}(\mathsf{d}_{\leq t}) - K_{>t}\mathcal{H}(\mathsf{d}_{>t}))}{|\mathsf{d}|}. \tag{4.84}$$

Dabei bezeichne K die Anzahl der verschiedenen Klassen in d sowie $K_{\leq t}$ und $K_{>t}$ die Anzahl vorkommender Klassen in den Teilstichproben $\mathsf{d}_{\leq t}$ und $\mathsf{d}_{>t}$. Wie leicht zu ersehen, handelt es sich bei dem Abbruchkriterium um die minimale Beschreibungslänge (MDL), die bereits im Zusammenhang mit Strukturlernalgorithmen für Bayessche Netze auf Seite 84 eingeführt wurde.

Zuletzt werden die festgehaltenen Intervalgrenzen aufsteigend sortiert. Das Ergebnis ist der gesuchte Vektor $\boldsymbol{t}$ mit $t_1 <, \ldots, < t_D$ von D Intervallgrenzen, die $D+1$ Intervalle definieren und im Folgenden die Diskretisierungsvorschrift $\text{discr}_{\boldsymbol{t}}$ festlegen.

Kapitel 5

Suche und Modellierung charakteristischer TFBS-Merkmale

In den einleitenden Worten der Abschnitte 3.1 und 3.3 wurden zwei Erklärungsmodelle für die niedrige Vorhersagekraft gegenwärtiger Modellierungsansätze für Transkriptionsfaktorbindungsstellen (TFBS) diskutiert. Der eine Standpunkt sieht in den hohen Anzahlen von Vorhersagen keine Falsch-Positiven, sondern eine biologische Realität, in der Transkriptionsfaktoren eben überall bei Vorhandensein einer günstigen Sequenz binden, unabhängig davon, ob diese Bindung eine Funktion, also eine Beeinflussung der Transkription zur Folge hat. Demnach wären die hohen Falsch-Positiv-Fehlerraten gegenwärtiger TFBS-Modelle nicht Ausdruck einer unzureichenden Charakterisierung der Bindungssequenzen. Die Erforschung der transkriptionellen Regulation sollte sich dann auf die Wechselwirkungen gemeinsam agierender Transkriptionsfaktoren, der Beschreibung ihrer Dynamik in regulativen Netzwerken, sowie der Suche optimaler gemeinsamer TFBS-Module konzentrieren. Abschnitt 5.5 in diesem Kapitel und Kapitel 6 widmen sich der Erkennung dieser TFBS-Module.

Der andere Standpunkt jedoch übt Kritik am Beschreibungsvermögen gegenwärtiger TFBS-Modellansätze, insbesondere an den weit verbreiteten PWM-Modellen. (siehe Seite 37). Angesichts der hochgradig komplexen Strukturen, die sowohl die DNA als auch das bindende Protein im dreidimensionalen Raum ausbilden und der vielfältigen biochemischen Einflussfaktoren in der Zelle erscheint es in der Tat kühn, die Beschreibung der Bindungsstellen auf die Häufigkeiten der Nukleotide an den einzelnen Positionen bekannter Bindungsstellen zu beschränken.

Leistungsfähigere Klassifikatoren könnten dadurch gewonnen werden, dass die zugrunde liegenden TFBS-Modelle die Einbeziehung weiterer auffälliger Eigenschaften ermöglichten. Der Modellierung immer komplexerer Eigenschaften steht jedoch ein erhöhter Aufwand für das Lernen und das Anwenden der Modelle gegenüber. Nicht nur, dass das entsprechende Wissen über besondere Eigenschaften der Bindungsstellen vorhanden sein muss. Die Ausprägungen dieser Merkmale müssen zudem in effektiver Weise mess- bzw. berechenbar sein. Erforderte ein äußerst detailliertes TFBS-Modell etwa, dass zur Suche von TFBS in einer DNA-Sequenz von dieser eine NMR-Struktur angefertigt werden müsse, so stünde dieser Aufwand in keinem Verhältnis zu den zu erwartenden niedrigen Klassifikationsfehlerraten. Ein weiteres Risiko bei einer detaillierteren TFBS-Modellierung ist die Überanpassung der Modelle an die unter Umständen wenigen

Lernbeispiele. Die detaillierteren Modelle sollten aus diesem Grund möglichst wenige zu lernende Parameter besitzen.

Es sind im Wesentlichen zwei vermeintliche Unzulänglichkeiten, die dem PWM-Ansatz zugeschrieben werden: 1.) die ausschließliche Berücksichtigung der Spalten eines Alignments bekannter TFBS und 2.) die Annahme statistischer Unabhängigkeit zwischen diesen Spalten. Diese Vereinfachungen tragen jedoch zur enormen Effektivität beim Lernvorgang und bei der Suche in Sequenzen bei. Zudem können PWM schon mit Lerndaten geringen Umfangs gelernt werden. Aufgrund ihrer einfachen Struktur lässt sich die statistische Signifikanz der PWM-Treffer exakt berechnen [Sta89a].

In dem Artikel [Pud05] habe ich einen TFBS-Modellierungsansatz veröffentlicht, der die beiden unterstellten Schwächen einer Gewichtsmatrix zu vermeiden versucht, gleichzeitig jedoch ähnlich effektiv trainierbar und anwendbar ist. Dieser Ansatz entfernt sich von der Vorstellung einer TFBS als einfache, zusammenhängende Teilsequenz innerhalb eines Promotors. Vielmehr werden Bindungsstellen allgemein als Vektoren von Merkmalsausprägungen eines gegebenen Satzes von Merkmalen aufgefasst. Diese Merkmale werden durch diskrete Zufallsvariablen in einem BN-Klassifikator repräsentiert. Die Parameter der (eventuell bedingten) Wahrscheinlichkeitsverteilungen dieser Zufallsvariablen werden anhand einer etikettierten Lernstichprobe gelernt. Zur Klassifikation von DNA-Sequenzen werden analog zu *log-odds*-Bewertungen Statistiken berechnet, für die ebenfalls die Bestimmung der statistischen Signifikanz ermöglicht wird.

Dieses Kapitel wird diesen TFBS-Modellierungsansatz im Folgenden vorstellen. Da es für die folgenden Ausführungen sinnvoll ist, eine möglichst kurze Bezeichnung für die hergeleiteten TFBS-Modelle zur Verfügung zu haben, sei hiermit der Name *TFBS-BN* für diese Modelle vereinbart. In Abschnitt 5.1 werden die verschieden Klassen von Merkmalen eingeführt, die bisher in dem TFBS-Modellierungssystem implementiert sind. Abschnitt 5.2 beschäftigt sich mit der Modellierung dieser Merkmale in Bayesschen Netzen. Weiterhin beschreibt der Abschnitt, wie TFBS-BN zur Suche von TFBS in einer langen DNA-Sequenz eingesetzt werden und wie ein TFBS-BN trainiert wird. Abschnitt 5.3 untersucht schließlich die Leistungsfähigkeit des vorgestellten Modellierungsansatzes in Form eines Vergleiches mit PWM-Modellen. Abschnitt 5.4 enthält einige Angaben zur softwaretechnischen Realisierung des TFBS-Modellierungssystems. Insbesondere wird dort auf eine Web-Anwendung, *BioBayesNet* [Nik07], eingegangen, die auf Basis dieses Systems entwickelt wurde. In Abschnitt 5.5 wird ein Ansatz entwickelt, die TFBS-BN als Ausgabeverteilungen von HMM-Zuständen einzusetzen. Zu guter Letzt werden in Abschnitt 5.6 die Ergebnisse dieser Teilarbeit diskutiert und mögliche Fortsetzungen und Verbesserungen vorgeschlagen.

5.1 Merkmalsklassen

In Unterabschnitt 4.3.1 auf Seite 72 wurden bereits die Begriffe *Muster* bzw. *Musterraum* und *Merkmal* bzw. *Merkmalsraum* definiert. Wie fügt sich nun die konkrete Klassifika-

tionsaufgabe, das Erkennen von TFBS, in diesen Formalismus ein?

Die Klassifikationsaufgabe besteht darin, für eine beliebige Position einer DNA-Sequenz zu entscheiden, ob an dieser Stelle eine TFBS ist oder nicht. Unser Problemkreis Ω, der Musterraum, besteht deshalb aus allen Positionen aller DNA-Sequenzen. Ein konkretes Muster $\boldsymbol{\omega}$ ist eine bestimmte Position in einer bestimmten DNA-Sequenz. Solch ein Muster enthält die Gesamtheit an Informationen zu dieser Sequenz-Position. Selbstverständlich gehört dazu die Nukleotidsequenz in der Umgebung dieser Position. Ein Muster enthält jedoch weitaus mehr Komponenten, beispielsweise Informationen über das nachfolgende proteinkodierende Gen (Gewebespezifität dieses Gens, Expressionsprofile), gesichertes Wissen über weitere TFBS in der Umgebung der Positionen oder gemessene Bindungsaffinitäten.

Der hier zu entwickelnde Klassifikator soll jedes dieser Muster einer von zwei Klassen zuordnen: entweder der Klasse von TFBS $\Omega_1 := \Omega_{\mathsf{TFBS}}$ oder in die Klasse $\Omega_2 := \Omega_{\neg\mathsf{TFBS}}$ aller anderen Sequenzen. Letztere Klasse wird auch *Hintergrundklasse* genannt, die ihr zugeordneten Muster heißen *Hintegrundmuster*. Gemäß den Erläuterungen auf Seite 72 werden dazu dem Muster Merkmale entnommen bzw. aus ihm berechnet, mit denen sich die Klassifikationsaufgabe möglichst gut lösen lässt.

Dieser Abschnitt wird verschiedene Typen von Merkmalen vorstellen, die *Merkmalsklassen*, die bis zum jetzigen Zeitpunkt in dem hier vorgestellten TFBS-Detektionssystem entwickelt wurden.

DEFINITION 5.1: *Eine* **parametrisierte TFBS-Merkmalsklasse** $\mathcal{M}$ *ist eine Menge von Merkmalen* M*. Sie wird charakterisiert durch einen Satz* $\boldsymbol{\Phi}$ *von freien Parametern* Φ_i *mit* $i \in \{1, \ldots, \nu\}$ *und Wertemengen* D_{Φ_i}*. Ein* Merkmal $M \in \mathcal{M}$ *ergibt sich durch Belegung der freien Parameter mit konkreten Werten* $\phi_i \in D_{\Phi_i}$*.*

Im folgenden wird abkürzend der Begriff *Merkmalsklasse* verwendet. Die freien Parameter Φ_i einer Merkmalsklasse $\mathcal{M}$ beschreiben die Abbildungscharakteristik seiner Merkmale in den durch die Merkmalsklasse gesetzten Grenzen.

Bevor mit der Einführung konkreter Merkmalsklassen fortgefahren wird, noch eine Anmerkung zur verwendeten Notation: Muster werden zwar als Gesamtheit aller Informationen verstanden, die zu einer gegebenen DNA-Position vorliegen. Da jedoch die genaue Struktur eines Musters $\boldsymbol{\omega}$ nicht genauer definiert werden kann, wird es im Folgenden stets mit der gemeinten Position $i_{\boldsymbol{\omega}}$ im Genom bzw. in einer DNA-Sequenz gleichgesetzt, der so genannten *Referenzposition*. Die Konsequenz ist, dass, obwohl Merkmale Funktionen aus dem Musterraum Ω sind, sie der Einfachheit halber als Funktionen aus dem Indexbereich einer DNA-Sequenz definiert werden.

Eine zweite Vorbemerkung betrifft die Gemeinsamkeiten aller hier vorgestellten Merkmalsklassen. Obwohl der Modellierungsansatz grundsätzlich offen gegenüber beliebigem Wissen zu einem Muster ω bzw. zu einer Referenzposition $i_{\boldsymbol{\omega}}$ ist, lassen sich die Ausprägungen der folgenden Merkmale allein aus der Sequenz in der Umgebung von $i_{\boldsymbol{\omega}}$

berechnen. Genauer benötigt ein Merkmal M zur Berechnung eines Wertes an Position i_ω die Sequenzinformation des Intervalls $[i_\omega + \mathsf{le}_M, i_\omega + \mathsf{ri}_M]$ mit $\mathsf{le}_M, \mathsf{ri}_M \in \mathbb{Z}$ und $\mathsf{le}_M \leq \mathsf{ri}_M$ (siehe Abbildung 5.1). Die beiden Parameter le_M und ri_M gehören zu den freien Parametern Φ einer jeden Merkmalsklasse. Da im Weiteren keine Verwechslungsgefahr zwischen den Parametern verschiedener Merkmale bestehen, wird auf die Nennung des Merkmals M im Index der Parameter verzichtet.

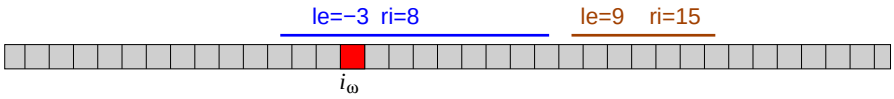


Abbildung 5.1: Verwendeter Sequenzintervall von Merkmalen. Blau: Ausgehend von i_ω verwendet dieses Merkmal den Sequenzintervall $[i_\omega - 3, i_\omega + 8]$, Braun: dieses Merkmal verwendet den Intervall $[i_\omega + 9, i_\omega + 15]$

5.1.1 $\mathcal{M}_{PWM}$: Nukleotide an definierten Positionen

Trotz aller Bemühungen, der DNA-Sequenz einer TFBS versteckte aber höchst trennscharfe Eigenschaften zu entlocken, gelten die Nukleotide an bestimmten Positionen der TFBS aufgrund offensichtlicher Sequenzähnlichkeiten zwischen bekannten TFBS eines Transkriptionsfaktors als wichtige Merkmale für die Klassifikation.

Ein Merkmal M der Merkmalsklasse $\mathcal{M}_{PWM}$ gibt für Referenzposition i_ω der DNA-Sequenz $s_1 s_2 \ldots s_L$ das Nukleotid an einer bestimmten, relativ zu i_ω benannten Stelle an. Welche Stelle gemeint ist, wird durch den einzigen Parameter $\mathsf{pos} \in \mathbb{Z}$ der Merkmalsklasse $\mathcal{M}_{PWM}$ angegeben[1].

Wertebereich aller Merkmale $M \in \mathcal{M}_{PWM}$ ist $\Sigma_{DNA} \cup \{\perp\}$, wobei $\perp$ für *nicht definiert* steht, etwa in dem Falle, dass die aus pos und i_ω errechnete Position außerhalb des Indexbereichs der Sequenz liegt oder falls die Nukleotidsequenz unvollständig vorliegt[2]. Das Gesagte soll abschließend in einer Definition zusammengefasst werden.

DEFINITION 5.2: *Sei $s_1 \ldots s_L$ mit $s_i \in \Sigma_{DNA}$, $1 \leq i \leq L$ eine DNA-Sequenz. Ein $\mathcal{M}_{PWM}$-Merkmal M ist ein Merkmal mit einem freien Parameter $\mathsf{pos} \in \mathbb{Z}$, dem Wertebereich $D_M = \Sigma_{DNA} \cup \{\perp\}$, und folgender Abbildungsvorschrift:*

$$M(i_\omega) = \begin{cases} s_{i_\omega + \mathsf{pos}} & : \quad 1 \leq i_\omega + \mathsf{pos} \leq L \wedge s_{i_\omega + \mathsf{pos}} \in \Sigma_{DNA} \\ \perp & : \quad \textit{sonst} \end{cases} . \tag{5.1}$$

Gemeinsam mit einer Wahrscheinlichkeitsverteilung über den möglichen Merkmalsausprägungen, geschätzt aus einem Alignment bekannter TFBS, entspricht ein Merkmal

[1] Es wurde gesagt, dass jede Merkmalsklasse die freien Parameter le und ri besitzt. Der Parameter pos der Klasse $\mathcal{M}_{PWM}$ steht abkürzend für eine Wahl von $\mathsf{le} = \mathsf{ri}$.

[2] z.B. markiert durch den IUPAC-Platzhalter `N`

$M \in \mathcal{M}_{PWM}$ einer Spalte eines PWM-Modells. Ein naiver BN-Klassifikator (NB), der Zufallsvariablen entsprechender Merkmale für $\mathsf{pos} = 1, \ldots, W$ enthält, ist demnach identisch mit einer PWM der Länge W. Durch die Möglichkeit Bayesscher Netze, Abhängigkeiten zwischen den Zufallsvariablen zu berücksichtigen, kann ein TAN-Modell mit den gleichen Merkmalen schon ein Fortschritt gegenüber einem PWM bedeuten (siehe [Bar03]).

5.1.2 $\mathcal{M}_{STRUCT}$: Sequenzabhängige, lokale DNA-Strukturparameter

Gemäß den Erläuterungen in Abschnitt 2.1.1 auf Seite 8 und in Abschnitt 2.2.2 auf Seite 18 und den dort zitierten Arbeiten muss angenommen werden, dass die hohe Sequenzähnlichkeit unter den TFBS zum Teil nur ein Resultat der Schaffung eines geeigneten strukturellen Profils für die Bindung eines TF ist. Für die Entwicklung trennscharfer TFBS-Modelle besteht die Hoffnung, durch direkte Modellierung der strukturellen Eigenschaften der TFBS bessere Merkmale zu finden als die mutmaßlich sekundären Sequenzmerkmale.

Die Hoffnung wird dadurch getragen, dass eine möglicherweise wichtige strukturelle Eigenschaft[3] durch verschiedene Nukleotidfolgen gewährleistet werden kann. Gewichtsmatrizen bzw. Merkmale der Klasse $\mathcal{M}_{PWM}$ würden im Extremfall aufgrund der großen Varianzen richtige TFBS kaum aus dem Hintergrund lösen können, während die allen TFBS gemeinsame strukturelle Eigenschaft weitaus weniger variiert. Abbildung 5.2 dient dazu, einen Eindruck von der Variabilität eines strukturellen Parameters zu erhalten. Die Merkmale der hier vorgestellten Klasse $\mathcal{M}_{STRUCT}$ messen jeweils den Mittelwert eines bestimmten sequenzabhängigen strukturellen Parameters in einer bestimmten Teilsequenz relativ zur Referenzposition $i_{\boldsymbol{\omega}}$, der anschließend mit dem Diskretisierungsverfahren von Fayyad und Irani (siehe Seite 96) diskretisiert wird [4].

Sowohl der strukturelle Parameter struct selbst, die Diskretisierungsintervallgrenzen als auch die Grenzen le und ri jener Teilsequenz sind deshalb freie Parameter der Merkmalsklasse. Bevor $\mathcal{M}_{STRUCT}$-Merkmale formal definiert werden, soll dies zuvor für die strukturellen Parameter geschehen.

DEFINITION 5.3: *Ein* sequenzabhängiger DNA-Strukturparameter $\rho^{(Z)}$ *mit* $1 \leq Z \leq 38$ *ist eine Abbildung*

$$\rho^{(Z)} = \begin{cases} \Sigma_{DNA} \times \Sigma_{DNA} & \rightarrow & \mathbb{R} \\ (x,\, y) & \mapsto & \rho^{(Z)}(x,\, y) \end{cases} \tag{5.2}$$

von der Menge von Dinukleotiden in die Menge der reellen Zahlen.

[3] z.B. die DNA-Biegsamkeit für TATA-Boxen

[4] Zwar ist der Wertebereich der dieser Teilsequenz-Mittelwerte nicht kontinuierlich, jedoch ist die Anzahl möglicher Zahlenwerte, die als Linearkombination der jeweils zehn Dinukleotidwerte berechnet werden können, zu hoch für eine sinnvolle Modellierung als diskrete Zufallsvariable in Bayesschen Netzen.

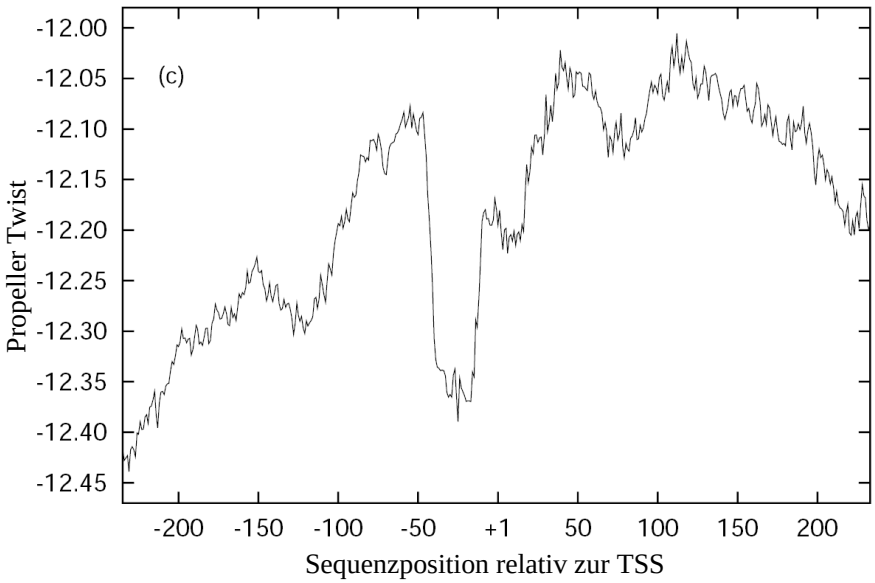


Abbildung 5.2: Durchschnittlicher *propeller twist* von 624 menschlichen Promotoren an verschiedenen Positionen relativ zur TSS. Besonders im Bereich der TATA-Box (ca. -35bp oberhalb der TSS) zeigt sich ein deutlich anderes Niveau als in der Umgebung. Die Kurve ist gleitend gemittelt. Quelle: [Ped99]

Wie aus der Definition hervorgeht, werden in dieser Arbeit 38 verschiedene strukturelle Parameter unterschieden, allesamt aus der Datenbank B-DNA-VIDEO von Ponomarenko et al. [Pon99] (siehe Tabelle 5.1). Diese Abbildungen sind nicht umkehrbar, da es aus Symmetriegründen nur 10 verschiedene Werte gibt (vergleiche die Fußnote auf Seite 10).

DEFINITION 5.4: *Sei $s_1 \ldots s_L$ mit $s_i \in \Sigma_{DNA}$, $1 \leq i \leq L$ eine DNA-Sequenz. Ein $\mathcal{M}_{STRUCT}$-*Merkmal *M mit den Parameter*

- $\mathsf{le}, \mathsf{ri} \in \mathbb{Z}$ *mit* $\mathsf{le} < \mathsf{ri}$*, welche die Teilsequenz von $s_1 \cdots s_L$ ausgehend von einer Referenzposition $i_{\boldsymbol{\omega}}$ festlegen, die zur Berechnung der Merkmalsausprägungen von M verwendet wird,*
- $\mathsf{struct} := \rho^{(Z)}$ *mit* $1 \leq Z \leq 38$*, der sequenzabhängige DNA-Strukturparameter,*
- $\boldsymbol{t}_M = (t_{1,M}, \ldots, t_{D,M})$ *mit* $t_{1,M} < \ldots < t_{D,M}$*, die Intervallgrenzen eines Diskretisierers*

besitzt die Wertemenge $D_M = \{0, 1, \ldots, D^{(M)}\}$ und hat die Abbildungsvorschrift

$$M(i_{\boldsymbol{\omega}}) = \mathrm{discr}_{\boldsymbol{t}_M} \left(\frac{1}{\mathsf{ri} - \mathsf{le}} \sum_{j=i_{\boldsymbol{\omega}}+\mathsf{le}}^{i_{\boldsymbol{\omega}}+\mathsf{ri}-1} \mathsf{struct}(s_j s_{j+1}) \right). \tag{5.3}$$

		Basenpaar									
DNA-Stadium	**Parameter**	AA/TT	AC/GT	AG/CT	AT	CA/TG	CC/GG	CG	GA/TC	GC	TA
B-DNA	helical slide [Å]	−0.03	−0.13	0.47	−0.37	1.46	0.6	0.63	−0.07	0.29	0.74
B-DNA	Δ freie Energie [Kcal/mol]	−1.2	−1.5	−1.5	−0.9	−1.7	−2.1	−2.8	−1.5	−2.3	−0.9
free DNA	helical tilt [°]	0.5	0.1	2.8	0.0	−0.7	2.7	0.0	0.9	0.0	0.0
B-DNA	Δ Entropie [cal/mol/K]	−21.9	−25.5	−16.4	−15.2	−21.0	−28.4	−29.0	−23.5	−26.4	−18.4
B-DNA	Drehsinn [°]	−154.0	143.0	2.0	0.0	−64.0	−57.0	0.0	120.0	180.0	0.0
B-DNA	helical twist [°]	38.9	31.12	32.15	33.81	41.41	34.96	32.91	41.31	38.5	33.28
B-DNA	helical wedge [°]	7.2	1.1	8.4	2.6	3.5	2.1	6.7	5.3	5.0	0.9
B-DNA	helical tip [°]	1.76	2.0	0.9	1.87	−1.64	0.71	0.22	1.35	2.5	6.7
B-DNA	Neigungswinkel [°]	−1.43	−0.11	−0.92	0.0	1.31	−1.11	0.0	−0.33	0.0	0.0
B-DNA	Persistenzlänge [nm]	35.0	60.0	60.0	20.0	60.0	130.0	85.0	60.0	85.0	20.0
B-DNA	Wahrsch. für Kontakt mit Nukleosomkern [%]	18.4	10.2	14.5	7.2	15.7	10.2	1.1	11.3	5.2	6.2
DNA-Protein-Komplex	helical slide [Å]	0.1	−0.6	−0.3	−0.7	0.4	−0.1	0.7	0.1	−0.3	0.1
B-DNA	Abweichung kleine Furche [Å]	2.94	4.22	2.79	4.2	3.09	2.8	3.21	2.95	4.24	2.97
B-DNA	helical twist 3 [°]	35.8	35.8	30.5	33.4	36.9	34.3	31.1	39.3	38.3	40.0
B-DNA	helical rise [Å]	3.16	3.41	3.63	3.89	3.23	4.08	3.6	3.47	3.81	3.21
freie DNA	helical slide [Å]	−0.1	−0.2	0.4	−0.4	1.6	0.8	0.7	0.0	0.4	0.9
B-DNA	helical twist 2 [°]	35.62	34.4	27.7	31.5	34.5	33.67	29.8	36.9	40.0	36.0
DNA-Protein-Komplex	helical tilt [°]	1.9	0.3	1.3	0.0	0.3	1.0	0.0	1.7	0.0	0.0
B-DNA	helical bend [°]	3.07	2.97	2.31	2.6	3.58	2.16	2.81	2.51	3.06	6.74
B-DNA	helical tilt [°]	−0.4	−0.9	−2.6	0.0	0.6	−1.1	0.0	−0.4	0.0	0.0
DNA-Protein-Komplex	helical twist [°]	35.6	31.1	31.9	29.3	35.9	33.3	34.9	35.9	34.6	39.5
B-DNA	Potential zur Bindung über große Furche [μ]	1.18	1.06	1.06	1.12	1.06	0.99	1.02	1.08	0.98	1.07
DNA-Protein-Komplex	helical roll [°]	0.8	−0.2	5.6	0.0	6.4	3.3	6.5	2.4	−2.0	2.7
freie DNA	helical roll [°]	0.3	0.5	4.5	−0.8	0.5	6.0	3.0	−1.3	−6.2	2.8
freie DNA	helical twist [°]	35.3	32.6	31.2	31.2	39.2	33.3	36.6	40.3	37.3	40.5
B-DNA	propeller twist [°]	−17.3	−6.7	−14.3	−16.9	−8.6	−12.8	−11.2	−15.1	−11.7	−11.1
B-DNA	Breite der großen Furche [Å]	12.15	12.37	13.51	12.87	13.58	15.49	14.42	13.93	14.55	12.32
B-DNA	Potential zur Bindung über kleine Furche [μ]	1.04	1.1	1.09	1.02	1.16	1.27	1.25	1.12	1.17	1.05
B-DNA	Tiefe der großen Furche [Å]	9.12	9.41	8.96	8.96	8.67	8.45	8.81	8.76	8.67	9.6
B-DNA	Δ Enthalpie [Kcal/mol]	−8.0	−9.4	−6.6	−5.6	−8.2	−10.9	−11.8	−8.8	−10.5	−6.6
B-DNA	helical roll [°]	0.5	0.4	2.9	−0.6	1.1	6.5	6.6	−0.1	−7.0	2.6
B-DNA	Tiefe der großen Furche [Å]	9.03	8.79	8.98	8.91	9.09	8.99	9.06	9.11	8.98	9.0
B-DNA	Abweichung kleine Furche [Å]	3.38	3.03	3.36	3.02	3.79	3.38	3.77	3.4	3.04	3.81
B-DNA	Größe der großen Furche [Å]	3.98	3.98	4.7	4.7	3.98	3.98	4.7	3.26	3.26	3.26
B-DNA	Schmelztemperatur [°]C	54.5	97.73	58.42	57.02	54.71	85.97	72.55	86.44	136.12	36.73
B-DNA	Breite der großen Furche [Å]	5.3	6.04	5.19	5.31	4.79	4.62	5.16	4.71	4.74	6.4
B-DNA	Clash-Stärke [Å]	0.64	0.95	2.53	1.68	0.8	1.78	2.42	0.03	0.22	0.0

Tabelle 5.1: Sequenzabhängige strukturelle Parameter definiert auf der Menge der Dinukleotide.

Eine berechtigte Frage bei der Verwendung sequenzabhängiger DNA-Strukturparameter ist, ob dabei nicht lediglich die Sequenzinformation anders kodiert verwendet wird, da die strukturellen Parameter einfache Abbildungen aus der Menge der Dinukleotide sind.

Aus mehreren Gründen ist dieser Einwand zurück zuweisen. Zum Einen werden die Funktionswerte der einzelnen Dinukleotide meist über einen größeren Intervall gemittelt, und dieser Mittelwert anschließend diskretisiert. Es leuchtet ein, dass hohe Mittelwerte durch höchst unterschiedliche Sequenzen erreicht werden können. Sollte ein hoher Wert eines Parameters ein guter Prädiktor einer TFBS sein, dann ist dies nicht zwingend durch eine hohe Sequenzähnlichkeit an dieser Stelle verursacht. Des Weiteren liegt schon auf Ebene einzelner Dinukleotide keine Bijektion zwischen den Dinukleotiden und den Parameterwerten vor, da es jeweils nur 10 verschiedene Werte für einen Parametern, jedoch 16 Dinukleotide gibt. Zu guter Letzt konnte in [Bal98] gezeigt werden, dass die strukturellen Parameter untereinander nicht stark korrelieren. Die strukturellen Parameter sind demnach nicht gegenseitig ersetzbar.

5.1.3 $\mathcal{M}_{CON}$: Treffer kurzer Consensusmotive

Der Aufbau von PWM-Modellen impliziert, dass sie nur dann eine trennscharfe TFBS-Repräsentation gewährleisten können, wenn sichergestellt ist, dass die gesuchten TFBS ein starkes *lückenloses* Alignment besitzen, also auch die exakt gleiche Länge haben. Gerade für Transkriptionsfaktoren, welche die DNA mit mehreren, unabhängigen Bindungsdomänen binden, gilt dies nicht immer. Häufig bestehen die TFBS dieser Faktoren aus mehreren kurzen, jedoch hoch-konservierten Teilsequenzen, die durch eine unterschiedlich kurze [5], nicht konservierte Zwischensequenzen voneinander getrennt sind.

Die Merkmalsklasse $\mathcal{M}_{CON}$ nimmt sich dieser Verschiebung an, indem Merkmale dieses Typs die Startposition des *ersten* Treffers einer kurzen Consensussequenz innerhalb eines Suchfensters angeben. Das Suchfenster wird wie schon bei $\mathcal{M}_{STRUCT}$-Merkmalen durch die Parameter le und ri relativ zur jeweiligen Referenzposition $i_{\boldsymbol{\omega}}$ des Musters definiert. Das in diesem Intervall zu suchende Consensusmotiv $\mathsf{A} = A_1 \cdots A_W \in \Sigma_{IUPAC} \cup \Sigma_{DNA}$, ist ein weiterer Parameter dieser Merkmalsklasse (siehe auch Kapitel 3 aufSeite 34)

Da der erste Treffer dieser Consensussequenz innerhalb des durch le und ri Intervalls gesucht wird, und dessen Position als Merkmalsausprägung verwendet wird, hängt die Größe des Wertebereichs D_M eines Merkmals $M \in \mathcal{M}_{CON}$ von le und ri und von der Länge W der Consensussequenz ab. Selbstverständlich wird dabei $W \leq \mathsf{ri} - \mathsf{le} + 1$ gefordert, so dass gilt: $D_M = \{j | j = 1 \ldots \mathsf{ri} - \mathsf{le} - W + 2\} \cup \{0\}$. Der Wert 0 ist für den Fall reserviert, dass es in dem Sequenzintervall keinen Treffer für die Consensussequenz A gibt.

Auch für diese Merkmalsklasse wird auf das Undefiniertheitssymbol $\perp$ verzichtet, indem an jenen Positionen $i_{\boldsymbol{\omega}}$, für welche das Sequenzintervall $[i_{\boldsymbol{\omega}}+\mathsf{le}, i_{\boldsymbol{\omega}}+\mathsf{ri}]$ die Sequenzgrenzen

[5] gemeint sind hier nur wenige Nukleotide, beispielsweise 0, 1 oder 2

überlappt, entweder der entsprechend kürzere Intervall verwendet wird (bei gleich bleibenden Wertebereich) oder in dem Fall, dass dieser verkürzte Intervall kleiner ist als die Consensussequenz, einfach der Wert 0 vergeben wird. Demnach kann für jede Position einer Eingabesequenz $s_1 \ldots s_L$ ein Wert aus obiger Wertemenge vergeben werden.

Ein zusätzlicher Merkmalsparameter, dir, ermöglicht die Wahl einer von drei Suchmodi: $\mathsf{dir} = forward$ für die Suche des ersten Treffers von links nach rechts, $\mathsf{dir} = backward$ für den ersten Treffer bei der rückwärtigen Suche von rechts nach links und schließlich $\mathsf{dir} = exist$. In diesem letzten Fall interessiert nur die Existenz eines Treffers, nicht seine Position. Die beiden möglichen Zustände werden durch die beiden Werte 0 =*kein Treffer* und 1 =*Treffer* kodiert.

DEFINITION 5.5: *Sei $s_1 \ldots s_L$ mit $s_i \in \Sigma_{DNA}$, $1 \leq i \leq L$ eine DNA-Sequenz. Ein Merkmal M der Klasse $\mathcal{M}_{CON}$ besitzt die freien Parameter*

- $\mathsf{A} = A_1 \ldots A_W$ *mit* $A_i \in \Sigma_{IUPAC} \cup \Sigma_{DNA}$*, die Consensus-Sequenz, für die Treffer gesucht werden,*
- $\mathsf{le}, \mathsf{ri} \in \mathbb{Z}$ *mit* $\mathsf{le} < \mathsf{ri}$*, welche die Teilsequenz von $s_1 \cdots s_L$ ausgehend von einer Referenzposition $i_{\boldsymbol{\omega}}$ festlegt, in denen Treffer für* A *gesucht werden,*
- $\mathsf{dir} \in \{forward, backward, exist\}$*,*

die Wertemenge $D_M = \{j | j = 1 \ldots \mathsf{ri} - \mathsf{le} - W + 2\} \cup \{0\}$*, falls* $\mathsf{dir} \in \{forward, backward\}$ *oder* $D_M = \{0, 1\}$*, falls* $\mathsf{dir} = exist$ *und folgende Abbildungsvorschrift:*

- $\mathsf{dir} = forward$*:*

$$M(i_{\boldsymbol{\omega}}) = \max\left\{\min\left\{j \in D_M \setminus \{0\} : s_{i_{\boldsymbol{\omega}}+j+\mathsf{le}} \ldots s_{i_{\boldsymbol{\omega}}+j+\mathsf{le}+W-1} \in \mathcal{L}(\mathsf{A})\right\}, 0\right\}$$

- $\mathsf{dir} = backward$*:*

$$M(i_{\boldsymbol{\omega}}) = \max\left\{\max\left\{j \in D_M \setminus \{0\} : s_{i_{\boldsymbol{\omega}}+j+\mathsf{le}} \ldots s_{i_{\boldsymbol{\omega}}+j+\mathsf{le}+W-1} \in \mathcal{L}(\mathsf{A})\right\}, 0\right\}$$

- $\mathsf{dir} = exist$*:*

$$M(i_{\boldsymbol{\omega}}) = \begin{cases} 1 & : \exists j \in D_M \setminus \{0\} : s_{i_{\boldsymbol{\omega}}+j+\mathsf{le}} \ldots s_{i_{\boldsymbol{\omega}}+j+\mathsf{le}+W-1} \in \mathcal{L}(\mathsf{A}) \\ 0 & : \text{sonst} \end{cases}$$

Abbildung 5.3 illustriert die Verwendung der $\mathcal{M}_{CON}$-Merkmale. Das eingangs erwähnte Problem des variablen Abstands zweier ausdrucksstarker Motivteile kann z.B. mit einem Merkmal dieser Klasse in Verbindung mit weiteren PWM-Merkmalen gelöst werden. Dazu dient ein Merkmal $M \in \mathcal{M}_{CON}$, dessen Consensussequenz beispielsweise einen prägnanten Teil des zweiten Motivteils beschreibt, als Indikator, wie weit dieser vom ersten Teil entfernt ist. Die PWM-Merkmale für den zweiten Motivteil stehen sinnvollerweise in Abhängigkeit dieses Merkmals. Auf diese Weise wird sichergestellt,

dass die PWM-Merkmale trotz Verschiebung relativ zur Referenzposition stets die gleiche Position des zweiten Motivteils abbilden. Diese Konstruktion beschreibt jedoch nur die Intention, die hinter der Merkmalsklasse $\mathcal{M}_{CON}$ steht. Welche *Verdrahtung* eines Bayesschen Netzes, das diese Merkmale enthält, schließlich gewählt wird, obliegt den Lernalgorithmen.

Des Weiteren sei vermerkt, dass allein durch die Möglichkeit, Abhängigkeiten zwischen den $\mathcal{M}_{PWM}$-Merkmalen der beiden Motivteile zu modellieren, eine Möglichkeit gegeben wäre, das Problem der Verschiebung des zweiten Teils zu beheben. Bei geschickter Dimensionierung der $\mathcal{M}_{CON}$-Merkmale kann jedoch dem gegenüber eine zuverlässigere Bestimmung des Abstandes beider Motivteile bei einer geringeren Anzahl zu schätzender Modellparameter erfolgen. Die potentielle Parameterreduktion ist im Übrigen die zweite Motivation für die Merkmalsklasse $\mathcal{M}_{CON}$, vor allem im Suchmodus dir $= exist$. So kann ein nahezu obligatorischer Teil des charakteristischen TFBS-Sequenzmotivs durch ein einziges Merkmal dargestellt werden, das lediglich zwei mögliche Werte besitzt.

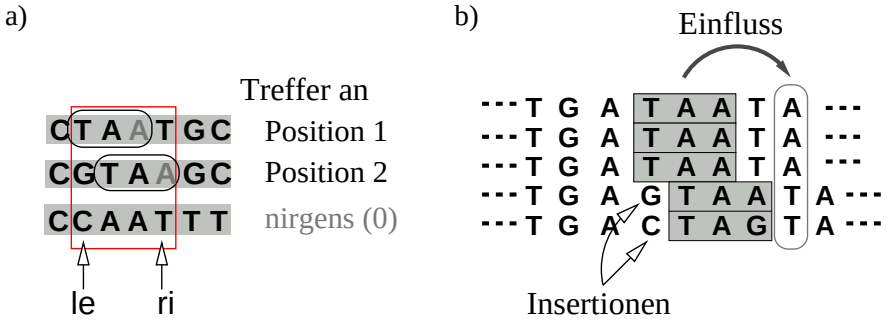


Abbildung 5.3: Funktionsweise des $\mathcal{M}_{CON}$-Merkmals für die Consensussequenz TAR und den Parametern le $= 2$, ri $= 6$ und dir $= forward$. Daraus ergibt sich ein Wertebereich $D_M = \{1, 2\} \cup \{0\}$. a.) zeigt drei Sequenzmuster, wobei $i_{\omega} = 1$ ist, der rote Rahmen hebt die Teilsequenz hervor, in der nach Treffern für TAR gesucht werden. In jedem der drei Muster hat das Merkmal eine andere Ausprägung. b.) Anwendung des Merkmales. Zu sehen ist ein Alignment von TFBS, wobei die unteren beiden ein eingefügtes Nukleotid besitzen, dass die Konserviertheit nachfolgender Spalten verhindert. Das Merkmal misst die Verschiebung. Ein $\mathcal{M}_{PWM}$-Merkmal für Spalte 8 könnte von M abhängen, um den Einfluss zu modellieren.

5.1.4 $\mathcal{M}_{PRF}$: Nukleotidverteilungen von Teilsequenzen

Beim Betrachten eines in seine Sequenzumgebung eingebetteten Alignments von TFBS eines Faktors zeigen sich vor allem im TFBS-Kern klare, spaltenweise Übereinstimmungen. Auch in den benachbarten Bereichen gibt es häufig auffällige Häufungen bestimmter Nukleotide, die jedoch nicht an festgelegten Spalten auftreten, wodurch sie zur Verbesserung eines PWM-Modells nicht beitragen können. So treten TFBS des Transkripti-

onsfaktors Sp1 vornehmlich in G-reichen Sequenzen auf. Ein möglicher Grund für die Überrepräsentation einiger Nukleotide in TFBS-Umgebungen könnte sein, dass Sequenzen, die reich an diesen Nukleotiden sind, bestimmte strukturelle Eigenschaften besitzen, die eine Bindung des Transkriptionsfaktors erleichtern. Ein anderer Grund könnte sein, dass durch Anreicherung von den Nukleotiden in der TFBS-Nachbarschaft, die im Kern der TFBS selbst häufig vorkommen, die Wahrscheinlichkeit eines Bindungsereignisses erhöht ist, da die Nachbarschaft selbst als schwache TFBS des Faktors dienen kann. Dieses Erklärungsmodell könnte beispielsweise für Sp1 gelten, deren Kern-TFBS ebenfalls sehr G-reich ist.

Die Merkmale der Klasse $\mathcal{M}_{PRF}$ messen den Anteil einer Teilmenge NUCS von Σ_{DNA} in einem durch le und ri definierten Sequenzintervall. Ähnlich wie bei $\mathcal{M}_{STRUCT}$ muss der quasi-kontinuierliche Wertebereich diskretisiert werden. Hierfür wird ebenfalls das Diskretisierungsverfahren von Fayyad und Irani eingesetzt.

DEFINITION 5.6: *Sei $s_1 \ldots s_L$ mit $s_i \in \Sigma_{DNA}$, $1 \leq i \leq L$ eine DNA-Sequenz. Ein* $\mathcal{M}_{PRF}$-Merkmal M *besitzt die freien Parameter*

- $\mathsf{NUCS} \subset \Sigma_{DNA}$, *die Nukleotide, deren Anteil bestimmt werden soll*
- $\mathsf{le}, \mathsf{ri} \in \mathbb{Z}$ *mit* $\mathsf{le} < \mathsf{ri}$, *welche die Teilsequenz von $s_1 \cdots s_L$ ausgehend von einer Referenzposition $i_{\boldsymbol{\omega}}$ festlegt, in welcher der Anteil der Nukleotidmenge bestimmt wird,*
- $\boldsymbol{t}_M = (t_{1,M}, \ldots, t_{D,M})$ *mit $t_{1,M} < \ldots < t_{D,M}$, die Intervallgrenzen eines Diskretisierers,*

den Wertebereich $D_M = \{0, 1, \ldots, D^{(M)}\}$ und die Abbildungsvorschrift

$$M(i) = \mathrm{discr}_{\boldsymbol{t}_M} \left(\frac{\sum_{j=i_{\boldsymbol{\omega}}+\mathsf{le}}^{i_{\boldsymbol{\omega}}+\mathsf{le}+\mathsf{ri}-1} |\{s_j\} \cap \mathsf{NUCS}|}{\mathsf{ri} - \mathsf{le} + 1} \right) \tag{5.4}$$

5.1.5 $\mathcal{M}_{CPG}$: CpG-Inseln

In Kapitel 2 auf Seite 21 wurde eine statistische Auffälligkeit regulativer DNA-Sequenzen beschrieben, die CpG-Inseln. So treten CG-Dinukleotide im Mittel $10 - 20$ mal häufiger in Promotorsequenzen auf als im restlichen Genom.

CpG-Inseln sind sicher keine lokale Eigenschaft einer TFBS. Gerade beim Durchsuchen von ganzen Genomabschnitten anstelle von eingegrenzten Promotoren könnte die starke Überschneidung von Promotor- und CpG-Inselbereichen dazu dienen, die für die Detektion von TFBS interessanteren Promotorsequenzen probabilistisch hervorzuheben. Aus diesem Grund werden $\mathcal{M}_{CPG}$-Merkmale eingeführt, die für ein Muster $i_{\boldsymbol{\omega}}$ untersuchen, ob dessen Umgebung Teil einer CpG-Insel ist. Die Umgebung wird wieder durch die Intervallgrenzen le und ri festgelegt. Als Kriterium für das Vorhandensein einer CpG-Inseln gilt die Definition von Gardiner-Garden und Frommer [GG87b].

DEFINITION 5.7: *Sei $s_1 \ldots s_L$ mit $s_i \in \Sigma_{DNA}$, $1 \leq i \leq L$ eine DNA-Sequenz. Ein $\mathcal{M}_{CPG}$-Merkmal M besitzt die freien Parameter* $\mathsf{le}, \mathsf{ri} \in \mathbb{Z}$ *mit* $\mathsf{le} < \mathsf{ri}$*, den Wertebereich* $D_M = \{true, false\}$ *und die Abbildungsvorschrift*

$$M(i_{\boldsymbol{\omega}}) = \begin{cases} true & : \; p_{\mathsf{G}}(\boldsymbol{s}_{i_\omega}) + p_{\mathsf{C}}(\boldsymbol{s}_{i_\omega}) > 0.5 \wedge p_{\mathsf{CG}}(\boldsymbol{s}_{i_\omega}) \geq 0.6 p_{\mathsf{G}}(\boldsymbol{s}_{i_\omega}) p_{\mathsf{C}}(\boldsymbol{s}_{i_\omega}) \\ false & : \; \text{sonst} \end{cases}, \quad (5.5)$$

wobei $\boldsymbol{s}_{i_\omega}$ die Teilsequenz $s_{i_\omega+\mathsf{le}} \ldots s_{i_\omega+\mathsf{ri}-1}$ bezeichnet.

In der Praxis konnten $\mathcal{M}_{CPG}$-Merkmale jedoch nicht gewinnbringend in einem TFBS-Modell eingesetzt werden. Ein Grund hierfür könnte in der Negativ-Stichprobe liegen, die beim Lernen der TFBS-Modelle verwendet wurde. Diese setzt sich ebenso wie die Positivstichproben aus Promotorenabschnitten zusammen, so dass ein statistisch auffälliger Trend hin zu CpG-Inseln in der Positiv- jedoch zu weniger CpG-Inseln in der Negativstichprobe nicht einstellt. Der Lernalgorithmus kann in diesem Fall keinen diskriminierenden Nutzen von $\mathcal{M}_{CPG}$-Merkmalen erkennen.

5.1.6 $\mathcal{M}_{KOOP}$: Benachbarte TFBS kooperierender Faktoren

In Abschnitt 2.2.2 auf Seite 19 wurde dargelegt, dass Transkriptionsfaktoren häufig nicht alleine ihre regulierende Wirkung entfalten, sondern in Kooperation mit weiteren Transkriptionsfaktoren. Bei der Klassifikation eines Musters $i_{\boldsymbol{\omega}}$ sollte deshalb auch untersucht werden, ob die Umgebung von Position $i_{\boldsymbol{\omega}}$ die Bindung dieser kooperierenden Faktoren ermöglicht, d.h. Bindungssequenzen für diese vorhanden sind.

$\mathcal{M}_{KOOP}$-Merkmale suchen in der Umgebung einer Sequenzposition nach möglichen Bindungsstellen eines bekannten kooperierenden Faktors. Wie bei den meisten anderen Merkmalsklassen wird der zu durchsuchende Sequenzintervall durch die Parameter le und ri festgelegt. Die Suche nach benachbarten TFBS erfolgt mit Hilfe eines gewöhnlichen PWM-Modells PWM. Als Treffer gelten Positionen j, für welche die PWM-Bewertung eine Schranke $\mathsf{score} \in \mathbb{R}$ überschreitet. Die Wertemenge von $\mathcal{M}_{KOOP}$-Merkmalen ist schlicht $D_M = \{true, false\}$.

DEFINITION 5.8: *Sei $s_1 \ldots s_L$ mit $s_i \in \Sigma_{DNA}$, $1 \leq i \leq L$ eine DNA-Sequenz. Ein* $\mathcal{M}_{KOOP}$-Merkmal *M besitzt die freien Parameter*

- PWM*, eine Gewichtsmatrix der Länge W,*
- score*, die Bewertungsschranke für die Suche von Treffern mit* PWM*,*
- $\mathsf{le}, \mathsf{ri} \in \mathbb{Z}$ *mit* $\mathsf{le} < \mathsf{ri}$*, welche die Teilsequenz von $s_1 \cdots s_L$ ausgehend von einer Referenzposition $i_{\boldsymbol{\omega}}$ festlegt, in welcher Treffer für* PWM *gesucht werden,*

die Wertemenge $D_M = \{true, false\}$ *und die Zuordnungsvorschrift*

$$M(i_{\boldsymbol{\omega}}) = \begin{cases} true & : \exists j \in \{i_{\boldsymbol{\omega}} + \mathsf{le}, \ldots, i_{\boldsymbol{\omega}} + \mathsf{ri} - 1\} : S_{\mathsf{PWM}}(s_j \ldots s_{j+W-1}) > \mathsf{score} \\ false & : \text{sonst} \end{cases} \tag{5.6}$$

Die Ausnutzung von Häufungstendenzen von TFBS stellt eine wichtige Möglichkeit dar, die Klassifikationsfehlerraten bei der Vorhersage von TFBS zu reduzieren. Mit den TFBS-Modul-Erkennungssystemen in Abschnitt 5.5 und in Kapitel 6 werden jedoch integrative Systeme vorgestellt, die gezielt Häufungen von TFBS suchen, und dafür TFBS-BN für Einzel-TFBS-Vorhersagen nutzen. Die zusätzliche Anwendung von $\mathcal{M}_{KOOP}$-Merkmalen erscheint dann nicht mehr sinnvoll. Auch konnten in verschiedenen Testreihen keine Erfolge mit ihrer Verwendung erzielt werden. Gründe dafür könnten beispielsweise die etwas strikte Entscheidungsregel bzw. Wertemenge dieser Merkmale sein. Ein anderer Grund ist sicherlich, dass die Lernstichproben, die zum Lernen der TFBS-BN verwendet werden, selten eine klare Tendenz zeigen, einen bestimmte Partner-TFBS in der Nachbarschaft zu enthalten. Aus diesen Gründen werden diese Merkmale in diesem Kapitel nicht weiter betrachtet.

5.2 Modellierung von Merkmalsmengen in Bayesschen Netzen

Nachdem die Merkmalsklassen vorgestellt wurden, aus denen potentielle Merkmalsmengen zur Beschreibung von TFBS rekrutiert werden können, wird in diesem Abschnitt beschrieben, wie solche Merkmale zu TFBS-Vorhersage-Modellen, den *TFBS-BN*, zusammengesetzt werden und wie diese Modelle anschließend eingesetzt werden.

Unterabschnitt 5.2.1 beschreibt die Suche von TFBS in einer (langen) DNA-Sequenz. Anschließend werden in Unterabschnitt 5.2.2 Details zum Lernvorgang für die TFBS-BN beschrieben.

5.2.1 Anwendung von TFBS-BN-Modellen

In diesem Abschnitt soll beschrieben werden, wie die auf Bayesschen Netzen basierenden TFBS-Modelle eingesetzt werden, um TFBS-Vorhersagen in einer beliebigen DNA-Sequenz zu treffen. Sei also $\boldsymbol{s} = s_1 \ldots s_L$ mit $s_i \in \Sigma_{DNA}$ für $1 \leq i \leq L$ eine solche Sequenz und $\mathcal{C}$ ein TFBS-BN mit d Merkmalen $M_1, \ldots, M_d$.

Die Verarbeitung der Sequenz $\boldsymbol{s}$ durch das Modell $\mathcal{C}$ ist in Abbildung 5.4 dargestellt. In einem ersten Schritt wird für jede Sequenzposition i mit $1 \leq i \leq L$, den Mustern in dem behandelten Problemkreis, durch Anwendung der den Merkmalen innewohnenden Abbildungsvorschrift ein Merkmalsvektor $\boldsymbol{m}^{(i)} = (m_1^{(i)}, \ldots, m_d^{(i)})$ erzeugt.

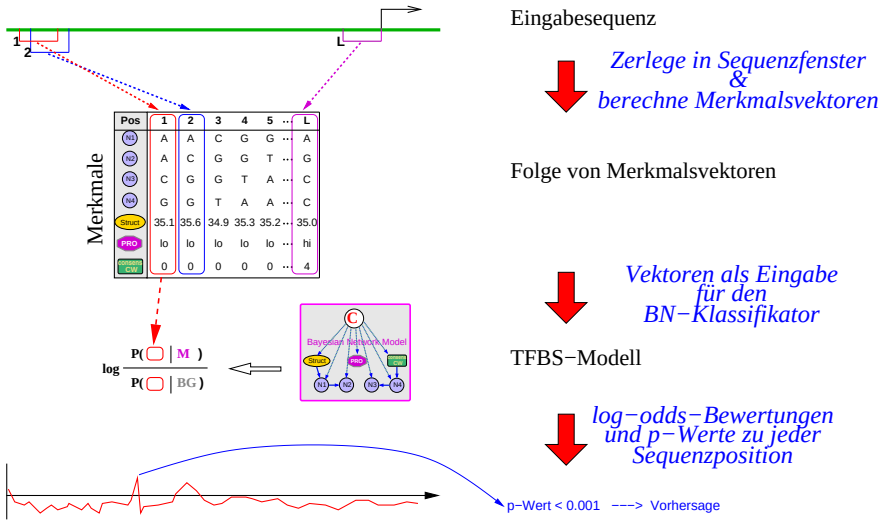


Abbildung 5.4: Ablauf des Durchsuchens einer DNA-Sequenz mit einem TFBS-Modell.

Es sei an dieser Stelle darauf hingewiesen, dass insbesondere für die Randmuster einer DNA-Sequenz nicht alle Merkmale eines Vektors definiert sind. Obwohl Bayessche Netze gerade im Falle unvollständiger Daten anderen stochastischen Modellierungsansätzen überlegen sind, werden diese wenigen unvollständigen Merkmalsvektoren an den beiden Enden der DNA-Sequenz nicht weiter betrachtet.

Das TFBS-Modell $\mathcal{C}$ ist ein BN-Klassifikator für die Klassen Ω_{TFBS} und $\Omega_{\neg\mathsf{TFBS}}$. Die Anwendung erfolgt, wie in Abschnitt 4.3 auf Seite 73 beschrieben, über Anfragen an die Klassenvariable C für einen konkreten Merkmalsvektor $\boldsymbol{m}^{(i)} = (m_1^{(i)}, \ldots, m_d^{(i)})$:

$$p_{TFBS}(\boldsymbol{m}^{(i)}) = P_{\mathcal{C}}(C = TFBS \,|\, \boldsymbol{M} = \boldsymbol{m}^{(i)}) \tag{5.7}$$

$$p_{\neg TFBS}(\boldsymbol{m}^{(i)}) = P_{\mathcal{C}}(C = \neg TFBS \,|\, \boldsymbol{M} = \boldsymbol{m}^{(i)}) \tag{5.8}$$

Die in der Definition auf Seite 73 vereinbarte Entscheidungsregel

$$\delta(\boldsymbol{m}) = \operatorname*{argmax}_{\kappa'} P(C = \kappa' \,|\, \boldsymbol{m})$$

ist für den vorliegenden Anwendungsfall wenig geeignet. Für jede Position i mit

$$p_{TFBS}(\boldsymbol{m}^{(i)}) > \frac{1}{2}$$

würde eine TFBS-Vorhersage getroffen werden. Selbst bei wirklich gut trennenden Klassifikatoren würde diese Entscheidungsregel um Größenordnungen mehr TFBS vorhersagen als in einer genomischen Sequenz zu erwarten wäre.

Die Schranke für eine TFBS-Vorhersage muss also höher gelegt werden, um nur die wirklich passenden Merkmalsvektoren in die Klasse Ω_{TFBS} zu klassifizieren. Zur Ableitung

einer adäquaten Entscheidungsregel wird analog zu PWM-Modellen verfahren. Die beiden Wahrscheinlichkeiten $p_{TFBS}(\boldsymbol{m}^{(i)})$ und $p_{\neg TFBS}(\boldsymbol{m}^{(i)})$ werden verwendet, um eine *Bewertung* $S_{\mathcal{C}}(\boldsymbol{m}^{(i)})$ zu berechnen:

$$S_{\mathcal{C}}(\boldsymbol{m}^{(i)}) = \log_2 \frac{p_{TFBS}(\boldsymbol{m}^{(i)})}{p_{\neg TFBS}(\boldsymbol{m}^{(i)})}. \tag{5.9}$$

Diese *log-odds*-Bewertung korreliert mit der Wahrscheinlichkeit dafür, dass $\boldsymbol{m}^{(i)}$ den charakteristischen Merkmalsvektoren der Klasse Ω_{TFBS} zugerechnet werden kann. Die hier angewendete Entscheidungsregel wird über eine Schranke S_{TFBS} für die *log-odds*-Bewertungen definiert.

$$\delta(\boldsymbol{m}^{(i)}) = \begin{cases} TFBS & : \quad S_{\mathcal{C}}(\boldsymbol{m}^{(i)}) > S_{TFBS} \\ \neg TFBS & : \quad \text{sonst} \end{cases} \tag{5.10}$$

Ein Grenzwert $S_{TFBS} = 0$ entspräche der originalen Entscheidungsregel aus der Definition auf Seite 73.

Entscheidend für die Suche in der DNA-Sequenz ist, dass ein Merkmalsvektor $\boldsymbol{m}^{(i)}$ mit der Sequenzposition i korrespondiert, und deshalb eine TFBS-Vorhersage, also die Klassifikation von $\boldsymbol{m}^{(i)}$ in die Klasse Ω_{TFBS}, ebenfalls auf die ursprüngliche DNA-Sequenzposition bezogen werden kann.

Signifikanzbestimmung von Treffern. Im Prinzip könnte nun eine beliebige Schranke S_{TFBS} festgelegt werden, die hinsichtlich der Anzahl von Vorhersagen *vertrauenerweckend* wirkt. Um der Entscheidung für eine Schranke mehr Gewicht zu verleihen, wird die statistische Signifikanz einer solchen Schranke bzw. von *log-odds*-Bewertungen $S_{\mathcal{C}}(\boldsymbol{m}^{(i)})$ berechnet.

Die statistische Signifikanz von $S_{\mathcal{C}}(\boldsymbol{m}^{(i)})$ wird durch den p-Wert angegeben, der Wahrscheinlichkeit dafür, mit einem zufälligen[6] Merkmalsvektor eine gleiche oder höhere Bewertung zu erhalten. Dazu ist es selbstverständlich nötig, die Verteilung der Bewertungen zufälliger Merkmalsvektoren zu kennen. Aufgrund der komplexen Zusammensetzung des Merkmalsraums und der Abhängigkeit dieser Verteilung von der Verteilung der zugrunde liegenden zufälligen DNA-Sequenzen ist es nicht denkbar, die Verteilung exakt zu bestimmen. Ein Ausweg wäre die Verwendung der empirischen Verteilung, die sich bei der Anwendung auf eine große Zahl zufälliger Vektoren ergibt. Dieses Vorgehen ist problematisch, denn zur Beurteilung einer *log-odds*-Bewertung zum einem Signifikanzniveau von 5% (nicht besonders streng) interessiert vor allem der Bereich hoher Bewertungen, die unter zufälligen Vektoren selten auftreten. Die empirische Verteilung ist in diesem Bereich also sehr dünn besetzt. Dieses Problem könnte nur durch eine unrealistisch große Anzahl von Bewertungen gelindert werden.

Mit einer kleineren Anzahl zufälliger Bewertungen kann eine parametrische Wahrscheinlichkeitsannahme getroffen werden. Die Parameter werden anhand einer Stichprobe von

[6] Präziser: mit einem Merkmalsvektor, der von einer zufälligen DNA-Sequenz erzeugt wurde.

Bewertungen zufälliger Merkmalsvektoren gelernt. In [Kar90] wurde gezeigt, dass *log-odds*-Bewertungen von lokalen Alignment-Verfahren gemäß einer *Fischer-Tippett-Verteilung* verteilt sind. Da auch Treffer von PWM-Modellen als Bewertung einer Art lokalen Alignments aufgefasst werden können, fand sie auch für die Signifikanzbestimmung von PWM-Treffern Anwendung. Obwohl diese Argumentation auf TFBS-BN nicht übertragen werden kann, zeigte sich, dass auch die Verteilung der *log-odds*-Bewertungen von TFBS-BN erstaunlich gut durch eine Fischer-Tippett-Verteilung approximiert wird (siehe Abbildung 5.5). Zu den *Extremwertverteilungen* gehörig modelliert sie das Maximum

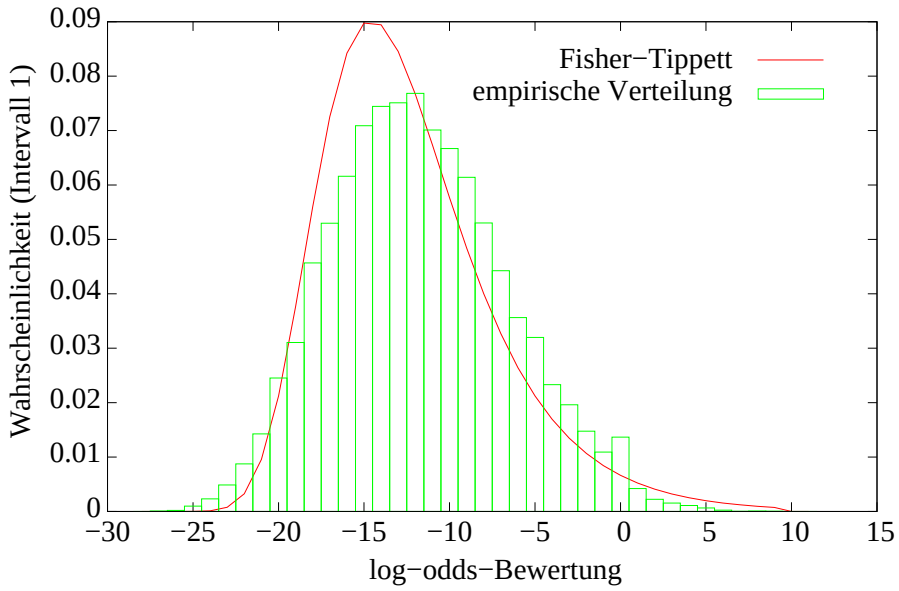


Abbildung 5.5: Anpassung einer Fischer-Tippett-Verteilung an die *log-odds*-Bewertungen eines TFBS-BN.

einer großen Anzahl identisch verteilter Zufallsvariablen. Mit den beiden Parametern $a, b \in \mathbb{R}$ und $b > 0$ ist ihre Dichte definiert durch

$$f(x) = \frac{1}{b} e^{-\frac{x-a}{b}} e^{-e^{-\frac{x-a}{b}}} \tag{5.11}$$

Entscheidend für die Berechnung eines p-Wertes ist die Verteilungsfunktion $F(x)$, welche die einfache Form

$$F(x) = e^{-e^{-\frac{x-a}{b}}} \tag{5.12}$$

besitzt. Der p-Wert einer *log-odds*-Bewertung $S_{\mathcal{C}}(\boldsymbol{m}^{(i)})$ ist dann einfach $1-F(S_{\mathcal{C}}(\boldsymbol{m}^{(i)}))$.

5.2.2 Überwachtes Lernen

TFBS-BN werden unter Verwendung einer etikettierten Stichprobe d, die sowohl TFBS, als auch Negativsequenzen enthält, gelernt. Da das diesbezügliche Vorgehen sich nicht

wesentlich von den in Abschnitt 4.4 beschriebenen Lernschritten für Bayessche Netz-Klassifikatoren unterscheidet, wird hier ein Hauptaugenmerk auf die feinen Unterschiede gelegt, die angewendet wurden, um den Lernvorgang auf das spezielle Klassifikationsproblem abzustimmen. Prinzipiell besteht der Lernprozess aus vier Schritten:

1. Aufstellen einer *Kandidatenmenge* von D Merkmalen, die zur Merkmalsauswahl zugelassen sind. Im Zuge der Generierung werden auch alle Diskretisierer dieser Merkmale mit Hilfe des Entropie-basierten Diskretisierungsverfahrens aus Unterabschnitt 4.4.4 konstruiert.
2. *Merkmalsauswahl* mittels des SFFS-Algorithmus.
3. Lernen der *TAN-Struktur*: hierfür wird die Methode *construct-TAN* angewendet, jedoch mit der im Folgenden erklärten Änderung.
4. Lernen der *Wahrscheinlichkeitsparameter*: Für alle Wahrscheinlichkeiten wird der konstante Dirichlet-Parameter $\alpha = 1$ verwendet.

Lernen der Struktur. Die Methode *construct-TAN* von Friedman et al [Fri97] erzeugt die optimale Struktur für eine gegebene Stichprobe, indem ein minimaler Spannbaum konstruiert wird. Gemäß des auf Seite 87 angegebenen Algorithmus ist dieser Graph zwingend zusammenhängend. Um diese Eigenschaft zu erreichen, müssen unter Umständen sehr schwach gewichtete Kanten berücksichtigt werden, obwohl zwischen den durch sie verbundenen Variablen nur eine unbedeutend schwache Abhängigkeit besteht. Dies kann die Leistung einer TFBS-BN erheblich verhindern.

Aus diesem Grund wird die Methode *construct-TAN* dahingehend abgewandelt, das nach der Konstruktion des minimalen Spannbaums jene Kanten wieder entfernt werden, deren Abhängigkeit nicht die erforderliche Stärke aufweist. Eine Parameteroptimierung ergab, dass ein MIC-Wert von -0.17 ein günstiger Wert ist. Kanten, deren Gewicht größer als -0.17 ist, werden also wieder entfernt.

5.3 Ergebnisse der Evaluierung

In diesem Abschnitt wird der zuvor vorgestellte Modellierungsansatz für TFBS hinsichtlich seiner Klassifikationsleistung evaluiert. Insgesamt werden TFBS-Modelle für 86 Transkriptionsfaktoren gelernt, die sich mit den bisher bevorzugt verwendeten PWM-Modellen messen müssen. In Unterabschnitt 5.3.1 wird näher auf die verwendeten Lern- und Testdaten eingegangen. Unterabschnitt 5.3.2 gibt Auskunft über den Leistungsvergleich zwischen PWM-Modellen und den hier vorgestellten, auf Bayesschen Netzen basierenden TFBS-Modellen.

5.3.1 Datensammlung

Die in Unterabschnitt 3.1.4 auf Seite 48 vorgestellte Datenbank TRANSFAC enthält eine umfangreiche Sammlung von ca. 820 PWM-Modellen[7]. Für viele Transkriptionsfaktoren liegen dabei mehrere PWM-Modelle vor, die jeweils auf unterschiedlichen Teilmengen aller verfügbaren TFBS trainiert wurden[8]. Gleichfalls sind die meisten PWM-Modelle in TRANSFAC für eine Reihe nah verwandter Transkriptionsfaktoren gültig. Die hier verwendeten Datensätze werden daher mit den von TRANSFAC vergebenen Datenbank-Schlüsseln der PWM-Modelle identifiziert anstatt mit einem konkreten Transkriptionsfaktor. Tabelle 5.2 enthält eine Zuordnung von TRANSFAC-Schlüsseln zu Transkriptionsfaktoren für die hier verwendeten Datensätze. Für einen beträchtlichen Teil der PWM-Modelle in TRANSFAC ist zusätzlich das lückenlose Alignment der TFBS angegeben, die zum Erlernen der PWM-Gewichte verwendet wurde. Zur Konstruktion eines Bayesschen Netzes für TFBS wird ebenfalls mindestens ein solches Alignment benötigt, so dass sich die hier angestellten Vergleiche auf PWM-Modelle beschränkten, für die diese Lerndaten in TRANSFAC bekannt gegeben waren. Zusätzlich wurde die Evaluierung auf Modelle beschränkt, für die das zugrunde liegende Alignment mindestens aus 10 Sequenzen besteht, da bei noch kleineren Datensätzen zum Einen ein robuster Leistungsvergleich nicht gewährleistet werden kann und zum Anderen die Merkmalsauswahl zu starker Überanpassung tendieren würde. Zudem muss die Qualität dieser PWM-Modelle aufgrund eben dieser Überanpassung stark angezweifelt werden [Rah04].

Eine weitere Einschränkung der Evaluierung betrifft den experimentellen Ursprung der TFBS in den Alignments. Da der hier erläuterte Ansatz auf der Arbeitsthese gründet, adäquatere Modelle dadurch erhalten zu können, dass weitere biologische Eigenschaften der genomischen Positionen berücksichtigt werden, wurden hier nur solche Datensätze berücksichtigt, die ausschließlich im Genom auftretende TFBS enthalten, nicht etwa SELEX-Sequenzen (siehe Seite 28). Erfahrungsgemäß ist ein Alignment aus genomischen TFBS wesentlich heterogener als ein SELEX-Alignment, so dass diese Alignments potentiell Problemfälle des PWM-Ansatzes darstellen. Demgegenüber ist zu erwarten, dass ein SELEX-Alignment schon sehr gut durch eine PWM modelliert wird, und keine Notwendigkeit für die Berücksichtigung weiterer Merkmale besteht.

Nach allen Einschränkungen blieben 86 PWM-Modelle übrig, deren Lerndaten nun zur Konstruktion entsprechender TFBS-BN-Modelle verwendet werden. Abbildung 5.6 zeigt ein Histogramm der Datensatzgrößen. Bei allen Datensätzen handelt es sich um TFBS aus Säugetiergenomen, in der Mehrheit sind es menschliche TFBS. Vereinzelt, wenn dies durch einen hohen Konservierungsgrad der TFBS und ihres Faktors gerechtfertigt ist, enthalten die Alignments auch TFBS von Vögeln und anderen Wirbeltieren. Für jede TFBS eines solchen Alignments gibt es einen Verweis auf einen Eintrag in die TFBS-Relation von TRANSFAC. Für genomische TFBS gibt es in der TRANSFAC TFBS-Relation einen Verweis auf die entsprechende Sequenz in EMBL einschließlich der Posi-

[7] Version 10.2

[8] z.B. alle TFBS einer bestimmten Veröffentlichung oder alle TFBS außer SELEX-Sequenzen

Datensatz	Transkriptionsfaktoren
M00466	HIF-1, HIF-1α, HIF-1α-isoform1
M00621	C/EBPδ
M00638	HNF-4, HNF-4α, HNF-4α1, HNF-4α2
M00650	MTF-1
M00690	AP-3, AP-3 (2)
M00695	ETF
M00733	Smad4
M00744	Pit-1, Pit-1A, Pit-1B
M00761	δNp63α, p53, p53-isoform-1, p63α, p63γ, p73α, p73β
M00762	COUP, COUP-TF1, COUP-TF2, HNF-4, HNF-4α, HNF-4γ, PPAR-α:RXR-α, PPAR-γ2:RXR-α
M00763	PPAR-α, PPAR-α:RXR-α, PPAR-β, PPAR-γ, PPAR-γ1, PPAR-γ2, PPAR-γ2:RXR-α, PPAR-γ:RXR-α
M00764	HNF-4, HNF-4α, HNF-4α1, HNF-4α2, HNF-4α3, HNF-4α4, HNF-4α7, HNF-4γ
M00771	ERF, Elf-1, Elk-1, Erg-1, Ets-1 δVII, Fli-1, NERF, NERF-1a, NERF-2, SAP-1a, Tel-2b, c-Ets-1, c-Ets-2
M00773	c-Myb, c-Myb-isoform1
M00774	NF-TNF, NF-κB, NF-κB(-like), NF-κB2 (p49), RelA-p65, p100, p105, p50, p52
M00775	CBF(2), CBF-A, CBF-B, CBF-C, CP1, NF-Y, NF-Y', NF-YA, NF-YA isoform-1, NF-YA-L, NF-YB, NF-YC, NF-YC-3
M00789	GATA-1, GATA-2, GATA-3, GATA-4, GATA-5, GATA-6
M00790	HNF-1α, HNF-1α-A, HNF-1α-B, HNF-1α-C, HNF-1β, HNF-1β-A, HNF-1β-B, HNF-1β-C
M00791	HNF-3, HNF-3α, HNF-3β, HNF-3γ
M00792	Smad1, Smad1.1, Smad2, Smad2-L, Smad3, Smad3:Smad4, Smad4
M00793	YY1
M00794	Nkx2-1, Nkx2-1-isoform1
M00795	Brn1, OCA-B, Oct-1, Oct-10, Oct-2, Oct-9, Oct-R, POU2F1, POU2F1a, POU2F1b, POU2F1c, POU4F1, POU5F1
M00796	USF, USF-1, USF1, USF1:USF2, USF1a, USF1b, USF2, USF2a, USF2b
M00799	Max, Max1, c-Myc
M00800	AP-2, AP-2α, AP-2αA, AP-2αB, AP-2β, AP-2γ
M00801	ATF, ATF-4, ATF2, ATF3, CREB, CREBβ, CREMα, CREMβ, CREMγ, CREMτ, CREMτ1, CREMτ2, δCREB
M00802	Pit-1, Pit-1-xbb3, Pit-1A, Pit-1B
M00803	DP-1, E2F, E2F+E4, E2F-1, E2F-3a, E2F-4
M00805	LEF-1, LEF-1S, TCF-1, TCF-1(P), TCF-3
M00806	CTF, CTF-1, CTF-2, NF-1, NF-1/L, NF-1A
M00808	Pax-1, Pax-2, Pax-2a, Pax-3, Pax-4a, Pax-4c, Pax-5, Pax-6, Pax-8, Pax6-1
M00809	FOXD3, FOXF1, FOXF2, FOXH1, FOXJ1, FOXJ1a, FOXJ1b
M00810	SRF, SRF-I, SRF-L, SRF-M, SRF-S
M00912	C/EBPα, C/EBPβ, C/EBPδ, C/EBPepsilon, C/EBPγ
M00913	B-Myb, c-Myb, c-Myb-isoform1
M00915	AP-2, AP-2α, AP-2αA, AP-2αB, AP-2β, AP-2γ
M00916	CREB, CREBβ, CREMα, CREMβ, CREMγ, CREMtau, CREMtau1, CREMtau2, CREMtauα, δCREB
M00917	CREB, CREBβ, CREMα, CREMβ, CREMγ, CREMtau, CREMtau1, CREMtau2, CREMtauα, δCREB
M00918	DP-1, E2F, E2F+E4, E2F-1, E2F-3a, E2F-4
M00919	DP-1, E2F, E2F+E4, E2F-1, E2F-3a, E2F-4
M00920	DP-1, E2F, E2F+E4, E2F-1, E2F-1:DP-1, E2F-3a, E2F-4, E2F-7
M00921	GR, GR-α, GR-β
M00922	SRF, SRF-I, SRF-L, SRF-M, SRF-S
M00924	AP-1, FosB, Fra-1, Fra-2, JunB, JunD, c-Fos, c-Jun
M00925	AP-1, FosB, Fra-1, Fra-2, JunB, JunD, c-Fos, c-Jun, c-Jun:c-Fos, δFosB
M00926	AP-1, FosB, Fra-1, Fra-2, JunB, JunD, c-Fos, c-Jun
M00928	E2
M00929	E12, E47, MRF4, MyoD, myogenin
M00930	Oct-1, POU2F1, POU2F1a
M00931	Sp1, Sp1 isoform 1, Sp3, Sp4
M00932	Sp1, Sp2, Sp3, Sp4
M00933	Sp1, Sp1 isoform 1, Sp2, Sp3, Sp4
M00934	Zeste
M00935	NF-AT1, NF-AT1C, NF-AT2, NF-AT3, NF-AT4
M00939	DP-1, E2F, E2F+E4, E2F-1, E2F-3a, E2F-4
M00940	E2F-1
M00941	MEF-2A, MEF-2C/δ8, MEF-2DAB, aMEF-2
M00959	ER-α, ER-α-L, ER-α:ER-β, ER-β, ER-β1
M00960	GR, GR-α, GR-β, PR, PR A, PR B, PR-α, PR-β
M00961	VDR
M00962	AR
M00963	RAR-α, RAR-β, RAR-γ, RXR-α, RXR-β, RXR-γ, T3R-α, T3R-α, T3R-β
M00964	CAR, FXR, FXR:RXR-α, LXR-α:RXR-α, LXR-β:RXR-α, PXR-1, PXR-1A, PXR-2, PXR:RXR-α
M00965	CAR, COUP, COUP-TF1, COUP-TF2, LXR-α, LXR-β, PXR-1, PXR-2, RAR-α, RAR-β, RAR-γ, RXR-α, RXR-β
M00966	CAR, PXR-1, PXR-2, RXR-α, RXR-β, SXR:RXR-α, VDR
M00967	COUP, COUP-TF1, COUP-TF2, HNF-4, HNF-4α, HNF-4α1, HNF-4α2, HNF-4α3, HNF-4α4, HNF-4α7, HNF-4γ
M00971	Elf-1, Elk-1, Erg-1, Erg-2, Fli-1, GABP-α, GABP-β, NERF, Net, PEA3, PU.1
M00972	IRF-1, IRF-10, IRF-2, IRF-3, IRF-4, IRF-5, IRF-7, IRF-7A, IRF-7H, IRF-8, IRF-9, ISGF-3
M00973	E12, E47, EMF1, HTF4, INSAF, ITF, MASH-1, MASH-2, MEF1, MRF4, Myf-5, Myf-6, MyoD, SEF2-1B, myf-5
M00974	Smad1, Smad10, Smad2, Smad3, Smad4, Smad5, Smad6, Smad7, Smad8
M00976	ARNT2, AhR, AhR2, AhR:Arnt, AhRR, AhRR:Arnt, Arnt, HIF-1, HIF-1α, HIF-1α-isoform1
M00978	LEF-1, LEF-1S, TCF-1, TCF-1(P), TCF-1A, TCF-1B, TCF-1C, TCF-1E, TCF-1F, TCF-1G
M00979	Pax-6, Pax-6 (Pax-QNR), Pax-6 / Pd-5a, Pax6-1
M00980	TBP, TFIID
M00981	ATF-1, ATF-2, ATF-4, ATF3, ATF4, ATF5, ATF6, CRE-BP1, CRE-BP2, CREB, CREM
M00982	Egr-1, Egr-2, Egr-3, Egr-4
M00983	Bach1, Bach2, LCR-F1, MAF, NF-E2, NF-E2 p45, Nrf1, Nrf2, Nrf3, c-Maf, v-Maf
M01010	HMG I, HMG-Y, HMGI-C
M01023	HSF1, HSF1-L, HSF1-S, HSF1long, HSF1short
M01034	DEC1, E12, E47, EMF1-4, MEF1, MRF4, Mad1, Mad3, Mad4, Myf-5, Myf-6, MyoD
M01035	YY1, factor δ
M01036	COUP, COUP-TF1, COUP-TF2
M01067	Gfi1, Gfi1b
M01130	PBF
M01135	GAMYB

Tabelle 5.2: Zuordnung von Datensatzbezeichnern zu Transkriptionsfaktoren.

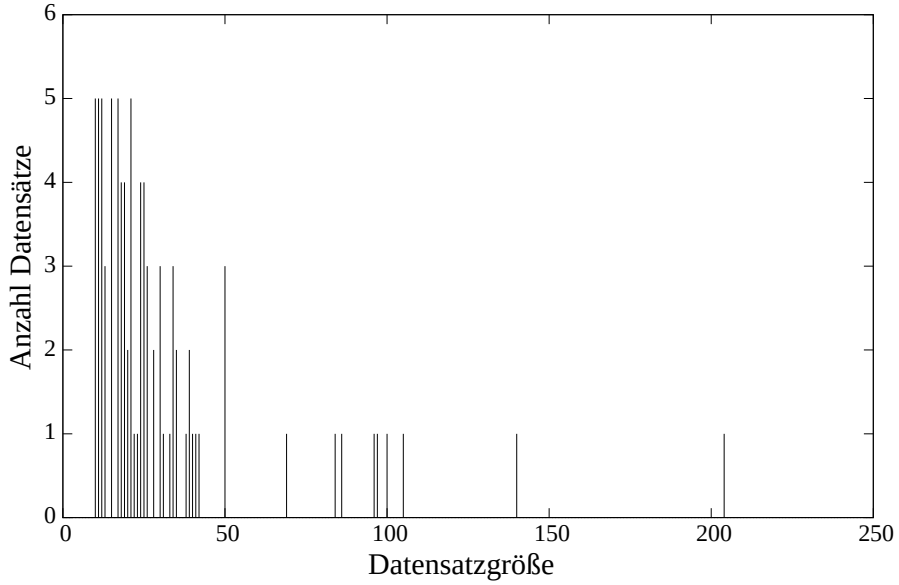


Abbildung 5.6: Histogramm der TRANSFAC-Datensatzgrößen, angegeben als Anzahl der TFBS, die zum Lernen des originalen PWM-Modells verwendet wurde.

tion in der EMBL-Sequenz. Um möglichst viele Freiheiten für die Berechnung von Merkmalen aus den TFBS-Mustern berechnen zu können, wurden diese EMBL-Sequenzen heruntergeladen und die TFBS aus TRANSFAC in ihnen annotiert. Ein Sequenzdatensatz besteht demnach nicht ausschließlich aus dem Alignment der TFBS. Vielmehr sind die Bindungsstellen des Alignments in ihre genomische Umgebung eingebettet.

Als Referenzposition $i_{\boldsymbol{\omega}}$, mit der ja gemäß den obigen Ausführungen ein TFBS-Muster identifiziert wird, und bezüglich der alle Merkmalsausprägungen berechnet werden, wird auf die erste Position der jeweiligen TFBS im TRANSFAC-Alignment gesetzt.

Die TFBS-Modelle sind Bayessche Netz-Klassifikatoren für das Zwei-Klassen-Problem Ω_{TFBS} vs. $\Omega_{\neg\mathsf{TFBS}}$. Zum Lernen dieser Klassifikatoren werden zusätzlich Lernbeispiele für die Hintergrundklasse $\Omega_{\neg\mathsf{TFBS}}$ benötigt[9]. Die $\Omega_{\neg\mathsf{TFBS}}$-Sequenzen müssen möglichst sinnvoll für den späteren Einsatz der Modelle gewählt werden. Eine schlechte Wahl wären kodierende Sequenzen, da diese sich statistisch stark von Promotorsequenzen unterscheiden, und ein zu optimistisches Bild über die Leistung der Klassifikatoren abgäben. Als *Negativdaten*, d.h. Lernbeispiele aus $\Omega_{\neg\mathsf{TFBS}}$, wurden deshalb alle 1871 Promotorsequenzen aus der Datenbank EPD (siehe Unterabschnitt 3.1.4) gewonnen, die zu Wirbeltiergenomen gehören. Für jede dieser Sequenzen wurde zufällig eine Position als Referenzposition $i_{\boldsymbol{\omega}}$ festgehalten.

[9]Für die Berechnung der Gewichte von PWM-Modellen werden Negativdaten nur in Form einer bestimmten Basenverteilung verwendet, die für die regulativen Sequenzen des untersuchten Genoms zutreffend ist.

5.3.2 Versuchsdurchführung

Vorauswahl günstiger Merkmale. Die in Abschnitt 5.1 eingeführten Merkmalsklassen bieten so viele Freiheitsgrade, dass eine nahezu unbegrenzte Anzahl von Merkmalen berechnet werden könnte. Damit wäre das SFFS-Verfahren, dass eine möglichst diskriminierende Teilmenge von Merkmalen auswählen soll, überfordert, denn in jeder Ebene des Suchraumes aller Merkmale müsste das Einfügen jedes dieser unzähligen Merkmale überprüft werden.

Die große Mehrheit aller möglichen Merkmale kann jedoch schon durch eine Analyse des Lerndatensatzes ausgeschlossen werden. Die Vorauswahl halbwegs viel versprechender Merkmale wurde anhand folgender Kriterien für die einzelnen Merkmalsklassen durchgeführt:

- $\mathcal{M}_{PWM}$**-Merkmale:** ausgehend von Referenzposition $i_{\boldsymbol{\omega}}$ wurden alle $\mathcal{M}_{PWM}$-Merkmale im Sequenzintervall $[i_{\boldsymbol{\omega}} - 10, i_{\boldsymbol{\omega}} + 30]$ zur Merkmalsauswahl zugelassen. Die Erfahrung zeigt, das die nötige Sequenzähnlichkeit bereits wenige Basenpaare vom PWM-Alignment verschwindend gering ist, so dass nicht befürchtet werden musste, eine hoch konservierte Spalte außerhalb des verwendeten Bereichs zu übersehen.
- $\mathcal{M}_{STRUCT}$**-Merkmale:** hier galt es, sinnvolle Merkmale für alle 38 sequenzabhängigen DNA-Strukturparametern und für verschiedene Sequenzintervalle zu bestimmen. Untersuchte Sequenzintervalle, angegeben als $(\mathsf{le}, \mathsf{ri})$-Paare waren:
  $(0, W)$, $(-10, 0)$, $(W, W + 10)$, $(-10, W + 10)$ sowie alle einzelnen Dinukleotide von $i_{\boldsymbol{\omega}} - 10$ bis $i_{\boldsymbol{\omega}} + W + 10$, wobei W ist die Länge des TRANSFAC-Alignments bezeichnet. Als Mittelwerte eines strukturellen Parameters müssen die Merkmalsausprägungen diskretisiert werden. Für die meisten der genannten Merkmale fand das Entropie-basierte Diskretisierungsverfahren (siehe Seite 97) keine Intervallunterteilung, die hinsichtlich der Klassifikation von Ω_{TFBS} vs. $\Omega_{\neg\mathsf{TFBS}}$ ein Gewinn wäre. Folgerichtig wurden nur jene Merkmale in der Kandidatenmenge belassen, für die das Diskretisierungsverfahren mindestens 2 Intervalle vorschlug. Außerdem wurden Merkmale ausgeschlossen, deren Diskretisierung mehr als 5 Intervalle ausgab, denn eine so große Anzahl tritt häufig bei sehr kleinen Datensätzen (hinsichtlich der Anzahl von Ω_{TFBS}-Beispielen) auf und lässt sich auf eine Überanpassung des Diskretisierers an die Lerndaten zurückführen.
- $\mathcal{M}_{PRF}$**-Merkmale:** die Auswahl von Kandidatenmerkmalen lief analog zu der Auswahl von $\mathcal{M}_{STRUCT}$-Merkmalen.
- $\mathcal{M}_{CON}$**-Merkmale:** mögliche Merkmale dieser Klasse wurden positionsweise ermittelt. Für jede Position der TFBS-Muster in einem Intervall von $[-5, 15]$ wurden all jene Sequenzen der Längen $1, 2, 3$ ermittelt, die an dieser Position in mindestens einem Muster auftraten. Für diese Sequenzen wurde ein entsprechendes Merkmal für diese Position im Suchmodus $\mathsf{dir} = \mathit{exist}$ als Kandidat zugelassen. Analog wurden solche Merkmale auch für alle IUPAC-Verallgemeinerungen erzeugt und in die Kandidatenmenge eingefügt. Merkmale im Modus $\mathsf{dir} = \mathit{forward}$ oder

dir = *backward* wurden genau dann erzeugt, wenn eine Consensus-Sequenz in mehreren, aufeinander folgenden Positionen vor kam. Gab es z.B. die Sequenz `AR` an Position b und an Position $b+1$, so wurde ein Merkmal eingefügt, dass im Bereich $[b, b+3]$ nach dem ersten bzw. letzten Treffer von `AR` sucht.

Die beschriebenen Vorauswahlkriterien wurden als Abwägung der beiden Extreme entwickelt, einerseits durch zu wenige Freiheitsgrade der Merkmalsparameter eine zu kleine Anzahl von Kandidaten zu erhalten, wobei viele vielversprechende Merkmale nicht in der Merkmalsmenge vertreten wären, oder andererseits durch zu viele Freiheitsgrade eine zu große Merkmalsmenge zu erhalten, die zuviele offensichtlich unbrauchbare Merkmale enthält und verhindert, dass das Auswahlverfahren in annehmbarer Zeit in interessante Bereiche des Suchraumes vordringen kann. Durch Anwendung obiger Regeln wurden Kandidatenmerkmale der wichtigsten Merkmalsklassen definiert, die ein Minimum an Relevanz für den betreffenden Sequenzdatensatz haben. Die Anzahl D solcher Kandidatenmerkmale liegt für die 86 Sequenzdatensätze im hohen dreistelligen Bereich (siehe das Histogramm in Abbildung 5.7).

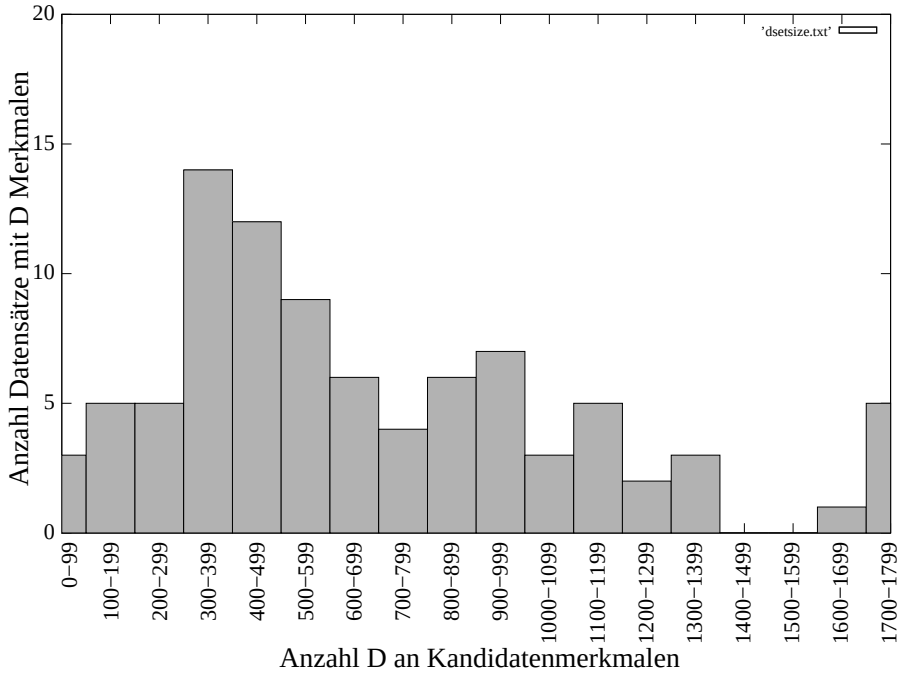


Abbildung 5.7: Histogramm der Anzahl D aller zur Merkmalsauswahl zugelassenen Merkmale in den 86 Datensätzen

Um einen Sequenzdatensatz in eine Lernstichprobe d gemäß der Definition auf Seite 76 umzuwandeln, wurden alle generierten Merkmale auf jedes Muster des Sequenzdatensatzes angewendet. Auf diese Weise enstanden also 86 etikettierte Stichproben.

Randomisierung der Stichproben. Die Merkmalsauswahl verwendete eine 10-fache Kreuzvalidierung zur Bestimmung der Güte von Merkmalsteilmengen. Es musste berücksichtigt werden, dass der Suchpfad von der Sortierung der Stichprobenelemente abhängen kann. Da auch der SFFS-Algorithmus nicht vollständig vor lokalen Maxima geschützt ist, können für verschiedene Permutationen der Stichprobenelemente unterschiedliche Merkmalsteilmengen generiert werden.

Um die Robustheit der Merkmalsauswahl zu untersuchen, wurden je fünf Permutationen einer jeden Stichprobe angefertigt. Modelle verschiedener Gestalt wurden jeweils auf allen fünf Permutationen trainiert. Für jede Stichprobe standen demnach anschließend fünf Modellpakete zur Verfügung, deren Modelle auf der gleichen Permutation trainiert wurden.

Die Modelle eines solchen Pakets wurden nun in mehreren Kreuzvalidierungen miteinander verglichen. Dazu wurde die jeweilige Stichprobenpermutation verwendet, um erneut zehn Permutationen herzustellen. Alle Modelle eines Modelpaketes wurden anhand jeder dieser zehn neuen Permutationen validiert. Aus diesem Grund waren die Einzelklassifikationen bzw. die daraus abgeleiteten Klassifikatorenkennzahlen paarweise miteinander vergleichbar[10]. Die mehrfache Durchführung der Kreuzvalidierung unter verschiedenen Permutationen der Stichproben diente dazu, den Effekt einer Überanpassung der Merkmalsteilmenge an jene Sortierung der Daten zu vermeiden, die für die Merkmalsauswahl verwendet wurde.

Untersuchte Modellvarianten. Zu jeder der fünf Stichprobenpermutationen wurden auf verschiedene Weise TFBS-BN-Modelle und ein PWM-Modell trainiert und anhand der davon abgeleiteten weiteren zehn Permutationen miteinander verglichen. Die TFBS-BN-Modellvarianten sind:

- **TAN-Modelle:** eine möglichst gute Merkmalsteilmenge, wobei Abhängigkeiten zwischen den Merkmalen berücksichtigt werden. Für die Merkmalsauswahl wurde der SFFS-Algorithmus eingesetzt (siehe Unterabschnitt 4.4.3 auf Seite 89). Die Ergebnismenge wurde in einem TAN-Modell modelliert. Hinsichtlich der Startmenge des SFFS-Algorithmus wurden hierbei unterschieden:
  - $TAN_{\emptyset}$, Start des SFFS auf der leeren Menge.
  - TAN_{PWM}, Start des SFFS mit der Menge jener $\mathcal{M}_{PWM}$-Merkmale, die in der originalen TRANSFAC-PWM vorkommen.
- **NB-Modelle:** eine möglichst gute Merkmalsteilmenge, wobei *keine* Abhängigkeiten modelliert werden. Diese Modelle wurden betrachtet, um den Leistungsbeitrag zu untersuchen, der sich allein aus der Zulassung von Abhängigkeiten ergibt. Analog zu den TAN-Modellen wurden unterschieden:
  - $NB_{\emptyset}$, Start des SFFS auf der leeren Menge.

[10] Die Klassifikation eines Musters geschah dadurch mit Modellen eines Pakets, die auf demselben Teil der Daten trainiert wurden

  - NB_{PWM}, SFFS mit den $\mathcal{M}_{PWM}$-Merkmale der TRANSFAC-PWM
- **PWM-Modell:** entspricht einem NB-Modell, dass nur die $\mathcal{M}_{PWM}$-Merkmale der TRANSFAC-PWM modelliert. In mathematischer Hinsicht ist es äquivalent zu einem PWM-Modell, dessen Gewichte Wahrscheinlichkeiten sind. In den Ergebnisübersichten werden diese Modelle einfach mit PWM bezeichnet.

Qualitätsmaße. Mittels der genannten Kreuzvalidierungen wurden verschiedene Qualitätsmaße für die Modelle eines Modellpaketes bestimmt, die aufgrund identischer Stichprobenpermutationen direkt miteinander vergleichbar waren.

Da es sich bei der Klassifikationsaufgabe weniger um ein klassisches 2-Klassen-Problem mit gleichberechtigten Klassen handelt, sondern um die treffsichere Erkennung weniger TFBS, die sich in einer großen Menge von Nicht-TFBS befinden, wurde bei der Wahl der Qualitätsmaße der Fokus auf die Klasse Ω_{TFBS} gelegt. Die Klassifikation entspricht deshalb einem statistischen Test, dessen Nullhypothese davon ausgeht, dass es sich bei einem betrachteten Merkmalsvektor *nicht* um eine TFBS handelt. Klassifikationen fanden für ein bestimmtes Modell $\mathcal{C}$ über die auf Seite 113 beschriebenen *log-odds*-Bewertungen $S_{\mathcal{C}}(\,\cdot\,)$ bzw. einer diesbezüglichen Schranke S_{TFBS} statt. Dabei wurde die Nullhypothese für einen Merkmalsvektor $\boldsymbol{m}^{(i)}$ verworfen, wenn für seine Bewertung $S_{\mathcal{C}}(\boldsymbol{m}^{(i)}) > S_{TFBS}$ gilt.

Für einen gegebenen Klassifikator und eine Schranke S_{TFBS} lassen sich zunächst die vier elementaren Statistiken bestimmen:

- **TP:** die Anzahl der TFBS, die als solche richtig erkannt wurden (für englisch *true positives*),
- **FP:** die Anzahl von $\Omega_{\neg\mathsf{TFBS}}$-Beispielen, die falsch als TFBS klassifiziert wurden (für englisch *false positives*). In der klassischen Statistik wird diese Kennzahl als α-Fehler bezeichnet.
- **TN:** die Anzahl der $\Omega_{\neg\mathsf{TFBS}}$-Beispiele, die korrekt nicht als TFBS erkannt wurden (für englisch *true negatives*),
- **FN:** die Anzahl der TFBS, die nicht erkannt wurden (für englisch *false negatives*). Dabei handelt es sich um den β-Fehler eines statistischen Tests.

Zusammengefasst ergeben diese Statistiken die *Kontingenztabelle* eines Klassifikators, angewendet auf eine Stichprobe. Zwei wichtige Kennzahlen der Leistung eines Klassifikators sind die TP-Rate

$$tp \;=\; \frac{TP}{TP+FN}, \tag{5.13}$$

die den Anteil an richtig erkannten TFBS an allen TFBS einer Stichprobe beschreibt und die FP-Rate

$$fp \;=\; \frac{FP}{TN+FP}, \tag{5.14}$$

die den Anteil aller $\Omega_{\neg\mathsf{TFBS}}$-Merkmalsvektoren einer Stichprobe, die als TFBS erkannt wurden, beschreibt. Beide Maße haben den Wertebereich [0, 1]. Es liegt auf der Hand, dass eines der Maße zu Lasten des anderen Maßes optimiert werden kann. Die Wahl einer Klassifikationsschranke S_{TFBS} bedeutet, einen Kompromiss zwischen einer möglichst hohen TP-Rate und einer gleichzeitig möglichst niedrigen FP-Rate zu schließen.

Ein visuelles Maß für die Güte eines Klassifikators ist die *ROC-Kurve* (für englisch: *receiver operator curve*), welche die TP-Rate tp in Abhängigkeit einer bestimmten FP-Rate fp darstellt. Diese Kurve gibt einen Gesamtüberblick über das Leistungsvermögen eines Klassifikators. Je weiter sich der Kurvenverlauf der Koordinate $(0, 1)$ annähert, desto besser der Klassifikator, denn schon bei niedrigen fp-Werten werden hohe tp-Raten erreicht. Verläuft die Kurve in der Nähe Diagonale des Koordinatensystems, handelt es sich um eine rein zufällige Klassifikation, für jedes richtig erkannte Muster wird ungefähr eines falsch erkannt. Zur Berechnung der ROC-Kurve wurde in dieser Arbeit wie folgt vorgegangen: Die *log-odds*-Bewertungen aller echten TFBS wurden sortiert. Nacheinander wurden diese Bewertungen als Klassifikationsschranke S_{TFBS} verwendet und die FP-Rate für diese festgelegte TP-Rate gemessen. Dabei wurden die FP-Raten der zehn Kreuzvalidierungen gemittelt, um eine einzige ROC-Kurve pro Modell zu erhalten.

Um die Klassifikationsleistung anhand der ROC-Kurve in einer Kennzahl auszudrücken, eignet sich der Flächeninhalt unter der ROC-Kurve, der AUC (für englisch *area under the ROC curve*), der ebenso einen Wertebereich [0, 1] hat, wobei 1 den perfekten Klassifikator bezeichnet. Zusätzlich wurden zum Vergleich der Modelle die FP-Raten bei einer in der Praxis relevanten fest eingestellten TP-Rate von 0.9 untersucht. Dieses Maß wird als FP_{90} bezeichnet.

Eine weitere interessante Frage ist die nach der Robustheit des SFFS-Algorithmus. Da jede Stichprobe für jedes Modell in fünf verschiedenen Permutationen zur Merkmalsauswahl verwendet wurd, und diese Merkmalssuchen mit je zwei verschiedenen Startmengen durchsucht werden, gibt es für jede Modellklasse (TAN und NB) zehn verschiedene Modelle. Es kann nun untersucht werden, wie viele der Merkmale in allen dieser zehn Modelle ausgewählt wurden. Im Idealfall sind alle Modelle identisch, im schlechtesten Fall sind die Merkmalsmengen der Modelle disjunkt. In letzterem Fall hat die Sortierung der Daten einen sehr großen Einfluss auf den Suchpfad, den der SFFS-Algorithmus eingeschlagen hat, und dieser landet unvorhersehbar in irgendeinem lokalen Maxima. Die Robustheit wurde deshalb durch den Quotienten

$$r = \frac{\#\text{Merkmale, die in jedem Modell vorkommen}}{\#\text{Merkmale, die in mind. einem Modell vorkommen}} \tag{5.15}$$

gemessen.

5.3.3 Ergebnisse

Klassifikationsraten. Die Mehrheit der ROC-Kurven von TAN- und NB-Modellen wiesen einen höheren AUC-Wert auf als die korrespondierenden PWM-Modelle. Verglichen

	$TAN_{\emptyset}$	TAN_{PWM}	$NB_{\emptyset}$	NB_{PWM}	PWM
Gewinner in [%] **Fällen**	73,67	65,82	72,15	72,91	–
⊘ **AUC**	0,99803	0,99643	0,99804	0,995536	0,99288
t**-Test AUC**	0,000244	$4,11e^{-8}$	$1,49e^{-8}$	0,00026	–
⊘ FP_{90}	0,0062	0,011	0,0059	0,014	0,025
t**-Test** FP_{90}	$1,02e^{-6}$	0,013	$2,74e^{-7}$	0,0009	–

Tabelle 5.3: Klassifikationsleistung der verschiedenen Modellvarianten im Vergleich zu PWM-Modellen. Die erste Zeile gibt an, in wie viel Prozent der Modellpakete ein bestimmtes Modell einen höheren AUC-Wert aufwies als das PWM-Modell. Die zweite Zeile gibt AUC-Mittelwerte, die dritte den p-Wert des t-Test-Mittelwertvergleiches mit dem PWM-Mittelwert an. Die vierte Zeile enthält die mittleren FP_{90}-Werte, die letzte die dazugehörigen t-Testergebnisse.

wurden, jeweils mit dem PWM-Modell, die vier Modellvarianten $TAN_{\emptyset}$, TAN_{PWM}, $NB_{\emptyset}$ sowie NB_{PWM} auf jedem der 5 Datensatzpermutationen aller 86 Datensätze. Es lagen also insgesamt 430 zu vergleichende Modellsätze vor. Die Mittelwerte der AUC-Werte für die vier TFBS-BN-Varianten wurden gegen den der PWM-Modelle mittels eines paarweisen t-Tests miteinander verglichen. Die Ergebnisse sind in Tabelle 5.3 aufgeführt.

Es zeigt sich, dass die AUC-Werte von TFBS-BN-Modellen im Mittel signifikant höher sind. Besonders stark ist der Unterschied in Fällen, in denen der SFFS-Algorithmus auf der leeren Menge gestartet wurde. Bezüglich des AUC-Wertes schnitten mehr als 73% der $TAN_{\emptyset}$-Modelle besser ab als das dazugehörige PWM-Modell, dicht gefolgt von den NB-Varianten, die in 72% der Fälle die PWM-Modelle übertrafen. Das zweite Maß zur Beurteilung der Klassifikationsleistung, das FP_{90} war in allen TFBS-BN-Varianten im Mittel signifikant niedriger (besser) als für PWM-Modelle. Auch hier ließ sich beobachten, dass $\emptyset$-Varianten ein besseres Ergebnis zeigten als die Modelle, bei denen der SFFS-Algorithmus mit den PWM-Spalten gestartet wurde.

In Abbildung 5.8a.) und 5.8b.) sind zwei ausgewählte ROC-Kurven abgebildet, in denen die Verbesserung besonders eindrucksvoll war. Abbildung 5.8c.) zeigt einen typischen ROC-Plot in Fällen, in denen das PWM-Modell besser abschnitt. Das Typische daran ist, dass die PWM-Modelle vor allem im Bereich niedriger TP-Raten besser abschneiden während die TFBS-BN-Modelle im Bereich hoher TP-Raten niedrigere Raten Falsch-Positiver aufweisen.

Trennung der Klassenbereiche. Die erhöhte Trennung der Klassenbereiche durch BN-Klassifikatoren dokumentieren die Verteilungen der *log-odds*-Bewertungen, die für ein Modell jeweils getrennt nach Klassen durch eine Gumbel-Verteilung approximiert wurden. Abbildung 5.9 zeigt die typischen Effekte anhand eines Vergleiches dieser Verteilun-

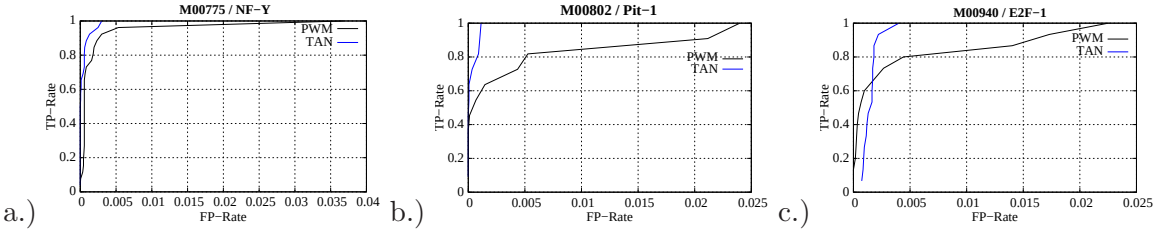


Abbildung 5.8: Exemplarische ROC-Kurven von TFBS-Modellen im Direktvergleich mit PWM-Modellen. a.) und b.) *Schöne* Beispiele: die ROC-Kurve des TFBS-BN-Modells liegt stets über der PWM-ROC-Kurve. c.) Ein *problematisches* Beispiel: Die ROC-Kurven schneiden im unteren Bereich sehr schlecht ab.

gen zwischen PWM und einem TAN-Modell. Sowohl die Mittelwerte der beiden Verteilungen werden im Vergleich zu dem PWM-Modell auseinander gezogen, als auch die Varianz beider Verteilungen verringert. Die Schnittfläche beider Kurven, die mit der Höhe der zu erwartenden Fehlerraten korrespondiert, wird dadurch verringert. In Beispielen, in denen die TFBS-BN-Modellierung die PWM-Modelle nicht übertreffen konnte, wird die Schnittfläche hauptsächlich durch eine Vergrößerung der Varianz der Verteilung der $\Omega_{\neg\mathsf{TFBS}}$-Bewertungen erhöht, während die Mittelwerte weiterhin einen größeren Abstand als für PWM-Modelle haben.

Anzahl der Modell-Parameter. Ein mutmaßlicher Nachteil von BN-Klassifikatoren gegenüber PWM-Modellen ist die erhöhte Anzahl von Wahrscheinlichkeitsparametern, die sich aufgrund der Modellierung von Abhängigkeiten mittels bedingter Wahrscheinlichkeitsverteilungen ergibt.

Ein überraschendes Ergebnis der Untersuchungen ist, dass durch den Einsatz aussagekräftiger Merkmale bei TAN- und NB-Modellen in der Mehrheit weniger Parameter benötigt wurden als bei PWM-Modellen. Dazu trugen vor allem $\mathcal{M}_{CON}$-Merkmale bei, die für die Modellierung einer hoch konservierten Teilsequenz der Länge W erheblich weniger Freiheitsgrade zulassen als die entsprechende Modellierung dieser Sequenz durch PWM-Spalten mit insgesamt $3 \times W$ Freiheitsgraden. Ein weiterer Grund dürfte die Beschränkung der Abhängigkeiten auf hoch korrelierende Merkmale in den TAN-Modellen sein. Tabelle 5.4 stellt die durchschnittliche Parameteranzahl der verschiedenen Modellvarianten gegenüber.

Untersuchung der Merkmalssuche. Die Bewertung der Robustheit des SFFS-Algorithmus ist schwer einzuordnen. Tabelle 5.5 zeigt die die Ergebnisse bei der Bestimmung des Robustheitsmaßes r. Im Mittel wurden 25% der Merkmale in allen Permutationen einer Stichprobe ausgewählt. Die Empfindlichkeit des Suchalgorithmus gegenüber unterschiedlichen Startmengen war noch etwas größer. Wurden z.B. die Merkmalsmengen von

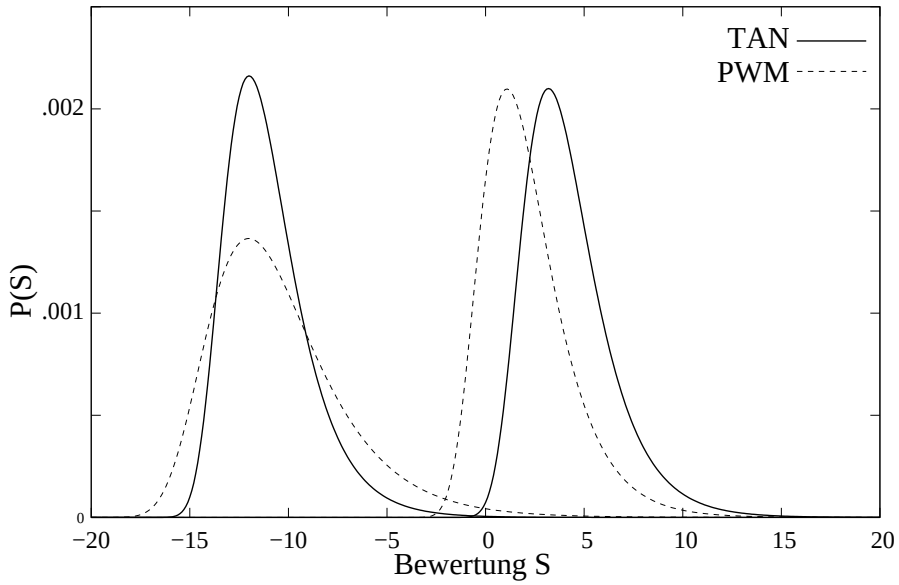


Abbildung 5.9: Gegenüberstellung der *log-odds*-Bewertungsverteilungen zwischen $TAN_{\emptyset}$-Modell und PWM angewendet auf Stichprobe M00930. Die beiden rechten Verteilungen wurden mit den TFBS dieser Stichprobe geschätzt, die beiden linken Verteilungen auf den Negativbeispielen.

	$TAN_{\emptyset}$	TAN_{PWM}	$NB_{\emptyset}$	NB_{PWM}	PWM
Mittlere Parameteranzahl	$33,72$	$63,90$	$34,73$	$52,0$	$87,61$

Tabelle 5.4: Durchschnittliche Anzahl von Modellparametern nach Modellklasse. Gemittelt wurde jeweils über die Modelle für alle Datensätze und den jeweils fünf Versuchsläufen.

$TAN_{\emptyset}$ und TAN_{PWM} miteinander verglichen, enthielten die Schnittmengen im Mittel nur etwas mehr als 15% der Merkmale der Vereinigungsmenge.

Tabelle 5.6 gibt Auskunft darüber, wie häufig Merkmale einer bestimmten Merkmalsklasse in finalen Merkmalsteilmengen vertreten waren, aufgeschlüsselt nach Modellvariationen. Bei Betrachten indvidueller Merkmalsteilmengen fiel auf, dass besonders $\mathcal{M}_{CON}$-Merkmale häufig eine Gruppe von $\mathcal{M}_{PWM}$-Merkmalen ersetzen, mit dem günstigen Effekt einer Parameterreduktion (siehe Tabelle 5.4). Dies ist insbesondere dann der Fall, wenn die Merkmalsauswahl auf der leeren Menge startet. Der Anteil an $\mathcal{M}_{PWM}$-Merkmalen liegt dann deutlich unter dem der Modelle, bei denen die Merkmalsauswahl mit den originalen PWM-Spalten startete. Auffällig in Tabelle 5.6 ist, dass in den NB-Modellen in keinem Fall ein strukturelles Merkmal ($\mathcal{M}_{PWM}$) verwendet wurde. Offensichtlich können diese nur erfolgreich eingesetzt werden, wenn gleichzeitig Abhängigkei-

	$TAN_\emptyset$	TAN_{PWM}	$TAN_\emptyset \cap TAN_{PWM}$	$NB_\emptyset$	NB_{PWM}	$NB_\emptyset \cap NB_{PWM}$
r	$0,2521$	$0,3013$	$0,1505$	$0,2387$	$0,2790$	$0,1443$

Tabelle 5.5: Übereinstimmungsindex r der Merkmalsmengen bei verschiedenen Permutationen und bei verschiedenen Startmengen. Ein Werte nahe 1.0 bezeugt eine hohe Robustheit gegenüber Stichprobenpermutationen.

	$TAN_\emptyset$	TAN_{PWM}	$NB_\emptyset$	NB_{PWM}
$\mathcal{M}_{PWM}$	$14,12$	$50,72$	$18,05$	$57,60$
$\mathcal{M}_{STRUCT}$	$7,68$	$5,22$	0	0
$\mathcal{M}_{CON}$	$77,21$	$43,43$	$75,23$	$38,46$
$\mathcal{M}_{PRF}$	$0,98$	$0,62$	$0,50$	$0,21$

Tabelle 5.6: Relative Häufigkeit der Auswahl von Instanzen der Merkmalsklassen in allen Modellen.

ten zu anderen Merkmalen Berücksichtigung finden.

5.4 BioBayesNet: eine Web-Anwendung zum Einsatz Bayesscher Netze in der Sequenzanalyse

Die softwaretechnische Umsetzung der in diesem Kapitel vorgestellten Konzepte wurde durch eine web-basierte Anwendung ergänzt, die es einem Anwender ermöglicht, für einen gegebenen Sequenzdatensatz Merkmale zu definieren, die besten Merkmale mittels Merkmalsauswahl zu bestimmen und diese für die Konstruktion eines BN-Klassifikators zu verwenden [Nik07]. Die hier vorgestellten Konzepte lassen sich mit Hilfe von *BioBayesNet* auf beliebige Klassifikationsaufgaben anwenden. Optimiert ist *BioBayesNet* jedoch für den Einsatz im Bereich der biologischen Sequenzanalyse.

5.4.1 Anwendungsfälle

Der Funktionsumfang von *BioBayesNet* gliedert sich in drei Szenarios: 1.) Dem Lernen von BN-Klassifikatoren, 2.) dem probabilistischen Schließen mit den gelernten BN-Klassifikatoren und 3.) der Anwendung der BN-Klassifikatoren auf ungesehene Daten.

Lernen. Dieses Benutzungsszenario sieht vor, dass ein Anwender einen etikettierten Lerndatensatz hat und ihn zur Konstruktion eines BN-Klassifikators einsetzen möchte.

Datensätze können in *BioBayesNet* folgender Natur sein:

- eine Menge von biologischen Sequenzen, wobei jede der Sequenzen eine Annotation für ihre Klassenzugehörigkeit besitzt. *BioBayesNet* unterstützt das weit verbreitete Sequenzformat *FASTA*. Die Annotation kann zudem eine Markierung einer Referenzposition enthalten. Liegt eine solche nicht vor, wird Position 1 der Sequenz als Referenz angenommen. Sollen Merkmale auf die Sequenzen angewendet werden, die in *BioBayesNet* nicht implementiert sind, so besteht die Möglichkeit, in den Annotationsteil der Sequenzen die vorberechneten Merkmalsausprägungen zu schreiben. Diese Merkmale werden den durch *BioBayesNet* erzeugten Merkmalen hinzugefügt. Abbildung 5.10a.) zeigt einen kleinen Sequenzdatensatz.
- eine Stichprobe von Merkmalsvektoren. Dieses Format gestattet dem Anwender größtmögliche Freiheit, eigene Merkmale zu definieren und zuvor auf seine Muster anzuwenden. Es bietet sich an, wenn die meisten der gewünschten Merkmale nicht in *BioBayesNet* implementiert sind [11]. *BioBayesNet* verwendet das C4.5-Format, dass von vielen *Machine Learning*-Paketen unterstützt wird (siehe Abbildung 5.10b.))

a.)
```
>Sequenz1 class(A) site(10,15) feature(3.5)
TATAGGAGGATATATAGGGATATATATTATGAGGAGGGATATATTA
>Sequenz2 class(B) site(20,27) feature(10.3)
ACAGGAGGACGCGCACGCATTACTCTCCGCTTACTAGATTATA
>Sequenz3 class(A) site(20,10) feature(2.7)
CCCGGCGGCAGGAGAGATTATTAGACGACTCGCTCAGCATCAGTA
  .              .                .
  .              .                .
  .              .                .
>Sequenz300 class(B) site(1,10) feature(23)
CCCCAGAGGATATGAGTAGGACAGCTATATTATGAGCGCAGTTTTG
```

b.)
```
A,B
quality:low,high.
size:1,2,3.
nuc-1:A,C,G,T.
score:continuous.
-----------------
low,1,A,3.5,A
low,2,C,2.7,A
high,1,G,10,B
 :  :  :
 :  :  :
high,1,G,9,B
```

Abbildung 5.10: Mögliche Eingabeformate für Datensätze in *BioBayesNet* . a.) ein Sequenzdatensatz, bestehend aus speziell annotierten FASTA-Sequenzen. b.) Lernstichprobe im C4.5-Format, bestehend aus einer Definitionsdatei und einer Merkmalsvektorendatei.

Hat ein Anwender einen Sequenzdatensatz hochgeladen, gelangt er im nächsten Schritt zu einer Eingabemaske, die ihm die Definition von Merkmalen gestattet. Dabei müssen nicht einzelne Merkmale definiert werden, sondern es können für die freien Parameter einer Merkmalsklasse Bereiche festgelegt werden, für die Merkmale automatisch erzeugt werden sollen.

Nach Abschluss der Definitionsphase oder bei Eingabe einer Stichprobe im C4.5-Format erhält der Anwender eine Übersicht über die erzeugten Merkmale. Er hat nun verschiedene Optionen, eine Teilmenge dieser Merkmale auszuwählen. Zur Beurteilung eines

[11] Insbesondere können so Merkmalsvektoren verarbeitet werden, die rein gar nichts mit der Analyse biologischer Sequenzen zu tun haben.

Merkmales kann er sich dessen Verteilung bezüglich der verschiedenen Klassen anzeigen lassen. Die mächtigste Funktion in dieser Phase ist die automatische Suche einer möglichst diskriminierenden Merkmalsmenge. Dafür wird der SFFS-Algorithmus eingesetzt. Für diesen kann ein Anwender aus verschiedenen Qualitätsfunktionen wählen. Die finale Auswahl kann beliebig auf die Bedürfnisse des Anwenders angepasst werden.

Anschließend wird auf Basis der ausgewählten Merkmale ein TAN-Klassifikator trainiert und evaluiert. Ein Anwender bekommt eine Ergebnisübersicht angezeigt, die ihm die folgenden Informationen bietet:

- Klassifikationsfehlerraten des BN-Klassifikators und davon abgeleitete Maße. Unter anderem bestehen Verknüpfungen zu ROC-Diagrammen und einer tabellarischen Übersicht über die Klassifikation einzelner Stichprobenelemente.
- Leistungsbeitrag der einzelnen Merkmale. Dazu wird die Verschlechterung gemessen, die der Klassifikator durch das Löschen eines Merkmals erfährt.
- der BN-Klassifikator kann im BIF-Format zur späteren Verwendung heruntergeladen werden.
- die erzeugten Merkmalsvektoren können im C4.5-Format heruntergeladen werden.

Probabilistisches Schließen. Über einen Verweis auf der Ergebnisseite hat der Anwender die Möglichkeit, den gelernten BN-Klassifikator zum probabilistischen Schließen einzusetzen. Neben einer grafischen Anzeige des Bayesschen Netzes steht eine Eingabemaske zur Verfügung, in der Beobachtungen einzelner Variablen vorgenommen werden können. Die a posteriori Verteilung einer beliebigen anderen Variablen kann daraufhin abgefragt werden. Als probabilistischer Schließalgorithmus kommt die auf Seite 70 vorgestellte Variableneliminierung zum Einsatz [Dec99].

Klassifikation. Ein Anwender kann einen BN-Klassifikator, den er zuvor trainiert und möglicherweise abgespeichert hat, verwenden, um bisher ungesehene Merkmalsvektoren zu klassifizieren bzw. Sequenzen nach Treffern einer bestimmten Klasse zu durchsuchen.

Dazu muss er im ersten Schritt ein Bayessches Netz im BIF-Format hochladen. Alternativ gelangt er von der Übersichtsseite eines soeben gelernten BN-Klassifikators zur Klassifikationsphase. Enthält das BN nur Merkmale, die von *BioBayesNet* angeboten werden, so kann der Anwender im nächsten Schritt neue Sequenzen eingeben. Anderenfalls müssen Merkmalsvektoren hochladen werden, die mindestens die Merkmale enthalten, die im BN enthalten sind.

Abbildung 5.11 enthält ein Flussdiagramm, dass alle Anwendungsszenarien zusammenfasst.

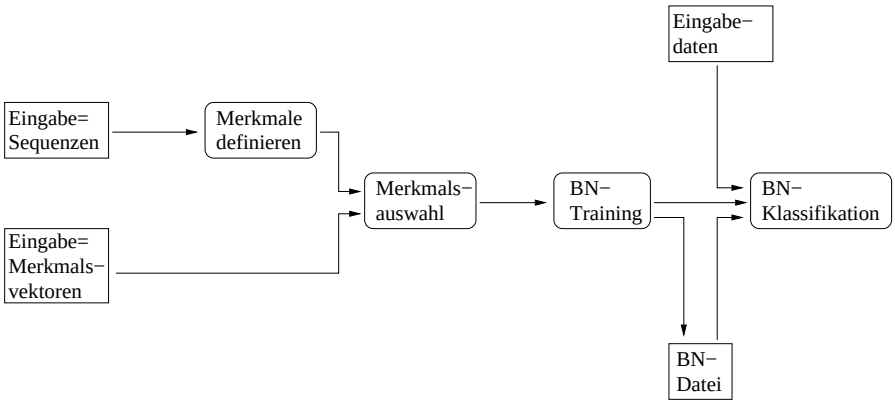


Abbildung 5.11: Ablaufdiagramm für Anwendungsmöglichkeiten in *BioBayesNet* .

5.4.2 Technische Umsetzung

BioBayesNet ist eine Web-basierte *Java Enterprise*-Anwendung in Client-Server-Architektur und wurde gemäß der für diese Spezifikation empfohlenen Entwurfsmustern entwickelt [Ash04]. Diese sehen eine klare Dreiteilung der Anwendung in drei Schichten vor:

1. *Präsentationsschicht* für die Kommunikation mit dem Anwender. In *BioBayesNet* besteht die Kommunikation im Anbieten von Eingabemasken im Webbrowser des Anwenders und der Darstellung von Ergebnissen.
2. *Steuerungsschicht* verarbeitet die Eingaben des Anwenders und leitet sie an die Algorithmenschicht weiter. Ergebnisse der Algorithmenschicht werden anschließend der richtigen Komponente der Präsentationsschicht weitergeleitet.
3. *Algorithmenschicht* [12] enthält alle Komponenten, die im Zusammenhang mit Datenhaltung oder den Algorithmen der Anwendung stehen. In *BioBayesNet* sind in der Logikschicht die Lernverfahren, die Erzeugung von Merkmalen und deren Ausprägungen sowie die Klassifikation von Mustern implementiert.

Als *Java Enterprise*-Anwendung benötigt *BioBayesNet* als Laufzeitumgebung eine spezielle Infrastruktur, den *Application Server*. Dieser automatisiert und kontrolliert die Lebenszeit der Objekte der Anwendung, die Abgrenzung der Sitzungsdaten verschiedener Anwender, die Erzeugung von HTML-Ausgaben sowie *Logging*. Da nicht alle Komponenten eines Java EE *Application Server* benötigt werden, wird der *Web-Container TOMCAT* verwendet, der die genannten Aufgaben erfüllt.

Der grundsätzliche Aufbau von *BioBayesNet* mit seinen drei Schichten ist in Abbildung 5.12 dargestellt und zeigt die Verarbeitungskette eines Anwendungsfalls durch die drei Schichten.

[12] eigentlich Logikschicht

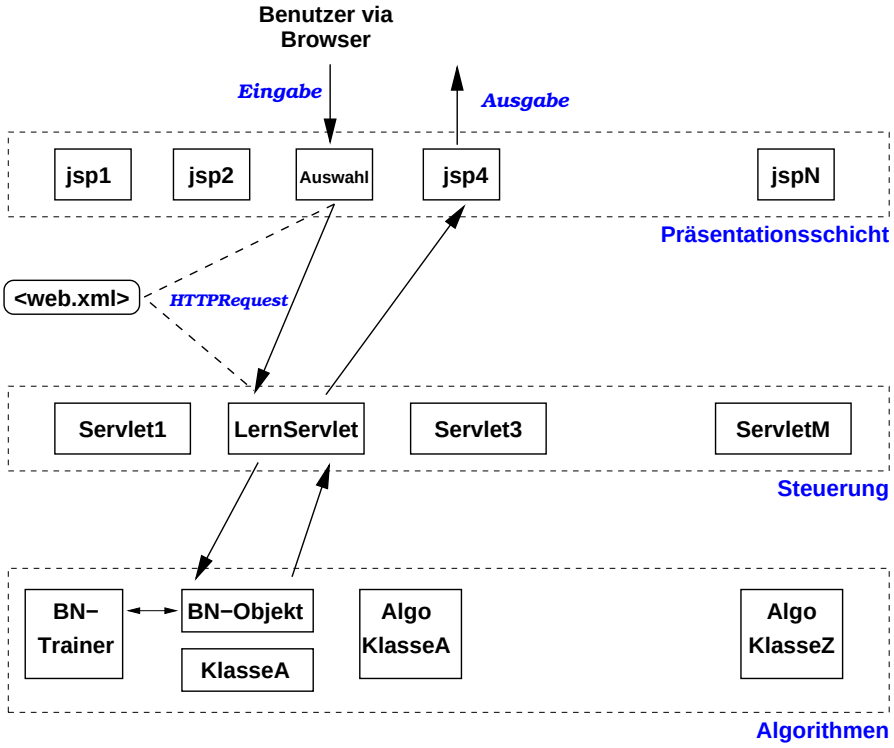


Abbildung 5.12: Dreischichtiger Aufbau von *BioBayesNet* gemäß den Empfehlungen für Web-basierte *Java Enterprise*-Anwendungen.

Algorithmenschicht. Die Algorithmenschicht muss die folgenden Hauptaufgaben bewältigen:

- Erzeugung von Merkmalsvektoren
- Merkmalsauswahl
- Lernen eines BN-Klassifikators
- Klassifikation
- Probabilistisches Schließen

Merkmalsklassen werden durch Ableiten der abstrakten Klasse `ModelFeature` [13] implementiert. Wie in Abbildung 5.13 dargestellt, besitzen diese Klassen die zentralen Methoden zum Berechnen eines Wertes für ein Sequenzmuster (mit Position i_ω), bzw. einer Wertefolge für eine Sequenz beim Iterieren über alle Positionen dieser Sequenz. Eine Merkmalsklasse enthält jeweils die in Abschnitt 5.1 beschriebenen Parameter als Eigenschaften. Ein konkretes Merkmal wird durch Instantiierung eines Objektes von einer Merkmalsklasse erzeugt. Eine Merkmalsteilmenge wird durch die Klasse

[13] Auf die Java-Paketnamen wird der Übersicht zuliebe verzichtet.

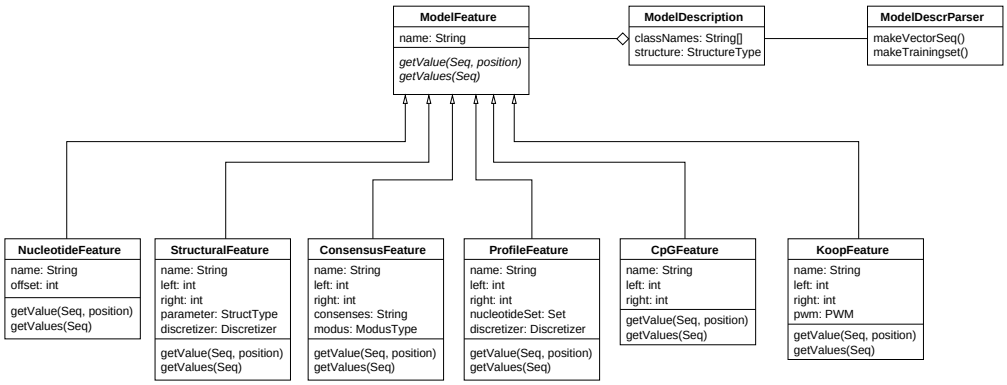


Abbildung 5.13: OO-Modellierung der Merkmalsklassen (stark vereinfacht). Ein Merkmal ist ein Objekt einer Unterklasse von `ModelFeature`. Eine Menge von Merkmalen ist eine Aggregation von Merkmalen in einer `ModelDescription` . Diese wird von einem Parser angewendet, der Methoden der einzelnen Merkmale zur Werterzeugung aufruft.

`ModelDescription` implementiert. Sie enthält neben einem Vektor von Merkmalsobjekten Informationen über die zu unterscheidenden Klassen und die angestrebte BN-Strukturklasse.

Das Umwandeln einer biologischen Sequenz in eine Folge von Merkmalsvektoren übernimmt die statische Methode `produceCaseSet()` der Klasse `ModelDescriptionParser.` Diese Methode besitzt zwei Argumente: 1.) Eine Sequenz und 2.) ein `ModelDescription` -Objekt. Eine zweite Methode erlaubt die Herstellung einer Stichprobe von Merkmalsvektoren aus einem Sequenzdatensatz. Ein solcher Datensatz wird durch die Klasse `TrainingSet` implementiert.

Die Merkmalsauswahl wird gemäß des Entwurfsmusters *Strategie* entworfen und implementiert. Es besteht aus einer Familie von gegenseitig austauschbaren Algorithmenklassen, die jeweils ein Merkmalsauswahlverfahren implementieren. Von Bedeutung ist schließlich nur der SFFS-Algorithmus geblieben. Jedes Merkmalsauswahlverfahren kann mit verschiedenen Qualitätsmaßen $J(\cdot)$ ausgeführt werden, die durch Ableiten einer abstrakten Klasse `QualityMeasure` realisiert wurden.

Für das Training und die Anwendung der TFBS-BN wurden Hüllklassen entwickelt, welche auf der BN-Datenstruktur von JavaBayes basieren und diese für die Belange dieser Anwendung erweitern. Ein TFBS-BN wird dabei durch die Klasse `BayesianNetworkModel` realisiert. Objekte dieser Klasse können sich selbst anhand einer Stichprobe, die zuvor aus einer Menge von Trainingssequenzen erzeugt wurde.

Die Aufgabe des probabilistischen Schließens übernehmen Klassen von *JavaBayes*, die den Schließalgorithmus implementieren.

Steuerungs- und Präsentationsschicht. Die Anzeige von Eingabemasken und Ausgabedaten im Webbrowser des Anwenders und die Weiterleitung der Eingabe in die Logikschicht übernimmt ein Zusammenspiel von den Java-Technologien *Java Servlets* und *Java Server Pages* (JSP).

JSP sind im einfachsten Fall einfache HTML-Seiten. Neben HTML-Formatierungen können JSP auch Java-Quellcode enthalten, die eine dynamische Erzeugung von HTML-Ausgabe ermöglicht. Typischerweise greift eine JSP dazu auf ein Objekt zu, das in den Sitzungsdaten des Anwenders bereit gehalten wird, und beispielsweise die Ergebnisse des letzten Lernvorganges enthält, und gibt die wichtigen Informationen in HTML-formatierter Weise aus. Die Eingabe von Anwenderdaten wird in normalen HTML-Formularen durchgeführt. Eine JSP leitet die Eingabedaten an ein Java Servlet weiter.

Ein Java Servlet ist eine spezielle Java-Klasse zur Verarbeitung von Anwendungsdaten und zur Kommunikation zwischen Logikschicht und Repräsentation. Servlets (und auch JSP) werden dazu in einer speziellen Laufzeitumgebung ausgeführt, einem *JAVA application server*, und gehören demnach zu den Server-seitigen Komponenten der Anwendung. Für *BioBayesNet* wurde hierfür Apache TOMCAT eingesetzt, der nur einen Teil der Java Enterprise Spezifikation implementiert. Ein Java Servlet interpretiert und validiert die Eingabe des Anwenders und leitet ihn entsprechend an die richtige Stelle der Logikschicht weiter, etwa durch Instantiieren von Objekten der Logikschicht und aufrufen deren Methoden. Nachdem das Logikschicht-Objekt seine Aufgabe (z.B. Lernen) erfüllt hat, setzt das Java Servlet die Ergebnisse in den Sitzungsdatenbereich der Anwendung und ruft eine JSP auf, welche die Ausgabe der Ergebnisse für den auf dem Bildschirm eines Anwenders übernimmt.

Verwendete Programm-Bibliotheken. Die Java-API *BioJava* [14] bietet eine reichhaltige Sammlung von Klassen für die Repräsentation, der Ein-und Ausgabe sowie der Verarbeitung von biologischen Sequenzen an. Sequenzen sind dabei keine einfachen Zeichenketten, sondern komplexe Objekte, die Annotationen aufnehmen können und typsicher eine bestimmte Art von biologischen Sequenzen garantieren. Viele alltägliche Aufgaben, wie das Ausschneiden einer Teilsequenz unter Berücksichtigung der positionsspezifischen Annotationen, dem Erzeugen der komplementären Sequenz im Fall von DNA, werden als Methoden angeboten. BioJava ermöglicht das Einlesen und Schreiben aller gängigen Sequenzformate.

BioBayesNet verwendet zudem JavaBayes [Dec99], eine Java-API, die Datenstrukturen für Bayessche Netze und Algorithmen zum Probabilistischen Schließen anbietet. Da diese API hauptsächlich als Algorithmenschicht der gleichnamigen Software dient, war es nötig, entsprechende Schnittstellen zu den Programmteilen von *BioBayesNet* herzustellen. Teilweise wurde dies durch Hüllklassen gelöst, in wenigen Fällen mussten zusätzliche Schnittstellen in JavaBayes implementiert werden.

[14] http://www.biojava.org

Zum Protokollieren (englisch *logging*) von Programmabläufen wurde die API *log4j* eingesetzt. Alle weiteren verwendeten API betreffen die grundsätzliche Funktion der Laufzeitumgebung und sollen hier nicht weiter erwähnt werden.

5.5 Verwendung der TFBS-BN in HMM für TFBS-Module

In Abschnitt 3.3 wurden Ansätze zur Modellierung von TFBS-Modulen vorgestellt. Viele dieser Verfahren basieren auf einer kombinierten Suche mit einer Menge von Sequenzmodellen für einzelne TFBS. Unterschieden werden zwei grundlegende Strategien: 1.) die fensterbasierten Ansätze und 2.) die HMM-Ansätze. Die Anwendung von Bayesschen Netzen als TFBS-Modellansatz in fensterbasierten Strategien anstelle der vorwiegend verwendeten PWM-Modelle ist trivial. Die Einbeziehung der TFBS-BN in Hidden-Markov-Modellen zur TFBS-Modulerkennung soll in diesem Abschnitt untersucht werden.

Unterabschnitt 5.5.1 bietet eine Kurzeinführung in Hidden-Markov-Modelle. In Unterabschnitt 5.5.2 wird der klassische Einsatz von PWM-Modellen zur HMM-basierten Modellierung von TFBS-Modulen im Detail betrachtet. Unterabschnitt 5.5.3 erläutert die Verwendung von TFBS-BN in solchen HMM. Unterabschnitt 5.5.4 enthält Bemerkungen zur softwaretechnischen Umsetzungen.

5.5.1 Hidden-Markov-Modelle

Hidden Markov-Modelle sind stochastische Automaten, die einen zweistufigen Zufallsprozess zur Erzeugung sequentieller Daten modellieren. Die erste Stufe des Zufallsprozesses entspricht einer einfachen, stationären *Markovkette* erster Ordnung über einer endlichen Menge von Zuständen. Als solcher ist er durch eine Übergangsmatrix definiert, welche die Wahrscheinlichkeiten für Wechsel zwischen den Zuständen enthält.

Die zweite Stufe des Zufallsprozesses besteht im Falle der hier betrachteten diskreten HMM darin, dass zu jedem Zeitpunkt der aktuelle Zustand ein Zeichen aus einem Ausgabealphabet ausgibt[15]. Jeder Zustand besitzt dafür eine eigene Ausgabeverteilung. Die Ausgabe eines Zeichens hängt *ausschließlich* von dieser Verteilung (d.h. vom aktuellen Zustand) ab.

DEFINITION 5.9: *Es sei S eine Menge von N inneren Zuständen und $\mathcal{A}$ eine Menge von M Ausgabezeichen. Der zweistufige Zufallsprozess $[U_t, O_t]$ heißt* **diskretes Hidden Markov-Modell (HMM)***, wenn $[U_t]$ eine einfache, stationäre Markovkette über S ist*

[15] In kontinuierlichen HMM werden zu jedem Zeitpunkt Werte aus einem kontinuierlichen Wertebereich gemäß einer zustands-eigenen Wahrscheinlichkeitsdichte gezogen.

und die Ausgabewahrscheinlichkeit eines Zeichens o_t zum Zeitpunkt t nur vom aktuellen Systemzustand abhängt:

$$P(O_t = o_t \,|\, q_1, \ldots, q_t, o_1, \ldots, o_t) = P(O_t = o_t \,|\, U_t = q_t). \tag{5.16}$$

Dass statistische Verhalten eines diskreten HMM ist durch das Parameterfeld

$$\lambda = (\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B}) \tag{5.17}$$

charakterisiert:

$$\boldsymbol{\pi} \in \mathbb{R}^N \quad \text{mit} \quad \pi_i = P(U_1 = s_i) \tag{5.18}$$

$$\boldsymbol{A} \in \mathbb{R}^{N \times N} \quad \text{mit} \quad a_{ij} = P(U_t = s_j \,|\, U_{t-1} = s_i) \tag{5.19}$$

$$\boldsymbol{B} = \{b_1, \ldots, b_N\} \quad \text{mit} \quad \text{Wahrscheinlichkeitsverteilungen } b_j : \mathcal{A} \mapsto \mathbb{R}, \text{ und} \tag{5.20}$$

$$b_j(k) = P(O_t = z_k \,|\, U_t = s_j). \tag{5.21}$$

Der einfacheren Notation zuliebe werden konkrete HMM mit ihrem Parameterfeld λ identifiziert. Die Matrix $\boldsymbol{\pi}$ ist die Matrix der Anfangswahrscheinlichkeiten und Matrix $\boldsymbol{A}$ heißt *Übergangsmatrix*. Als *Struktur* eines HMM wird die Gesamtheit der erlaubten Zustandsübergänge, an denen die Übergangsmatrix also keine Null stehen hat, bezeichnet. Die erlaubten Zustandsübergänge werden in Darstellungen von HMM als gerichtete Kanten eingezeichnet. Namensgebend für HMM ist die Tatsache, dass in der Regel eine Folge von Ausgabezeichen $o_1 \ldots o_T$ vorliegt, die von einem HMM erzeugt wurden, wobei nicht beobachtet werden kann, welche inneren Zustände die einzelnen Zeichen der Folge ausgegeben haben. Die inneren Zustände sind *verborgen* (englisch: *hidden*).

HMM werden in vielen Bereichen der Musteranalyse eingesetzt, in denen zeitliche oder sequentielle Daten verarbeitet werden müssen. So bilden sie ein zentrales Konzept in der automatischen Spracherkennung [ST95], aber auch in der biologischen Sequenzanalyse. Übereinstimmend teilt die Fachliteratur die Anwendung von HMM in drei zentrale Fragen ein (siehe z.B. [Rab89, Dur98]):

Die Frage der Produktionswahrscheinlichkeiten: Mit welcher Wahrscheinlichkeit $P(o \,|\, \lambda)$ hat ein HMM $\lambda = (\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$ eine Ausgabefolge $\boldsymbol{o} = o_1 \ldots o_T$ erzeugt?

$$P(\boldsymbol{o} \,|\, \lambda) = ?$$

Die naive Herangehensweise, um diese Wahrscheinlichkeit zu berechnen, ist die Summation über alle möglichen Zustandspfade, die dass HMM zur Erzeugung der Beobachtung $\boldsymbol{o}$ *durchlebt* haben könnte. Die Berechnung dieser enormen Summe mit $\mathcal{O}(N^T)$ Summanden kann durch Anwendung der dynamischen Programmierung vermieden werden. Der *Vorwärtsalgorithmus* entwickelt die gesuchte Wahrscheinlichkeit zeichenweise, indem er zu jeder Position t (Zeitpunkt) der Ausgabesequenz die *Vorwärtswahrscheinlichkeiten*

$$\alpha_t(j) = P(o_1 \ldots o_t, q_t = s_j \,|\, \lambda) \tag{5.22}$$

mit $j \in \{1, \ldots, N\}$ dafür berechnet, dass die Teilsequenz $o_1 \ldots o_t$ erzeugt wurde und das HMM sich zu Zeitpunkt t in Zustand j befindet. Die Wahrscheinlichkeiten zum Zeitpunkt t werden mit Hilfe der Vorwärtswahrscheinlichkeiten zum Zeitpunkt $t-1$ berechnet. Deshalb hat der Algorithmus eine polynomiale Laufzeit mit Schranke $\mathcal{O}(TN^2)$. Die Lösung für $P(\boldsymbol{o} \,|\, \lambda)$ ergibt sich schließlich durch Aufsummieren der Vorwärtswahrscheinlichkeiten an Position T.

Analog dazu kann die gesuchte Wahrscheinlichkeit rückwärts berechnet werden, mit Hilfe des *Rückwärtsalgorithmus*, der ebenso schrittweise die *Rückwärtswahrscheinlichkeiten*

$$\beta_t(i) = P(o_{t+1} \ldots o_T \,|\, q_t = s_i, \lambda) \tag{5.23}$$

für $i \in \{1, \ldots, N\}$ und $t \in \{1, \ldots, T\}$ entwickelt.

Einerseits werden sowohl Vorwärts- als auch Rückwärtsalgorithmus als Module in weiteren HMM-Aufgaben verwendet. Andererseits ist die Frage nach der Produktionswahrscheinlichkeit der zentrale Punkt in vielen HMM-Anwendungen. In der Bioinformatik werden beispielsweise HMM auf einer Menge von Proteinsequenzen einer gemeinsamen Proteinfamilie trainiert. Die Frage der Zugehörigkeit eines neuen Proteins zu dieser Familie wird über die Wahrscheinlichkeit, dieses Protein mit dem Familien-HMM erzeugt zu haben, beantwortet.

Die Dekodierungsfrage: Welche Folge $\boldsymbol{q} = q_1 \ldots q_T$ von Zuständen hat am wahrscheinlichsten eine bestimmte Ausgabefolge $\boldsymbol{o} = o_1 \ldots o_T$ erzeugt?

$$P(\boldsymbol{q} \,|\, \boldsymbol{o}, \lambda) \Longrightarrow \max$$

Eine typische Anwendung von HMM sieht vor, dass es sich bei den Beobachtungen um gemessene Muster bzw. aufbereitete Muster handelt, die mutmaßlich von inneren Zuständen des HMM erzeugt werden Die verborgenen Zustände enstprechen semantischen Einheiten des Anschauungsbereiches. Ein Beispiel ist die automatische Spracherkennung. Dort bestehen die Beobachtungen möglicherweise aus einer Sequenz von Kurzzeitspektren gesprochener Sprache. Die HMM-Zustände repräsentieren die Phoneme der zu erkennenden Sprache, wobei jeder Phonemzustand die typischen Kurzzeitspektren des Phonems in seiner Ausgabeverteilung codiert. Der Vorgang der Spracherkennung besteht darin, die Zustandsfolge und damit die Phonemfolge zu bestimmen, die am wahrscheinlisten eine Sequenz von Kurzzeitspektren erzeugt haben könnte.

Allgemein geht es bei der Dekodierungsaufgabe darum, eine optimale Zustandsfolge für eine Beobachtungssequenz zu bestimmen. Die Frage nach der optimalen Zustandsfolge zur Produktion einer Beobachtungssequenz $\boldsymbol{o}$ ist nicht so eindeutig zu beantworten wie die Frage nach den Produktionswahrscheinlichkeiten. Der Grund hierfür sind unterschiedliche Auffassungen davon, was als *optimal* zu bezeichnen ist. Für verschiedene Optimalitätskriterien gibt es verschiedene Verfahren, die bezüglich dieser Kriterien optimale Zustandsfolge ermitteln können.

Eine Möglichkeit besteht darin, zu jedem Zeitpunkt t genau den Zustand $\hat{q}_t$ zu wählen, der die maximale *a posteriori Zustandswahrscheinlichkeit* besitzt:

$$\hat{q}_t = \underset{j}{\operatorname{argmax}}\, P(q_t = j \,|\, \boldsymbol{o},\, \lambda). \tag{5.24}$$

Die daraus zusammengesetzte Zustandsfolge $\hat{\boldsymbol{q}}$ heißt *Maximum a posteriori* Zustandsfolge (MAP-Zustandsfolge) der Beobachtungssequenz $\boldsymbol{o}$. Die a posteriori Zustandswahrscheinlichkeiten können aus den Vorwärts- und Rückwärtswahrscheinlichkeiten vermöge

$$P(q_t = j \,|\, \boldsymbol{o},\, \lambda) = \frac{\alpha_t(j) \cdot \beta_t(j)}{\sum_{i=1}^{N} \alpha_t(i) \cdot \beta_t(i)} \tag{5.25}$$

berechnet werden. Die Strategie, sich zu jedem Zeitpunkt für den lokal wahrscheinlichsten Zustand zu entscheiden, hat zur Folge, dass die MAP-Zustandsfolge unter Umständen nicht mit der HMM-Struktur vereinbar ist.

Der *Viterbi*-Algorithmus ist ein weiteres Dekodierungsverfahren, dass anstelle individueller Maximierungen eine ganze Zustandsfolge $\boldsymbol{q}^*$ mit maximaler a posteriori Wahrscheinlichkeit bestimmt:

$$P(\boldsymbol{q}^* \,|\, \boldsymbol{o}, \lambda) = \max_{\boldsymbol{q} \in S^T} P(\boldsymbol{q} \,|\, \boldsymbol{o}, \lambda). \tag{5.26}$$

bestimmt. Eine solche Zustandsfolge heißt *Viterbi-optimal*. Der Viterbi-Algorithmus maximiert stellvertretend die gemeinsame Wahrscheinlichkeit $P(\boldsymbol{q}, \boldsymbol{o} \,|\, \lambda)$ von Beobachtungssequenz $\boldsymbol{o}$ und Zustandsfolge $\boldsymbol{q}$, denn für eine Viterbi-optimale Zustandsfolge ist wegen

$$P(\boldsymbol{q} \,|\, \boldsymbol{o}, \lambda) = \frac{P(\boldsymbol{q}, \boldsymbol{o} \,|\, \lambda)}{P(\boldsymbol{o} \,|\, \lambda)} \tag{5.27}$$

auch diese maximal. Der Viterbi-Algorithmus ist fast identisch mit dem Vorwärtsalgorithmus mit dem Unterschied, dass anstelle von Summationen im Viterbi-Algorithmus Maximierungsoperatoren angewendet werden. Zur Rekonstruktion des Pfads werden jeweils an den Maxima einer jeden Position Markierungen gesetzt, die anschließend zurückverfolgt werden.

Die Lernfrage: Wie müssen die Parameter des HMM gewählt werden, so dass die Wahrscheinlichkeit, die Ausgabefolge $\boldsymbol{o} = o_1 \ldots o_T$ zu erzeugen, maximal ist?

$$\hat{\lambda} = \underset{\lambda}{\operatorname{argmax}}\, P(\boldsymbol{o} \,|\, \lambda)$$

Die Parameter eines HMM λ, d.h. Anfangs- und Übergangswahrscheinlichkeiten sowie die Ausgabeverteilungen der einzelnen Zustände können anhand einer Stichprobe d von Beobachtungssequenzen geschätzt werden. Läge zu jeder Beobachtung $\boldsymbol{o}$ auch eine gewünschte Zustandsfolge vor, so handelte es sich um eine überwachte Stichprobe. Gerade bei HMM-Anwendungen ist das ein seltener Fall. Meist sind die Lerndaten also

unvollständig und die Parameter des HMM müssen mit iterativen, unüberwachten Lernverfahren gelernt werden. Zwei bekannte Verfahren sind das *Viterbi-Training* und der *Baum-Welch-Algorithmus*.

Das Viterbi-Training ist ein entscheidungsüberwachtes Lernverfahren. Es wird zunächst von einem initialen HMM ausgegangen[16]. In Iterativer Weise wird nun für das aktuelle HMM und jede Beobachtungssequenz $\boldsymbol{o} \in \mathsf{d}$ eine Viterbi-Dekodierung durchgeführt, diese Dekodierung stellvertretend als vollständige Daten verwendet und mit diesen die Parameter des HMM gemäß Maximum-Likelihood-Schätzung gelernt. Das Viterbi-Training findet ein lokales Maximum der kummulierten Viterbi-Bewertung der Stichprobe d, aber kein lokales Maximum der Stichprobenwahrscheinlichkeit (ML-Zielfunktion).

Der Baum-Welch-Algorithmus ist der EM-Algorithmus angewendet auf die Parameter eines HMM. Das EM-Prinzip wurde bereits in Abschnitt 3.2.2 auf Seite 54 eingeführt und spielt in Kapitel 7 eine zentrale Rolle. Als EM-Instanz findet der Baum-Welch-Algorithmus ein lokales Maximum der ML-Zielfunktion.

5.5.2 TFBS-Modulerkennung mit HMM

Zur Beschreibung eines HMM zur Erkennung von TFBS-Modulen wird von einer Modellstruktur ausgegangen, die erstmals in [Fri01] veröffentlicht wurde (siehe auch Abschnitt 3.3.2 auf Seite 58). Ein solches HMM ist in Abbildung 5.14 a.) dargestellt.

Das Frith'sche HMM operiert auf dem Ausgabealphabet $\Sigma_{DNA} \cup \{\mathtt{N}\}$. Zu einem Zeitpunkt t erzeugt das HMM das t-te Nukleotid einer DNA-Sequenz[17].

Die Struktur des HMM entspricht den Erwartungen, dass in einer langen DNA-Sequenz gelegentlich ein TFBS-Modul auftritt. Alle Sequenzpositionen, die nicht einem TFBS-Modul angehören, werden von einem einzigen Lückenzustand erzeugt. Dieser Lückenzustand hat eine Ausgabeverteilung, die sich an der empirischen Nukleotidverteilung der analysierten Sequenz orientiert. Die Übergangsmatrix sieht vor, dass das HMM entweder vom Lückenzustand erneut in diesen wechselt, oder dass das HMM mit der Erzeugung eines TFBS-Moduls beginnt.

Der Modulbereich besteht aus einer parallelen Verknüpfung von probabilistischen PWM-Modellen für mögliche TFBS eines Moduls. Das wird dadurch realisiert, dass für jede Spalte eines PWM ein HMM-Zustand definiert wird. Die Wahrscheinlichkeiten einer Spalte definieren die Ausgabeverteilung des Zustands. Zwischen aufeinanderfolgenden Spalten ist eine Übergangswahrscheinlichkeit von 1 festgelegt. Hat das HMM erst einmal den ersten HMM-Zustand eines PWM-Modells eingenommen, werden nacheinander

[16] Dieses initiale Modell darf keine Gleichverteilungen enthalten. Auf Null gesetzte Parameter werden stets Null bleiben.

[17] oder mit einer für alle Zustände konstanten Wahrschlichkeit $p_{\mathtt{N}}$ den IUPAC-Platzhalter `N` für ein undefiniertes Nukleotid

die zu den anderen PWM-Spalten gehörenden Zustände eingenommen, und die Nukleotide einer TFBS erzeugt. Für jeden berücksichtigten Transkriptionsfaktor werden zwei PWM-Modelle eingebaut, da die DNA-Sequenz nur auf einem Strang und einer Richtung durchsucht wird, und mögliche TFBS auf dem Gegenstrang durch PWM für revers-komplementäre TFBS repräsentiert werden.

Die Erzeugung eines TFBS-Moduls beginnt mit der Einnahme des stillen Zustandes, welcher der parallelen Anordung der PWM-Modelle vorgelagert ist[18]. Die Übergangswahrscheinlichkeiten von dem stillen Zustand in die PWM-Modelle sind bei Frith et al. konstant, wenngleich sie die Möglichkeit diskutieren, über diese Verteilung die erwartete Häufigkeit bestimmter TFBS zu berücksichtigen. Nach Erzeugung einer TFBS besteht gemäß HMM-Struktur die Wahl, einen innermodularen Zwischenraum zu erzeugen, oder das TFBS-Modul abzuschließen.

Im Frith'schen HMM gibt es drei frei wählbare Übergangswahrscheinlichkeiten, α, β und γ, mit denen folgende Eigenschaften der gesuchten TFBS-Module eingestellt werden können

- mittlerer Abstand a zwischen zwei TFBS eines Moduls: $\alpha = \frac{1}{a+1}$
- mittlere Anzahl b von TFBS innerhalb eines Moduls: $\beta = \frac{1}{b}$
- mittlerer Abstand g zwischen zwei TFBS-Modulen: $\gamma = \frac{1}{g+1}$

Das HMM von Frith et al. wird angewendet, in dem die MAP-Zustandsfolge einer Eingabesequenz berechnet wird. Ein Sequenzbereich wird als TFBS-Modul ausgegeben, wenn für jede Position des Bereichs die Summe der a posteriori Wahrscheinlichkeiten der modul-relevanten HMM-Zustände eine gewisse Schranke überschreiten. Innerhalb eines vorhergesagten Moduls werden einzelne TFBS an Stellen vorhergesagt, an denen der MAP-Zustand die erste Position einer entsprechenden PWM ist und die dazugehörige Wahrscheinlichkeit über einer weiteren Schranke liegt.

5.5.3 Integration von Bayesschen Netzen

In diesem Kapitel wurde bei der Suche von TFBS von der DNA-Sequenz als zu durchsuchendes Objekt abstrahiert. Die dort entwickelten Bayesschen Netze operieren auf einer Sequenz von Merkmalsvektoren, wobei sich jeder Merkmalsvektor einer Position der originalen DNA-Sequenz zuordnen lässt. Die Grundidee in diesem Abschnitt ist es, das DNA-Sequenzen erzeugende HMM von Frith et al. in ein HMM umzuwandeln, das den kompletten Merkmalsraum einer Menge von TFBS-BN als Ausgabealphabet hat. Die Ausgabeverteilungen der HMM-Zustände werden durch die TFBS-BN modelliert.

[18] Stille Zustände sind spezielle Zustände, die keine Ausgabe erzeugen. Ihre Verwendung erfordert eine einfache Anpassung der Dekodieralgorithmen. Stille Zustände sind ein Hilfsmittel, um die Struktur eines HMM übersichtlich zu halten. Prinzipiell kann ein HMM mit stillen Zuständen durch eines ohne stille Zustände emuliert werden.

Seien also zunächst TFBS-BN $\mathcal{C}_1, \ldots, \mathcal{C}_K$ gegeben. Da alle Zustände eines HMM das gleiche Ausgabealphabet besitzen müssen, sei zunächst angenommen, dass jedes Modell $\mathcal{C}_\kappa$ eine gemeinsame Verteilung über dem selben Merkmalsraums $\mathsf{D} = D_{X_1} \times \ldots \times D_{X_d}$ modelliert, der von einer Merkmalsmenge $\boldsymbol{X} = \{X_1, \ldots, X_d\}$ aufgespannt wird. Nachdem der Aufbau des HMM beschrieben wurden ist, wird das Vorgehen beschrieben, dass diese Forderung unnötig macht.

TFBS-BN-Modelle sind BN-Klassifikatoren, die dafür prädestiniert sind, die a posteriori Wahrscheinlichkeiten einer Auswahl von Klassen in Gegenwart eines Merkmalsvektors auszugeben. Sie besitzen zusätzlich zu den genannten Merkmalsvariablen eine Klassenvariable und modellieren deshalb eigentlich eine gemeinsame Verteilung über

$$D_{X_1} \times \ldots \times D_{X_d} \times \{\Omega_{TFBS}, \Omega_{\neg TFBS}\} \tag{5.28}$$

Um trotzdem eine Verteilung über den gewünschten Merkmalsraum zu erhalten, müsste über die Klassenvariable marginalisiert werden, so dass für ein Modell $\mathcal{C}$ und einen Merkmalsvektor $(x_1, \ldots, x_d)$ gilt:

$$\begin{aligned} P_{\mathcal{C}}(x_1, \ldots, x_d) &= P_{\mathcal{C}}(x_1, \ldots, x_d \,|\, \Omega_{TFBS}) \cdot P_{\mathcal{C}}(\Omega_{TFBS}) && (5.29) \\ &\quad + P_{\mathcal{C}}(x_1, \ldots, x_d \,|\, \Omega_{\neg TFBS}) \cdot P_{\mathcal{C}}(\Omega_{\neg TFBS}) && (5.30) \end{aligned}$$

Da die a priori Wahrscheinlichkeiten $P_{\mathcal{C}}(\Omega_{TFBS})$ und $P_{\mathcal{C}}(\Omega_{\neg TFBS})$ keine Bedeutung haben, kann die Marginalisierung durch Setzen von

$$P_{\mathcal{C}}(\Omega_{TFBS}) = 1 \tag{5.31}$$

umgangen werden.

Jeder HMM-Zustand, abgesehen von den stillen Zuständen des Modells, wird im Folgenden durch ein BN repräsentiert. Für die beiden Lückenzustände werden analog zur globalen Nukleotidverteilung des Frith'schen HMM TFBS-Modelle auf einer repräsentativen Menge von Merkmalsvektoren, die aus durchschnittlichen DNA-Sequenzmustern erzeugt wurden, trainiert[19]. Eine HMM-Zustandskette, die im Frith'schen HMM eine PWM repräsentiert, wird durch einen einzigen HMM-Zustand ersetzt, der mit seinem TFBS-BN die betreffenden TFBS anstelle der PWM modelliert. Abbildung 5.14 b.) zeigt ein hybrides HMM/BN, dass hinsichtlich der Struktur dem Originalmodell von Frith et al. entspricht.

Unterschiedliche Merkmalsmengen. Eine wesentliche Stärke der TFBS-BN ist es, dass jedes Modell nur die charakteristischen Merkmale von TFBS eines bestimmten Transkriptionsfaktors modelliert. Üblicherweise besitzen zwei verschiedene TFBS-BN völlig unterschiedliche Merkmalsmengen. Die Zustände eines HMM müssen jedoch jeweils das selbe Ausgabealphabet besitzen. Eine einfache Möglichkeit, die Kompatibilität

[19] Ebenfalls analog zu Frith et al. werden zum Lernen des Lückenmodells alle Merkmalsvektoren der aktuell zu durchsuchenden DNA-Sequenz verwendet.

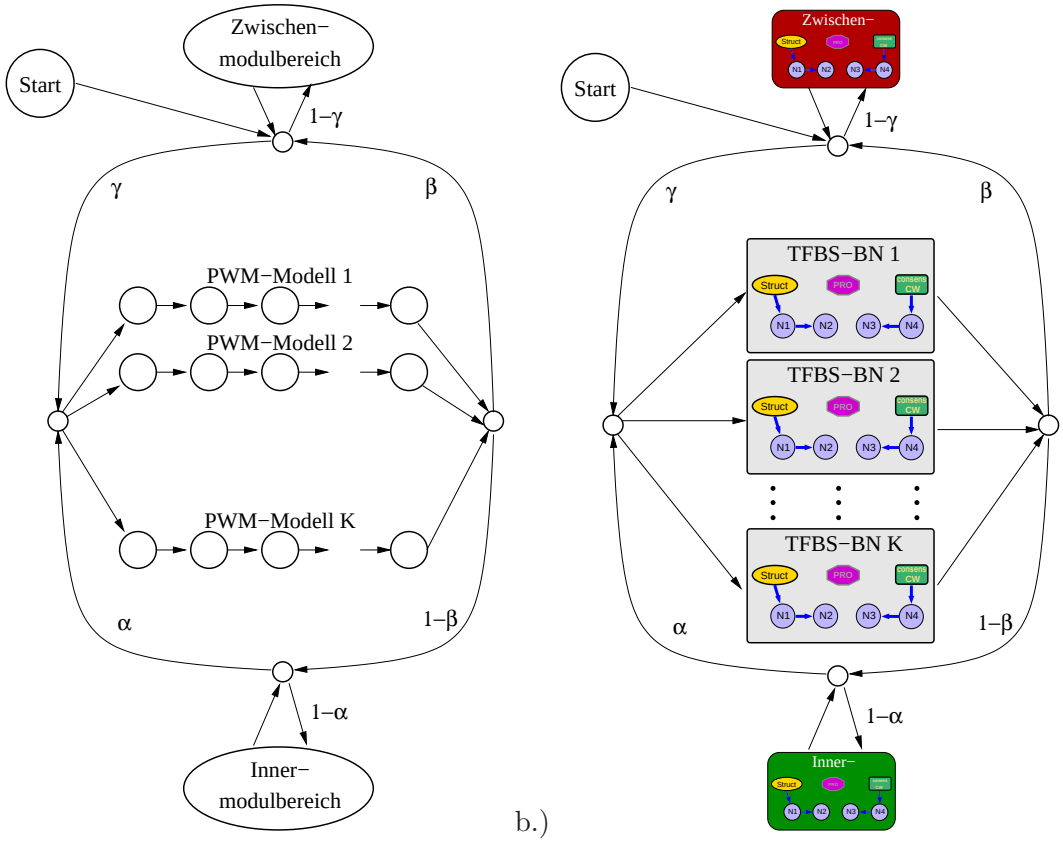


Abbildung 5.14: Gegenüberstellung von HMM für TFBS-Module. a.) Verwendung von PWM-Modellen als elementares Sequenzmodell. Dieses HMM ist ein DNA-Sequenz erzeugendes stochastisches Modell. b.) Anstelle aller nukleotidausgebenden Zustände treten Bayessche Netze. Diese hybride HMM erzeugen Sequenzen von Merkmalsvektoren.

der Zustände herzustellen, ist es, in jedes TFBS-BN zusätzlich all jene Merkmale einzufügen, die in mindestens einem der anderen Modelle vorkommen. Das würde jedoch die Klassifikationsleistung der TFBS-BN schwächen, da die neuen hinzugefügten Merkmale im besten Fall nicht diskriminierend für die modellierten TFBS sind und im schlechtesten Fall redundant und widersprüchlich zu den ursprünglichen Merkmalen des TFBS-BN sind.

Deshalb wird eine andere Methode angewendet, um die Uniformität der Zustandsalphabete herzustellen. Sei $\boldsymbol{U}$ die Menge aller Merkmale, die mindestens in einem der betrachteten TFBS-BN vorkommen. Sei $\mathcal{N}$ ein Hintergrund-BN für die Lückenzustände. Dieses wird für die Merkmalsmenge $\boldsymbol{U}$ trainiert. Damit ist klar, dass alle anderen HMM-Zustände ebenfalls das durch $\boldsymbol{U}$ definierte Ausgabealphabet unterstützen müssen, d.h. eine gemeinsame Verteilung über $\boldsymbol{U}$ bereitstellen müssen. Sei $\boldsymbol{F} \subset \boldsymbol{U}$ die Merkmalsmenge eines TFBS-BN $\mathcal{C}_{\boldsymbol{F}}$ und $\boldsymbol{G} \subset \boldsymbol{U} \setminus \boldsymbol{F}$ die Menge aller Merkmale, die nicht in

$\mathcal{C}_{\boldsymbol{F}}$ modelliert werden. Da angenommen werden kann, dass die Merkmale in $\boldsymbol{G}$ keinen Beitrag zur Klassifikation von den in $\mathcal{C}_{\boldsymbol{F}}$ modellierten TFBS haben[20], wird der Anteil der $\boldsymbol{G}$-Merkmalsausprägungen an der gemeinsamen Wahrscheinlichkeit durch das Hintergrundmodell $\mathcal{N}$ berechnet. Augehend von

$$P(\boldsymbol{u}) = P(\boldsymbol{f}, \boldsymbol{g}) \quad = \quad P(\boldsymbol{f}) \cdot P(\boldsymbol{g} \mid \boldsymbol{f}) \tag{5.32}$$

wird der erste Teil des Produkts auf der rechten Seite durch das TFBS-BN $\mathcal{C}_{\boldsymbol{F}}$ berechnet, der zweite Teil, der nicht durch $\mathcal{C}_{\boldsymbol{F}}$ modelliert wird, durch das Modell $\mathcal{N}$ der Lückenzustände:

$$P(\boldsymbol{u}) = P(\boldsymbol{f}, \boldsymbol{g}) \quad \sim \quad P_{\mathcal{C}_{\boldsymbol{F}}}(\boldsymbol{f}) \cdot P_{\mathcal{N}}(\boldsymbol{g} \mid \boldsymbol{f}). \tag{5.33}$$

Für die Berechnung der bedingten Wahrscheinlichkeiten $P_{\mathcal{N}}(\boldsymbol{g} \mid \boldsymbol{f})$ wird ein probabilistisches Schließverfahren eingesetzt – den *BucketTreeElimination* Algorithmus [Kas01], implementiert in der Java-API *JavaBayes*. Ein Vorteil dieses Schließalgorithmus ist es, dass er dafür konzipiert ist, eine große Anzahl von Anfragen bezüglich einer festen Menge von Variablen und einer festen Menge von beobachteten Variablen durchzuführen.

Dekodierung. Die Dekodierung funktioniert für hybride HMM/BN völlig analog zu gewöhnlichen HMM. In der experimentellen Implementierung werden jedoch im Gegensatz zu Frith et al. sowohl die MAP-Zustandsfolge als auch der Viterbi-Pfad bestimmt. Ein TFBS-Modul und einzelne TFBS werden gemäß dem Viterbi-Pfad bestimmt, die Stärke der einzelnen TFBS anhand der a posteriori Wahrscheinlichkeit bestimmt.

5.5.4 Details der Implementierung.

Zur Anwendung der hybriden HMM/BN wurde eine Java-basierte Web-Anwendung geschaffen. Über eine Benutzerschnittstelle können TFBS-BN zur Berücksichtigung im HMM ausgewählt und hochgeladen werden. Zur Auswahl stehen gegenwärtig Modelle für die in Abschnitt 5.3 untersuchten 86 Transkriptionsfaktoren. Hochgeladen werden können z.B. BN-Klassifikatoren, die mit Hilfe von *BioBayesNet* trainiert wurden.

Die Implementierung gewährt dem Anwender im Vergleich zu dem Frith'schen HMM zusätzliche Freiheiten bei der Gestaltung der HMM-Struktur. Erstens können zwei TFBS-Modelle in Reihe geschaltet werden, um ein obligatorisch gemeinsames Auftreten beider TFBS zu erzwingen. Zweitens ist es möglich, für jeden berücksichtigten Transkriptionsfaktor zu entscheiden, auf welchen DNA-Strängen Treffer erlaubt sind. Drittens kann über die Übergangswahrscheinlichkeiten vom vorgelagerten stillen Zustand zu den TFBS-BN Wissen darüber kodiert werden, dass einige TFBS-Typen wesentlich seltener sind als andere.

[20] ansonsten wären diese Merkmale wohl in $\mathcal{C}_{\boldsymbol{F}}$ berücksichtigt worden.

Nach Definition eines Modul-HMM kann dieses verwendet werden, um Eingabesequenzen nach TFBS-Modulen zu durchsuchen. Der Anwender erhält eine grafische Darstellung der Suchergebnisse.

5.6 Diskussion und Ausblick

In diesem Kapitel wurde ein stochastischer Modellierungsansatz für Transkriptionsfaktorbindungsstellen entwickelt, der die hohen Klassifikationsfehlerraten von PWM-Modellen dadurch zu verringern versucht, dass die TFBS durch eine Menge komplexer, aber sehr charakteristischer Merkmale beschrieben werden, und diese Merkmale in Bayesschen Netz-Klassifikatoren modelliert werden. Merkmale sind hier Funktionen, die DNA-Sequenzstücken gemäß einer definierten Semantik einen Wert ihrer Wertemenge zuweisen. Interessant sind Merkmale, deren Werteverteilung, wenn angewendet auf eine bestimmte Klasse von TFBS, stark von der Werteverteilung bei durchschnittlichen DNA-Sequenzen unterscheidet. Solche Merkmale diskriminieren besonders stark zwischen den beiden Klassen Ω_{TFBS} und $\Omega_{\neg\mathsf{TFBS}}$. Zur Identifizierung einer möglichst kleinen Teilmenge von Merkmalen, die gemeinsam besonders stark diskriminieren, wurde das SFFS-Merkmalsauswahlverfahren eingesetzt. Die resultierenden Teilmengen wurden in TAN-Klassifikatoren modelliert.

Diskussion der Ergebnisse. Der Modellierungsansatz wurde eingesetzt, um TFBS richtig zu erkennen. Die Erkennungsleistung wurde auf vielfältige Weise gemessen und mit der des gegenwärtigen Standardmodells, dem PWM, verglichen. Als zusätzliche Zwischenlösung wurden auch NB-Klassifikatoren untersucht.

TAN-Modelle und NB-Modelle zeigten in der Mehrheit der Datensätze eine bessere Erkennungsleistung als PWM-Modelle. In diesen Fällen lagen die FP-Raten für eine fest eingestellte TP-Rate weit unter denen eines PWM-Modells. Dies drückt sich auch in der signifikant höheren Fläche unter der ROC-Kurve aus. In den meisten Fällen wiesen die TAN- und NB-Modelle entgegen früher Befürchtungen weniger freie Modellparameter auf als das dazugehörige PWM-Modell. TAN- und NB-Modelle erreichten also in sparsamerer Weise niedrigere Fehlerraten. Auffällig war zudem, dass in den meisten Fällen völlig andere Merkmale ausgewählt wurden als die originalen $\mathcal{M}_{PWM}$-Merkmale.

NB-Modelle und TAN-Modelle zeigten meist nahezu identische Ergebnisse, was auf Eigenheiten des Merkmalsraums und auf den verwendeten Suchalgorithmus zurückzuführen ist. So tendierte der Suchalgorithmus dazu, möglichst unabhängige Merkmale für TAN auszuwählen, anstatt schwache Abhängigkeiten zwischen schwach redundanten Merkmalen zu berücksichtigen.

Alternativen für Bayesschen Netze. Am Anfang der Entwicklung der TFBS-Modelle stand die Idee, TFBS flexibler durch komplexere Merkmale zu beschreiben als dies durch die spaltenweisen Nukleotidverteilungen in PWM-Modellen geschieht. Eine konkrete TFBS wurde nun nicht durch die DNA-Sequenz repräsentiert, sondern allgemeiner über einen Vektor von Merkmalsausprägungen. Der zweite Schritt war die Modellierung der gemeinsamen Verteilung dieser Merkmalsvektoren in Bayesschen Netzen.

In Vorträgen und im Begutachtungsprozess der Veröffentlichung [Pud05] wurde ich häufig mit der Frage konfrontiert, warum anstelle von Bayesschen Netzen nicht die in den letzten Jahren populären *Support-Vector-Maschinen* (SVM) oder gar *Hidden-Markov-Modelle* eingesetzt wurden.

SVM gelten als leistungsfähige Zwei-Klassen-Klassifikatoren, die mit Merkmalsvektoren sehr hoher Dimension umgehen können. Anhand einer etikettierten Stichprobe wird eine Hyperebene in den hochdimensionalen Merkmalsraum gelegt, wobei die Breite der Umgebung der Ebene, in denen keine Stichprobenelemente auftreten, maximiert wird. Da eine solche *ebene* Hyperebene meist nicht gefunden werden kann, wird der Merkmalsraum mittels einer geeigneten Funktion, dem *Kern*, in einen noch höheren Raum transformiert. Die Rücktransformation der dort geschätzten Hyperebene erscheint im originalen Merkmalsraum häufig als beliebig gebogene Fläche, welche die Klassenbereiche nahezu perfekt trennt. SVM sind beliebt, weil sie eine vorherige Merkmalsauswahl unnötig machen und sie sehr gut auf bisher ungesehene Merkmalsvektoren abstrahieren können.

Problematisch ist ihr Einsatz hingegen, wenn diskrete oder gar nicht-nominale Merkmale verwendet werden sollen. Da dies für einige der wichtigsten der hier vorgestellten Merkmalsklassen zutrifft, sind SVM keine günstige Wahl für das bearbeitete Klassifikationsproblem. Die in der Fachliteratur veröffentlichten Arbeiten zur Sequenzanalyse, die SVM eingesetzt haben [Dro04], verwendeten stellvertretend für die nicht-nominalen Sequenzmerkmale Zählstatistiken von Nukleotiden in einer gewissen Teilsequenz. Ein weiterer Nachteil gegenüber Bayesschen Netzen ist, dass die Möglichkeit, die Gründe für eine Klassifikation zu erforschen, aufgrund der komplexen Struktur von Kern und Hyperebene stark beschränkt ist, während das Verhalten von Bayessche Netzen sehr intuitiv begreifbar ist.

HMM haben außer einer grafischen Ähnlichkeit wenige Gemeinsamkeiten mit Bayesschen Netzen. Während Bayessche Netze die gemeinsame Verteilung einer Menge von Zufallsvariablen repräsentieren, sind HMM Automaten, welche die Erzeugung von sequentiellen Daten über einen zweistufigen Zufallsprozess beschreiben. Ein HMM besitzt eine Menge von Zuständen, wobei die Wahrscheinlichkeit für das Einnehmen eines solchen allein vom vorherigen Zustand abhängt. Der aktuelle Zustand gibt zu jedem Zeitpunkt genau ein Zeichen eines Alphabets aus, dass für alle Zustände identisch ist. Abgesehen davon, dass die Merkmalsvektoren für sich genommen keine sequentiellen Daten sind, in denen sinnvollerweise eine Reihenfolge der Erzeugung festgelegt werden kann, ist die Grundvoraussetzung für den Einsatz in dieser Arbeit, nämlich die Möglichkeit der Modellierung von Merkmalen unterschiedlicher Wertebereiche, nicht gegeben.

Hybride HMM/BN-Modelle. In Abschnitt 5.5 wurden die TFBS-BN als Ausgabeverteilungen von HMM-Zuständen eingesetzt. Diese in der Bioinformatik bis zum heutigen Zeitpunkt neuartige und unveröffentlichte Technik wurde im Jahre 2006 bereits für die Spracherkennung verwendet [Mar06]. Im Zusammenhang mit der Erkennung von TFBS-Modulen bietet dieser Ansatz die Möglichkeit, die Vorteile von HMM bei der Verarbeitung von sequentiellen Daten auch auf mehrdimensionale Wertebereiche zu übertragen, wobei dank der Modellierung der Ausgabeverteilungen durch Bayessche Netze die Anzahl der Verteilungsparameter vergleichsweise niedrig gehalten wird. Ein weiterer Vorteil gegenüber dem Frith'schen HMM, welches PWM-Modelle für einzelne TFBS verwendet, ist, dass TFBS sich gegenseitig überlappen können, da TFBS-BN jeweils nur eine Position der Sequenz erzeugen.

Die hybriden HMM/BN erfordern weitere Forschungstätigkeit. Erste Experimente zeigen nur sehr marginale Verbesserungen gegenüber einem HMM mit PWM-Modellen. Bei der Simulation eines Frith'schen HMM durch ein hybrides HMM/BN stellten sich sogar leicht höhere Klassifikationsfehlerraten ein. Zwei Erklärungsmöglichkeiten sind denkbar. Zum Einen besteht beim Operieren auf hochgradig multivariaten sequentiellen Daten ein Ungleichgewicht zwischen dem durchschnittlichen Werteniveau individueller Ausgabewahrscheinlichkeiten, die sehr klein sein können, und Übergangswahrscheinlichkeiten. Dieses Phänomen scheint trotz der Berechnung im logarithmischen Raum einen Einfluss auf die dynamischen Programmieralgorithmen zu haben, die zum Dekodieren eingesetzt werden. So traten in MAP-Zustandsfolgen häufiger *verbotene* Zustandsfolgen auf. Zum Anderen wurden die verwendeten TFBS-BN hinsichtlich der Anforderung optimiert, zwei Klassen zu unterscheiden. In den hybriden HMM/BN ist in jedem Zustand nur noch eine Klasse von Interesse, die möglichst adäquat beschrieben sein soll. Möglicherweise erfordert das eine andere Qualitätsfunktion bei dem verwendeten Merkmalsauswahlverfahren.

Ausblick. Da vor allem aus Gründen der Offenheit gegenüber verschiedenen Wertebereichen die Wahl auf Bayessche Netze zur Modellierung von regulatorischen Sequenzen gefallen ist, bietet es sich an, in Zukunft weitere Merkmalsklassen zu entwickeln, die zusätzliche Freiheitsgrade für die Beschreibung von TFBS bieten. Insbesondere nichtsequenzbasierte Merkmalsklassen stellen eine interessante Erweiterungsmöglichkeit dar. Ein Nachteil der bisherigen Modelle ist es, dass ausschließlich diskrete Merkmale verarbeitet werden können. Merkmale mit eigentlich kontinuierlichem Wertebereich müssen derzeit diskretisiert werden, wobei sicher Information verloren geht. Bayessche Netze mit kontinuierlichen Variablen existieren, sind jedoch etwas sperriger in ihrer Anwendung. Der Nutzen einer Erweiterung der Bayesschen Netze um diesen Aspekt wäre eine interessante Fortführung dieser Arbeit.

Schließlich müsste in Zukunft über Möglichkeiten nachgedacht werden, TFBS-BN-Modelle auch aus nicht-etikettierten Stichproben zu lernen, beispielsweise durch Anwendung des EM-Prinzips. Eine Schwierigkeit hierbei ist, dass gleichzeitig eine diskriminierende Merkmalsmenge gefunden werden muss, und anhand dieser eine optimale Aufteilung der unüberwachten Daten mittels EM-Algorithmus gefunden werden muss. Vorstellbar

wäre ein alternierender Prozess, bei dem nach jedem EM-Schritt gemäß SFFS ein Merkmal hinzugefügt wird und erneut der EM-Algorithmus ausgeführt wird. Neben Komplexitätsproblemen eines solchen Vorgehens stellt sich die Frage, ob der EM-Algorithmus in diesem Fall in jeder Iteration eine Verbesserung erzielen kann.

Kapitel 6

Modellierung von TFBS-Module mit Hilfe dynamische a priori Modellwahrscheinlichkeiten

In der biologischen Einführung in Kapitel 2 auf Seite 19 wurde beschrieben, dass Transkriptionsfaktoren häufig im Verbund mit weiteren Transkriptionsfaktoren auf der DNA binden und gemeinsam eine biologische Funktion erfüllen. Das bedingt häufig, dass die TFBS dieser Faktoren in enger Nachbarschaft zueinander liegen. TFBS sind deshalb häufig nicht gleichförmig im Bereich der regulativen Sequenzen verteilt, sondern in Gruppen. Diese Gruppen bilden funktionale Einheiten, die einen spezifischen Teil des Expressionsverhaltens eines Gens bestimmen.

Die Häufungstendenzen von TFBS sind ein beliebter Ansatzpunkt, um die zu große Zahl von TFBS-Vorhersagen bei isolierter Suche, beispielsweise mit PWM-Modellen, zu reduzieren. Die wichtigsten Vertreter dieser Verfahren wurden in Abschnitt 3.3 vorgestellt. Diese Verfahren haben gemeinsam, dass sie per se Häufungen von TFBS suchen, ohne zu untersuchen, ob eine bestimmte Häufung funktionell sinnvoll ist. Einerseits könnten zwei benachbarte TFBS-Vorhersagen zu Transkriptionsfaktoren gehören, die gar nicht kooperieren. Andererseits zeigen viele TFBS keine Tendenzen, in TFBS-Modulen aufzutreten [Mur04]. Die vorgestellten Verfahren könnten dabei scheitern, eine korrekterweise isolierte TFBS zu erkennen.

Darüber hinaus kommt es vor, dass die TFBS kooperierender Faktoren in einer bestimmten relativen Lage und Orientierung vorkommen müssen, damit beide Faktoren binden und wirken können. Eine Häufung alleine wäre dann kein Hinweis für die Richtigkeit der TFBS-Vorhersagen. Ein einfaches Beispiel hierfür ist der Transkriptionsfaktor *Sp1*, dessen Bindungsstellen meist einige hundert Basenpaare oberhalb einer TATA-Box zu finden sind, jedoch äußerst selten unterhalb der TATA-Box. Die TATA-Box selbst liegt in der Mehrheit der Fälle auf dem Sinnstrang. Ein weiteres Beispiel sind die TFBS des Transkriptionsfaktors *CTF/NFI*, die aus zwei Teilen bestehen, die durch einen in engen Grenzen variablen Zwischenbereich voneinander getrennt sind [Rou00]. Yu et al. hat in [Yu06] Paare kooperierender Transkriptionsfaktoren untersucht und konnte zeigen, dass die TFBS von mehr als der Hälfte dieser Paare eine bevorzugte Orientierung und einen bevorzugten Abstand zueinander haben.

In diesem Kapitel wird ein TFBS-Vorhersagesystem entworfen, dass die Treffer von einzelnen stochastischen Sequenzmodellen mit einer Auswertung der Stimmigkeit dieser

Treffer mit den Umgebungen bzw. Kontexte dieser mutmaßlichen Treffer kombiniert. Dabei ist es möglich, beliebige räumliche Abhängigkeiten zwischen den TFBS-Vorhersagen zu definieren, die erfüllt sein müssen, damit diese Vorhersagen tatsächlich biologisch relevant sind. Neben räumlichen Abhängigkeiten können Sequenzannotationen ausgewertet werden. Die gesamte Kontextinformation wird in einem speziell konstruierten Bayesschen Netz ausgewertet, dass eine dynamisch angepasste a priori Modellverteilung ausgibt, die als Filter für die Sequenzmodelltreffer dient.

Das Kapitel ist wie folgt aufgebaut: In Abschnitt 6.1 wird das gesamte Vorhersagesystem entworfen. Abschnitt 6.2 beschreibt die Versuche, die zu Validierungszwecken unternommen wurden. Abschnitt 6.3 diskutiert die Ergebnisse des Kapitels und gibt einen Ausblick für mögliche Erweiterungen.

6.1 A posteriori TFBS-Vorhersagen

In diesem Abschnitt soll eine a posteriori Verteilung über einer Menge $\boldsymbol{\mathcal{C}} = \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$ von K verschiedenen stochastischen Sequenzmodellen hergeleitet werden. Ein *Sequenzmodell* $\mathcal{C}_\kappa$ ist eine Wahrscheinlichkeitsverteilung über Sequenzen einer gegebenen Länge W über einem Alphabet Σ. Ohne Beschränkung der Allgemeinheit wird die Länge W für alle Sequenzmodelle als konstant angenommen. Handelt es sich um DNA-Sequenzen, wären PWM-Modelle der Länge W beispielsweise solche Sequenzmodelle, wenn die Gewichte der PWM-Spalten echte Wahrscheinlichkeiten sind. Ein anderes einfaches Sequenzmodell, dass direkt auf DNA-Sequenzen arbeiten kann, sind Markovketten. Handelt es sich hingegen um Sequenzen von Merkmalsvektoren, wie in Kapitel 5, könnten die dort entwickelten BN-Klassifikatoren eingesetzt werden, indem über die Klassenvariable marginalisiert wird. Da es in diesem Kapitel nicht um die Modellierung einzelner TFBS geht, sondern vielmehr um die Modellierung ihrer Umgebung, werden in den Versuchen in Abschnitt 6.2 PWM-Modelle eingesetzt. Aus diesem Grund wird auch hier die für DNA-Sequenzen vereinbarte Notation $\boldsymbol{s} = s_1 \ldots s_L$ für eine Sequenz der Länge L verwendet. Für eine solche Sequenz sei $\boldsymbol{s}_i = s_1 \ldots s_{i+W-1}$ die Teilsequenz der Länge W, die an Position i beginnt.

6.1.1 Sequenzmodellwahrscheinlichkeiten vs. a posteriori Modellwahrscheinlichkeit

Ein Sequenzmodell $\mathcal{C}_\kappa$ weist jeder Teilsequenz $\boldsymbol{s}_i$ die Wahrscheinlichkeit $P(\boldsymbol{s}_i \,|\, \mathcal{C}_\kappa)$ zu, von dem Sequenzmodell erzeugt worden zu sein. Umso größer $P(\boldsymbol{s}_i \,|\, \mathcal{C}_\kappa)$ ist, desto ähnlicher ist die Sequenz denen, mit denen $\mathcal{C}_\kappa$ trainiert wurde.

Obwohl diese Wahrscheinlichkeit mit der Frage zusammenhängt, wie wahrscheinlich eine Teilsequenz $\boldsymbol{s}_i$ ein Treffer für $\mathcal{C}_\kappa$ ist, ist diese Information nicht direkt abrufbar. Sie ermöglicht keine Vergleichbarkeit mit der Wahrscheinlichkeit $P(\boldsymbol{s}_i \,|\, \mathcal{C}_{\kappa'})$ eines alternativen Modells, insbesondere in dem allgemeinen Fall, dass die Modellierungslänge W nicht

konstant ist. Eigentlich interessant ist die a posteriori Wahrscheinlichkeit $P(\mathcal{C}_\kappa \mid \boldsymbol{s}_i)$, denn sie drückt eher die Möglichkeit aus, dass an Position i ein Treffer für $\mathcal{C}_\kappa$ vorliegt. Diese a posteriori Wahrscheinlichkeit kann über das Bayes-Theorem berechnet werden:

$$P(\mathcal{C}_\kappa \mid \boldsymbol{s}_i) \quad = \quad \frac{P(\boldsymbol{s}_i \mid \mathcal{C}_\kappa) P(\mathcal{C}_\kappa)}{\sum_{\kappa'=1}^{K} P(\boldsymbol{s}_i \mid \mathcal{C}_{\kappa'}) P(\mathcal{C}_{\kappa'})} \tag{6.1}$$

Dabei wird angenommen, dass jede Teilsequenz $\boldsymbol{s}_i$ für $i = 1 \ldots L$ einem der K Modelle zugeordnet werden kann. Diese Annahme ist gerechtfertigt, da eines der Modelle ein Hintergrundmodell sein muss, dass beliebige, nicht-funktionale Sequenzen modelliert.

Um die a posteriori Wahrscheinlichkeiten zu berechnen, werden die a priori Wahrscheinlichkeiten $P(\mathcal{C}_\kappa)$ benötigt. Diese drücken die Erwartung aus, auf einen Treffer für Modell $\mathcal{C}_\kappa$ zu stoßen, ohne die Sequenz schon beobachtet zu haben. Selbstverständlich müssen sich die a priori Wahrscheinlichkeiten aller Modelle zu 1 summieren:

$$\sum_{\kappa=1}^{K} P(\mathcal{C}_\kappa) = 1. \tag{6.2}$$

Diese Verteilung heißt im Weiteren kurz *a priori Modellverteilung* der Modelle in $\boldsymbol{\mathcal{C}}$.

Log-odds-Bewertungen. Als kurzer Einschub soll gezeigt werden, dass das soeben Entworfene eine Verallgemeinerung der schon bekannten *log-odds*-Bewertungen ist. Dazu sei angenommen, dass es nunmehr nur zwei Modelle $\mathcal{C}_{TFBS}$ und $\mathcal{C}_{\neg TFBS}$ für TFBS-Sequenzen und beliebige andere Sequenzen gibt. In diesem Fall kann die a posteriori Wahrscheinlichkeit in Form der logistischen Funktion

$$P(\mathcal{C} \mid \boldsymbol{s}_i) = \frac{e^x}{1 - e^x} \tag{6.3}$$

geschrieben werden, wobei gilt:

$$x = \log \left(\frac{P(\boldsymbol{s}_i \mid \mathcal{C}_{TFBS})}{P(\boldsymbol{s}_i \mid \mathcal{C}_{\neg TFBS})} \right) + \log \left(\frac{P(\mathcal{C}_{TFBS})}{P(\mathcal{C}_{\neg TFBS})} \right). \tag{6.4}$$

Der erste Summand in Gleichung 6.4 entspricht den *log-odds*-Bewertungen bei PWM-Modellen. Der zweite Summand ist ein konstanter Wert [Dur98], der in der Regel weggelassen wird, da meist keine besondere Information über die a priori Wahrscheinlichkeiten der Modelle vorliegt. Die *log-odds*-Bewertungen werden positionsweise und losgelöst von ihrem probabilistischen Hintergrund häufig als Gewichte in PWM-Modellen verwendet [Sto00]. Die logistische Funktion wird häufig verwendet, um Bewertungen zu konvertieren, die aus Summen von Wahrscheinlichkeiten gebildet wurden [Dur98].

6.1.2 Dynamische a priori Modellwahrscheinlichkeiten

Seien nun wieder K verschiedene Sequenzmodelle gegeben. Die grundsätzliche Idee des in diesem Kapitel vorgestellten Modellierungsansatzes besteht darin, die a priori Modellverteilung als Filtereinheit einzusetzen, anstatt sie komplett zu ignorieren. Das bedeutet, dass nicht nur die Sequenzmodellwahrscheinlichkeit $P(\boldsymbol{s}_i \,|\, \mathcal{C}_\kappa)$ zu der Entscheidung beitragen soll, ob an Position i ein Treffer für $\mathcal{C}_\kappa$ ist, sondern auch die a priori Wahrscheinlichkeit $P(\mathcal{C}_\kappa)$. Diese drückt die sequenzunabhängige Erwartung aus, dass an Position i Modell $\mathcal{C}_\kappa$ zutreffend ist. Da konstante a priori Wahrscheinlichkeiten keinen Fortschritt bieten würden, sollen sie dynamisch, in Abhängigkeit des an Position i vorherrschenden *Kontextes* ξ_i, angepasst werden. Falls der Kontext ξ_i Hinweise dafür enthält, dass an Position i Modell $\mathcal{C}_\kappa$ zutreffend ist, so wird an Position i die a priori Wahrscheinlichkeit für $\mathcal{C}_\kappa$ erhöht. Dadurch werden Treffer für $\mathcal{C}_\kappa$, bezogen auf die Sequenzmodellwahrscheinlichkeit, verstärkt. Wenn hingegen ξ_i gegen Modell $\mathcal{C}_\kappa$ spricht, drückt sich dies in einer verkleinerten a priori Wahrscheinlichkeit aus. Ein potentieller Treffer des Sequenzmodells würde bestraft werden.

Die a priori Modellwahrscheinlichkeiten unterliegen also dem Einfluss des Kontextes ξ_i. Unter der Annahme, dass die Erzeugung der Teilsequenz $\boldsymbol{s}_i$ durch Modell $\mathcal{C}_\kappa$ unabhängig von dem Kontext ξ_i ist, können durch Anwendung des Bayes-Theorems die Sequenzmodellwahrscheinlichkeiten dem neuen Filter unterworfen werden:

$$P(\mathcal{C}_\kappa \,|\, \boldsymbol{s}_i,\, \xi_i) = \frac{P(\boldsymbol{s}_i \,|\, \mathcal{C}_\kappa,\, \xi_i) P(\mathcal{C}_\kappa \,|\, \xi_i)}{\sum_{\kappa`=1}^{K} P(\boldsymbol{s}_i \,|\, \mathcal{C}_{\kappa`},\, \xi_i) P(\mathcal{C}_{\kappa`} \,|\, \xi_i)} \tag{6.5}$$

$$= \frac{P(\boldsymbol{s}_i \,|\, \mathcal{C}_\kappa) P(\mathcal{C}_\kappa \,|\, \xi_i)}{\sum_{\kappa`=1}^{K} P(\boldsymbol{s}_i \,|\, \mathcal{C}_{\kappa`}) P(\mathcal{C}_{\kappa`} \,|\, \xi_i)}. \tag{6.6}$$

Bevor damit fortgefahren wird, die Gestalt des Kontextes ξ_i zu erläutern, sollten noch ein paar Worte zu der Verwendung der Bezeichnungen *a priori* und *a posteriori* in diesem Zusammenhang gesagt werden. Natürlich erscheint es seltsam, die Wahrscheinlichkeiten $P(\mathcal{C}_\kappa \,|\, \xi_i)$ als a priori zu bezeichnen, obwohl sie einen Bedingungsteil haben. Der Begriff wird jedoch in Bezug auf die Sequenzmodelle gebraucht, die einfache, lokal abgeschlossene Sequenzen der Länge W modellieren, ohne die Umgebung dieser Sequenzen zu berücksichtigen. Selbst die aktuelle Position i ist den Sequenzmodellen unbekannt. Das rechtfertigt zudem die Annahme, dass die Erzeugung von Teilsequenzen durch Sequenzmodelle unabhängig von Position und Kontext ist.

Der Kontext ξ_i ist nur ein Platzhalter, um das Prinzip der dynamischen a priori Verteilungen zu erläutern. Der folgende Abschnitt wird sich damit beschäftigen, was unter dem Begriff *Kontext* zu verstehen ist. Anschließend wird das System vorgestellt, dass automatisch zu jeder Sequenzposition eine a priori Verteilung durch Auswertung des Kontextes erzeugt.

6.1.3 Der Kontext einer Sequenzposition

Der Kontext einer Sequenzposition bedeutet allgemein jegliches Wissen über die Umgebung, dass von Bedeutung ist, um zu entscheiden, welches Sequenzmodell an Position i am Wahrscheinlichsten ist. In diesem Kapitel sind das Informationen über mögliche Treffer anderer Sequenzmodelle in der Nachbarschaft oder beliebige andere Sequenzsignale. Das schließt auch textuelle Annotationen der Sequenz ein[1]. Ein Beispiel ist die Markierung eines experimentell validierten Transkriptionsstartpunktes. Diese Information könnte Aufschluss darüber geben, wo sich die aktuelle Suche gerade befindet, eher im Kernpromotorbereich oder in einem möglichen *Enhancer*. Ein weiteres Beispiel wäre eine Annotation einer Gewebespezifität der untersuchten Sequenz [2].

Damit, abgesehen von einer hohen Sequenzähnlichkeit, ein echter Treffer für Modell $\mathcal{C}_\kappa$ an Position i vorliegt, müssen eine bestimmte Menge von Bedingungen erfüllt sein. Der Kontext ξ_i gibt Auskunft über die Erfülltheit oder Nichterfülltheit dieser Bedingungen.

Formalisierung. Es wird gezeigt werden, dass die Bedingungen, die für einen gültigen Treffer von Modell $\mathcal{C}_\kappa$ gelten müssen, als logische Ausdrücke formalisiert werden können. Die Atome dieser logischen Ausdrücke sind spezielle Prädikate, die *Kontextprädikate*, welche die Erfülltheit elementarer Bedingungen überprüfen.

Es werden drei Kategorien von Kontextprädikaten unterschieden: 1.) Prädikate des Typs neighbour_match für Treffer in der Nachbarschaft einer Position, 2.) Prädikate neighbour_annotation für Annotationen der Sequenz in der Nachbarschaft einer Position und 3.) Prädikate annotation für positionsunabhängige Annotationen einer Sequenz.

DEFINITION 6.1: *Sei* $\boldsymbol{s} = s_1 \ldots s_L$ *eine DNA-Sequenz der Länge* L *und* $\mathcal{A}(\boldsymbol{s}) = \bigcup_{i=1}^{L} \mathcal{A}_i(\boldsymbol{s})$ *eine Menge von Annotationen für* $\boldsymbol{s}$*, wobei* $\mathcal{A}_i(\boldsymbol{s})$ *alle Annotationen* A *enthält, die Position* i *der Sequenz betreffen. Seien weiterhin die* Parameter $left, right \in \mathbb{Z}$*,* $\rho \in \mathbb{R}$ *sowie* $strand \in \{sens, anti, any, same, opposite\}$ *gegeben und* $\mathcal{C}$ *ein Sequenzmodell.*

Die Hilfsfunktion best_score$(i \,|\, left, right, strand, \mathcal{C})$ *berechne die höchste Sequenzwahrscheinlichkeit für Modell* $\mathcal{C}$ *in dem Sequenzintervall* $s_{i+left} \ldots s_{i+right}$ *auf dem angegebenen DNA-Strang (bzw. auf beiden Strängen). Dann gelten die folgenden Vereinbarungen:*

1. *Ein* **Kontextprädikat** neighbour_match$_{(left,right,strand,\rho,\mathcal{C})}$ *ist eine Funktion*

$$\text{neighbour\_match}_{(left,right,strand,\rho,\mathcal{C})} : \mathbb{N} \mapsto \mathbb{B} \tag{6.7}$$

[1] z.B. bei DNA-Sequenzen im EMBL-Format

[2] das betreffende Gen wird in einem bestimmten Gewebe expremiert

mit folgender Abbildungscharakteristik:

$$\mathsf{neighbour\_match}_{(left,right,strand,\rho,\mathcal{C})}(i) = \begin{cases} \mathsf{true} & : \quad \mathsf{best\_score}(i \mid \cdots) > \rho \\ \mathsf{false} & : \quad \text{sonst} \end{cases} \tag{6.8}$$

2. *Ein* **Kontextprädikat** $\mathsf{neighbour\_annotation}_{(left,\,right,\,A)}$ *für eine Annotation A ist eine Funktion*

$$\mathsf{neighbour\_annotation}_{(left,\,right,\,A)} : \mathbb{N} \mapsto \mathbb{B} \tag{6.9}$$

mit folgender Abbildungscharakteristik:

$$\mathsf{neighbour\_annotation}_{(left,\,right,\,A)}(i) = \begin{cases} \mathsf{true} & : \quad \exists_{j=i+left}^{i+right} A \in \mathcal{A}_j(\boldsymbol{s}) \\ \mathsf{false} & : \quad \text{sonst} \end{cases} \tag{6.10}$$

3. *Ein* **Kontextprädikat** $\mathsf{annotation}_{(A)}$ *für eine Annotation ist eine Funktion*

$$\mathsf{neighbour\_annotation}_{(A)} : \mathbb{N} \mapsto \mathbb{B} \tag{6.11}$$

mit folgender Abbildungscharakteristik

$$\mathsf{neighbour\_annotation}_{(left,\,right,\,A)}(i) = \begin{cases} \mathsf{true} & : \quad \exists_{j=1}^{L} A \in \mathcal{A}_j(\boldsymbol{s}) \\ \mathsf{false} & : \quad \text{sonst} \end{cases} \tag{6.12}$$

Das in diesem Kapitel bedeutungsvollste Prädikat, neighbour_match, bedarf weiterer Bemerkungen. Bisher wurde, beispielsweise in Gleichung 6.5, davon ausgegangen, dass für jede Sequenzposition eine a posteriori Wahrscheinlichkeit für jedes Modell berechnet wird. Die obige Definition sieht nun aber eine Unterscheidung der beiden DNA-Stränge vor. Dazu sei festgestellt, dass die Suche in einer Sequenz selbstverständlich auf beiden DNA-Strängen erfolgt. Der Ergebnisse der Suchen eines Strangs können als Fühlerwerte bei der Analyse des anderen Strangs verwendet werden.

Die logischen Ausdrücke zur Beschreibung der Bedingungen für eine erfolgreiche Bindung sollen im folgenden *Kontextausdrücke* heißen und sind wie folgt aufgebaut:

DEFINITION 6.2: *Ein* **Kontextausdruck** ξ *hat eine der folgenden Formen:*

- *die logischen Konstanten* true *oder* false
- *ein Kontextprädikat* neighbour_match, neighbour_annotation *oder* annotation
- *die Negation* $\neg\xi$‘, *eines Kontextausdruck* ξ‘
- *die Konjunktion* $\xi_1 \wedge \xi_2$ *von Kontextausdrücken* ξ_1 *und* ξ_2
- *die Disjunktion* $\xi_1 \vee \xi_2$ *von Kontextausdrücken* ξ_1 *und* ξ_2

Die Kontextprädikate eines Kontextausdruckes untersuchen elementare Bedingungen, die für ein Zutreffen eines Modells $\mathcal{C}_\kappa$ erfüllt sein müssen (bzw. im Falle der Negation nicht erfüllt sein dürfen). In dem nachfolgend vorgestellten Modell zur Ableitung einer a priori Modellverteilung aus dem Kontext wird die Erfülltheit eines Prädikats von so genannten *Fühlervariablen* gemessen.

6.1.4 MVBN: Modelle zur Auswertung des Kontextes

Das in diesem Kapitel entworfene System wertet für jede Position i den Kontext ξ_i aus und erzeugt daraus dynamisch eine a priori Verteilung über alle betrachteten Modelle $\mathcal{C}_\kappa$. Einerseits müssen also strikte logische Ausdrücke ausgewertet werden, andererseits die Ergebnisse dieser Auswertung in eine weiche probabilistische Steuerung der Verteilung umgesetzt werden. Diese beiden Aspekte werden in einem speziell konstruierten Bayesschen Netz gelöst. Dessen Struktur und Parameter werden in diesem Fall nicht trainiert, sondern gemäß den kausalen Erfordernissen, die sich aus den definierten Ausdrücken und ihren Prädikaten ergeben, gesetzt.

Gemäß dem Aufbau dieser Bayesschen Netze handelt es sich um spezielle BN-Klassifikatoren. Als Bezeichnung für diese Netze wird im folgenden *MVBN* (für *Modellverteilungs-BN*) verwendet. Zunächst wird nun ein MVBN formal definiert, und der definierte Aufbau anschließend diskutiert.

DEFINITION 6.3: *Seien $\mathcal{C}_1, \ldots, \mathcal{C}_K$ stochastische Sequenzmodelle. Ein* **MVBN** *ist ein BN-Klassifikator mit Variablenmenge $\{C\} \cup \boldsymbol{T} \cup \boldsymbol{E}$ und folgenden Eigenschaften:*

1. *C ist die Klassenvariable mit Wertemenge $D_C = \{1, \ldots, K\}$*
2. *$\boldsymbol{T} = \{T_1, \ldots, T_N\}$ ist eine Menge von* **Fühlervariablen**. *Ihre einzige Elternvariable ist die Klassenvariable C. Die Wertemenge aller Fühlervariablen ist $\mathbb{B}$.*
3. *$\boldsymbol{E} = \{E_1, \ldots, E_K\}$ ist eine Menge von* Ausdrucksvariablen. *Jede Ausdrucksvariable E_κ hat als Elternvariablen C sowie eine Auswahl $\boldsymbol{\Pi}_{E_\kappa}^{(\boldsymbol{T})}$ von Fühlervariablen. Die Wertemenge aller Ausdrucksvariablen ist $\mathbb{B}$.*

Die Klassenvariable C eines MVBN stellt die Ausgabevariable dar, denn die gewünschte a priori Verteilung wird über probabilistische Anfragen der Form

$$P(C = \kappa \,|\, \boldsymbol{E} = \boldsymbol{e}, \boldsymbol{T} = \boldsymbol{t}) \tag{6.13}$$

bestimmt. Die Wahrscheinlichkeitsparameter $\boldsymbol{\theta}_C$ sind eine *echte* a priori Verteilung über den Modellen $\mathcal{C}_\kappa$. Mit Hilfe dieser Wahrscheinlichkeiten kann Wissen darüber kodiert werden, dass für einige Modelle generell weniger oder mehr Treffer zu erwarten sind, als für die anderen Modelle[3].

Eine Fühlervariable T_n modelliert ein im vorherigen Abschnitt definiertes Kontextprädikat. Bei diesen Fühlern handelt es sich rein formal um Merkmale im Sinne von Unterabschnitt 4.3.1 auf Seite 72. Am ehesten ähneln sie den $\mathcal{M}_{KOOP}$-Merkmalen des TFBS-BN-Ansatzes, dass in Kapitel 5 vorgestellt wurde. Da C die einzige Elternvariable einer Fühlervariable T_n ist,

$$(\theta_{\mathsf{true}\,|\,\kappa}, \theta_{\mathsf{false}\,|\,\kappa}). \tag{6.14}$$

[3] Es gibt Transkriptionsfaktoren, die nur einige wenige Gene regulieren, andere wirken auf einem breiten Spektrum von Genen

Auch diese Wahrscheinlichkeiten werden zur Ableitung der a priori Modellverteilung aus einem Kontext genutzt. Dabei werden für einen Fühler T_n *unterstützende* Modelle und *verhindernde* Sequenzmodelle unterschieden. Für eine unterstützendes Modell $\mathcal{C}_\kappa$ wird stark angenommen, dass das durch T_n repräsentierte Kontextprädikat erfüllt ist. Deshalb wird für unterstützende Modelle $\mathcal{C}_\kappa$ die Wahrscheinlichkeit $\theta_{\mathsf{true}\,|\,\kappa}$ auf einen hohen, konstanten Wert gesetzt, $\theta_{\mathsf{false}\,|\,\kappa} = 1 - \theta_{\mathsf{true}\,|\,\kappa}$ ist also entsprechend klein. Umgekehrt verhält es sich, wenn ein $\mathcal{C}_\kappa$ ein verhinderndes Modell ist. Dann wird eher davon ausgegangen, dass Fühler T_n nicht fündig wird ($T_n = \mathsf{false}$) und die Wahrscheinlichkeit $\theta_{\mathsf{false}\,|\,\kappa}$ erhält den hohen Wert.

Als Anwendungsbeispiel sei angenommen, dass ein Fühler T_n nach einer Sequenzannotation sucht, die aussagt, dass das betreffende Gen hauptsächlich in Leberzellen expremiert wird. Sei außerdem $\mathcal{C}_\kappa$ ein Sequenzmodell für einen Transkriptionsfaktor, der die Transkription von Lebergenen reguliert. Bestünde der Glaube, dass an der aktuellen Position ein Treffer für $\mathcal{C}_\kappa$ vorliegt, ausgedrückt in $C = \kappa$, dann besteht gleichfalls ein großer Glaube darin, dass es sich um eine leberspezifische Sequenz handelt. Die Wahrscheinlichkeit dafür, eine entsprechende Sequenzannotation zu finden, wäre dann höher als für andere Modelle, die TFBS modellieren, die in anderen Geweben vorherrschend sind.

Die genannte hohe Wahrscheinlichkeit ist allgemein eine Konstante $p_{\sim 1}$, die eine Art probabilistisches Version des Wahrheitswerts true repräsentiert. Wie anschließend leicht ersichtlich wird, beeinflusst die Wahl dieser Konstante den Einfluss des a priori Modells auf die Vorhersage. Im Falle eines Wertes, der nahezu bei 1 liegt, wird der a priori Verteilung ein großer Einfluss eingeräumt. Selbst perfekte Sequenzmodelltreffer würden im Falle eines ungünstigen Kontextes hart bestraft werden. Das andere Extrem wäre eine Konstante von 0.5. In diesem Fall wäre der Einfluss des a priori Modells komplett ausgeschaltet und nur die Sequenzwahrscheinlichkeiten würden einen Beitrag zur Vorhersage liefern.

Eine Ausnahme bilden die Fühlervariablen für Prädikate $\mathsf{neighbour\_match}$ im Falle unterstützender Modelle. Hier werden die Wahrscheinlichkeiten $\theta_{\mathsf{true}\,|\,\kappa}$ und $\theta_{\mathsf{false}\,|\,\kappa}$ an jeder Position dynamisch gesetzt. Und zwar ist $\theta_{\mathsf{true}\,|\,\kappa}$ dann proportional zu der Bewertung des besten Treffers der Hilfsfunktion $\mathsf{best\_score}$ in den durch das Prädikat festgelegten Grenzen. Werden also nur schwache Treffer des Fühlers gefunden, sinkt auch die Wahrscheinlichkeit $\theta_{\mathsf{true}\,|\,\kappa}$ dafür, dass es einen solchen Treffer wirklich gibt.

Die Ausdrucksvariablen E_κ sind probabilistische Versionen der Kontextausdrücke der Modelle $\mathcal{C}_\kappa$. Im Grunde realisiert E_κ ein logisches Gatter für den Kontextausdruck für Modell $\mathcal{C}_\kappa$, wobei die Eingänge des logischen Gatters die Fühlervariablen aus $\mathbf{\Pi}_{E_\kappa}^{(\boldsymbol{T})}$ sind, und die Funktionswerte über die gezielte Setzung der bedingten Wahrscheinlichkeiten von E_κ gesteuert werden. Ein Beispiel für ein solches logisches Gatter ist die Wahrscheinlichkeitstabelle von E_1 in Abbildung Abbildung 6.1, auf dem eine Konjunktion zweier Fühlervariablen dargestellt ist.

Die Klassenvariable C ist ebenfalls Elternvariable von E_k. Dies ergibt sich aus den fol-

genden Überlegungen. Für jedes κ kodiert E_κ den Kontextausdruck für Modell $\mathcal{C}_\kappa$[4]. Wird nun aktuell untersucht, ob Modell $\mathcal{C}_\kappa$ vorliegt, dann soll auch nur E_k einen Einfluss besitzen. Da in einem Bayesschen Netz jedoch die anderen Ausdrucksvariablen nicht einfach ignoriert werden können, weiß jede Ausdrucksvariable über die Verbindung zu C, welches Modell $\mathcal{C}_\kappa$ gerade untersucht wird und kann ihren Einfluss selbstständig ausschalten, falls sie nicht zuständig ist. Das funktioniert folgendermaßen: Sei $C = \kappa$, also Modell $\mathcal{C}_\kappa$ ausgewählt. Der Teil der Wahrscheinlichkeitstabelle von E_κ, in dessen Bedingungsteil $C = \kappa$ ist, modelliert das geforderte logische Gatter wie in Abbildung 6.1 gezeigt. In allen anderen Ausdrucksvariablen $E_{\kappa'}$ ist der Teil der Tabelle, in dem $C = \kappa$ ist, nur mit Wahrscheinlichkeiten 0.5 aufgefüllt. Diese Ausdrucksvariablen liefern also einen konstanten, neutralen Beitrag von 0.5 zur gesamten a posteriori-Wahrscheinlichkeit. Die Ausdrucksvariable E_κ hingegen liefert eine sehr kleine Wahrscheinlichkeit, wenn der Kontextausdruck für $\mathcal{C}_\kappa$ nicht erfüllt ist, und eine sehr hohe, wenn dieser erfüllt ist. Sei $p_{\sim 1}$ eine Konstante $\gg 0.5$ eine Konstante für eine große Wahrscheinlichkeit. Dann gilt für eine konkrete Belegung $\boldsymbol{\pi}_{E_\kappa}^{(\boldsymbol{T})}$ der Fühlervariablen $\boldsymbol{\Pi}_{E_\kappa}^{(\boldsymbol{T})}$ für E_κ:

$$P(E_\kappa = \mathsf{true} \,|\, \boldsymbol{\pi}_{E_\kappa}^{(\boldsymbol{T})}) \quad = \quad \begin{cases} p_{\sim 1} & : \ \boldsymbol{\pi}_{E_\kappa}^{(\boldsymbol{T})} \text{ macht den Kontextausdruck wahr} \\ 1 - p_{\sim 1} & : \ \text{sonst} \end{cases} \tag{6.15}$$

Wie schon die Konstanten für die Fühlervariablenparameter beeinflusst die Wahl von $p_{\sim 1}$ den Einfluss des MVBN auf die a posteriori Wahrscheinlichkeiten. Als sinnvoller Wert hat sich $p_{\sim 1} = 0.75$ erwiesen.

Beim Probabilistischen Schließen bezüglich der Klassenvariable C werden alle E_κ auf den Beobachtungswert true gesetzt. Dies ist verständlich, da, wenn Modell $\mathcal{C}_\kappa$ getestet werden soll, gerade die *Erfülltheit* von E_κ von Wichtigkeit ist. Liegt diese nicht vor, so erhält der Wert $E_\kappa = \mathsf{true}$ eine niedrige Wahrscheinlichkeit. Das ist das gewünschte Verhalten, denn die niedrige Wahrscheinlichkeit bestraft über die resultierende niedrigere a priori Wahrscheinlichkeit des MVBN das Modell $\mathcal{C}_\kappa$.

Als Ergebnis dieses Abschnittes steht, dass die recht abstrakten a priori Modellwahrscheinlichkeiten $P(\mathcal{C}_\kappa \,|\, \xi_i)$ für eine Sequenzposition i nun durch probabilistische Anfragen an die Klassenvariable eines MVBN gestellt werden:

$$P(\mathcal{C}_\kappa \,|\, \xi_i) \equiv P(C = \kappa \,|\, \boldsymbol{E} = \mathbf{1}^K, \boldsymbol{\Pi}_{E_\kappa}^{(\boldsymbol{T})} = \boldsymbol{\pi}_{E_\kappa}^{(\boldsymbol{T})}(i)). \tag{6.16}$$

Abbildung 6.1 zeigt ein beispielhaftes MVBN für vier verschiedene Modelle.

6.1.5 Zusammenbau des Erkennungssystems

Nachdem der Aufbau und die Funktionsweise eines MVBN zur Ableitung positionsspezifischer a priori Modellverteilungen erläutert wurde, kann es als abgeschlossene Einheit,

[4] Eine andere Möglichkeit wäre es gewesen, die Kontextausdrücke aller Modelle zu vereinen, und in einer einzigen Ausdrucksvariable zu modellieren. Diese hätte jedoch sehr viele Elternvariablen besessen, und das Setzen der Wahrscheinlichkeiten gemäß des riesigen Kontextausdruckes wäre um einiges schwieriger gewesen.

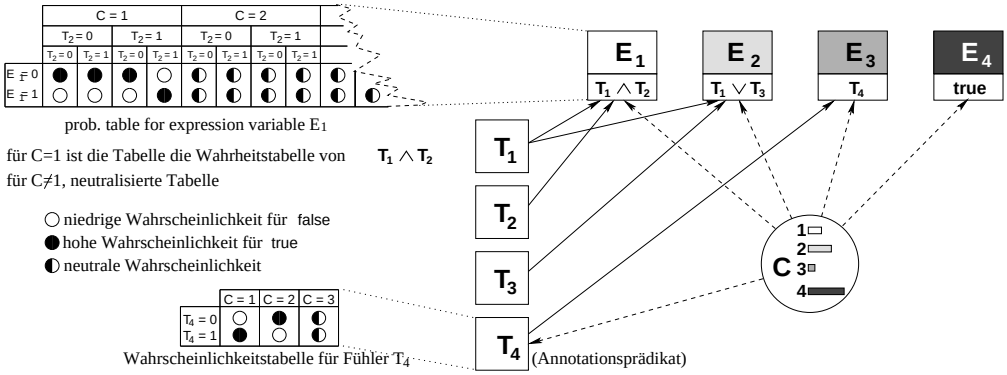


Abbildung 6.1: Ein MVBN für die Ableitung von a priori Modellverteilungen für vier Modelle $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$, $\mathcal{C}_4$. Es gibt vier Fühlervariablen, wovon eine ein annotation-Prädikat repräsentiert. Für jedes Modell M_κ gibt es eine AAusdrucksvariable E_κ. Die Ausdrucksvariable E_1 wird mit hoher Wahrscheinlichkeit true, wenn $t_1 \wedge t_2$ ebenfalls true ist.

bzw. als *black box* verwendet werden, um gemäß Gleichung 6.5 die a posteriori Wahrscheinlichkeiten dafür zu berechnen, dass Modell $\mathcal{C}_\kappa$ an Position i bei gegebener Sequenz und gegebenen Kontext zutreffend ist. Abbildung 6.2 zeigt schematisch den Ablauf der Suche nach TFBS-Modulen in einer DNA-Sequenz. Im ersten Schritt werden die Sequenzwahrscheinlichkeiten $P(\boldsymbol{s}_i \,|\, \mathcal{C}_\kappa)$ für jede Position i und unter Anwendung aller Modelle $\mathcal{C}_\kappa$ bestimmt. Die sich ergebende Matrix der Dimension[5] $2K \times L$ wird gemeinsam mit Annotationsinformationen der Sequenz eingesetzt, um eine Folge der Länge L von Wertevektoren für alle beteiligten Fühlervariablen zu erhalten. Jeder dieser Vektoren wird auf das MVBN angewendet, es entsteht eine Matrix von Modellwahrscheinlichkeiten für jedes Sequenzmodell an jeder Position. Im letzten Schritt wird die Matrix der Sequenzwahrscheinlichkeiten und die Matrix der a priori Modellwahrscheinlichkeiten spaltenweise, unter Anwendung von Gleichung 6.5 zu einer Matrix von a posteriori Wahrscheinlichkeiten $P(\mathcal{C}_\kappa \,|\, \boldsymbol{s}_i, \xi_i)$ verarbeitet. Diese dienen als Bewertung für die Klassifikation einer Position in eine der K Klassen. Als Entscheidungsregel dient trivialerweise

$$\delta(i) = \underset{\kappa}{\operatorname{argmax}}\, P(\mathcal{C}_\kappa \,|\, \boldsymbol{s}_i, \xi_i). \tag{6.17}$$

6.1.6 Wenn es an Vorwissen mangelt...

Um die MVBN-Modelle gewinnbringend einzusetzen, bedarf es Vorwissen seitens eines Anwenders bezüglich räumlicher Zusammenhänge zwischen den TFBS eines TFBS-Moduls bzw. bestimmter Eigenschaften der Sequenz. Häufig ist dieses Vorwissen nicht

[5] Es werden anders, als in der Abbildung angedeutet, beide DNA-Stränge durchsucht.

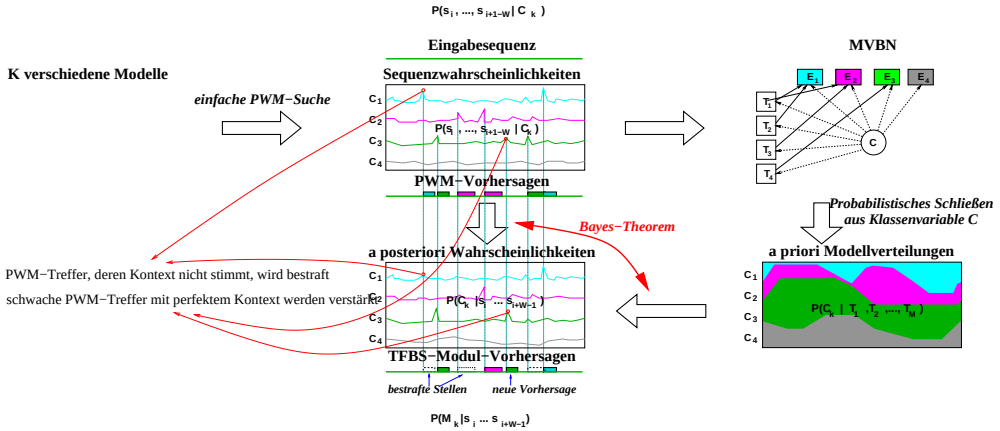


Abbildung 6.2: Schematische Darstellung des Suchprozesses, der aus drei Schritten besteht: 1.) Suche auf der Sequenz mit allen Sequenzmodellen 2.) Herstellung von Fühlervariablen-Wertevektoren und Anwendung dieser Vektoren auf ein MVBN 3.) Beide Teile unter Anwendung des Bayes-Theorems zu a posteriori-Wahrscheinlichkeiten für die Modelle an allen Sequenzpositionen verarbeiten.

vorhanden. In solchen Fällen bieten sich Lernverfahren an, welche logische Zusammenhänge aus einer Lernstichprobe ermitteln können. Diese Verfahren sind in der Literatur unter dem Begriff *association mining* zu finden. Ein bekanntes Verfahren ist der APRIORI-Algorithmus von Agrawal et al. [Agr94].

Lernverfahren für Kontextausdrücke. Zu Versuchszwecken wurde in dieser Arbeit eine einfache Suchmethode entwickelt, die hier kurz erläutert werden soll. Ziel ist es, Regeln für die Beschaffenheit des Kontextes für TFBS eines bestimmten Transkriptionsfaktors herzuleiten.

Dazu sei eine Positivstichprobe d^+ und eine Negativstichprobe d^- von DNA-Sequenzen gegeben. Jede positive Sequenz enthält eine bekannte TFBS des Typs $\mathcal{C}$. Ohne Beschränkung der Allgemeinheit seien alle Sequenzen gleich lang und die TFBS in der Positivstichprobe jeweils in mittlerer Position einer Sequenz.

Es sollen nun Überrepräsentationen von TFBS-Vorhersagen kooperierender Transkriptionsfaktoren detektiert werden. Aus diesen Überrepräsentationen werden dann zunächst neighbour_match-Prädikate entwickelt, die anschließend zu größeren Ausdrücken verknüpft werden. *Überrepräsentation* eines Modells $\mathcal{C}_i$ bedeutet, dass in der Positivmenge in dem jeweils untersuchten Intervall (z.B. oberhalb der TFBS) signifikant höhere Bewertungen erreicht werden als in der Negativmenge.

Seien also Sequenzmodelle $K_1, \ldots, K_M$ gegeben[6], jeweils für TFBS eines kooperierenden

[6] In Abgrenzung zu den K Modellen $\mathcal{C}_\kappa$, über die in diesem Kapitel eine Rolle spielen, werden hier die

Faktors von $\mathcal{C}$. Mit der Hilfsmethode best_score wird der beste Treffer für jedes Modell in bestimmten Sequenzstücken jeder Sequenz gesucht. Dabei werden die Sequenzintervalle

- oberhalb der TFBS
- unterhalb der TFBS
- ober- oder unterhalb der TFBS

und die Orientierungen

- Sinnstrang
- Antistrang
- beide Stränge
- der gleiche Strang wie die TFBS
- der gegenüberliegende Strang

unterschieden. Die Sequenzintervalle werden durch konstante Werte für $left = -250$ und $right = +250$ realisiert.

Insgesamt können über die verschiedenen best_score-Aufrufe $M \times 3 \times 3$ Kandidaten-neighbour_match-Prädikate definiert werden, die mit Hilfe der Stichproben getestet werden müssen. Die Kandidatenmenge stellt die Anfangssaat für die Konstruktion der Kontextausdrücke dar. Dazu wird jedem Prädikat ξ ein Qualitätswert $J(\xi)$ zugewiesen.

Die Qualitätsfunktion $J(\xi)$ basiert auf einem t-Test. Jedes Prädikat ξ (bzw. die dazugehörige best_score-Konfiguration) erzeugt für jede Sequenz einen Wert. Die Überrepräsentation wird dadurch gemessen, wie sehr sich der Mittelwert der maximalen Bewertung in den Positivsequenzen von denen der Negativen unterscheidet. Die beiden Mittelwerte werden via t-Test miteinander verglichen. für den sich ergebenden p-Wert p_ξ ist $J(\xi) = 1 - p_\xi$ der Qualitätswert von Prädikat ξ.

Die Liste aller Kandidatenprädikate wird aufsteigend bezüglich der Qualitätswerte sortiert. Die sortierte Liste wird verwendet, um die signifikantesten Prädikate zu komplexeren Kontextausdrücken über Konjunktion oder Disjunktion zu verknüpfen.

Für eine Konjunktion wird folgendermaßen vorgegangen: Seien ξ_1 und ξ_2 zwei Prädikate, und b_1 und b_2 die best_score-Ergebnisse auf einer bestimmten Sequenz (entweder in d^+ oder d^-). Der Konjunktion der beiden Prädikate entspricht es, das Minimum $\min(b_1, b_2)$ als Teststatistik zu verwenden. Wieder entstehen zwei Zahlenfolgen für d^+ und d^-, und wieder kann die statistische Signifikanz des neuen Ausdruckes berechnet werden. Der zusammengesetzte Ausdruck wird in die sortierte Liste eingefügt. Es wird darauf hingewiesen, dass nur zwei starke atomare Kontextprädikate in der Lage sind, eine signifikante Konjunktion zu bilden.

Bezeichner K_m verwendet.

Um zwei Prädikate ξ_1 und ξ_2 über eine Disjunktion zu verknüpfen, wird jeweils das Maximum $\max(b_1, b_2)$ für den statistischen Test verwendet. Hierbei kann es durchaus vorkommen, dass zwei zuvor eher schwache Prädikate eine signifikante Disjunktion ergeben.

Die Methode, neue Verknüpfungen aus bereits bestehenden Kontextausdrücken herzustellen, wird so lange fortgeführt, wie noch statistisch signifikante Kontextausdrücke entstehen. Die in der sortierten Liste oberen Ausdrücke sind gute Kandidaten eines Kontextausdrucks für das Modell $\mathcal{C}$ in einem MVBN.

Triviale MVBN-Konfigurationen. Die Modellierung von Vorwissen in MVBN ist ein Angebot, kein Zwang. Falls obige Methode keine stichhaltigen Regeln für das Bindungsverhalten eines bestimmten Transkriptionsfaktor finden, und kein Vorwissen darüber vorliegt, kann das MVBN auch in einer Weise konfiguriert werden, so dass jede Häufung von TFBS belohnt wird. Dazu wird für jedes Modell $\mathcal{C}_\kappa$ eine Fühlervariable T_κ eingerichtet, welche eine bestimmte Umgebung der aktuellen Position nach Treffern durchsucht. Die Ausdrucksvariablen E_κ sind jeweils Disjunktionen aller Fühlervariablen. In dieser Konfiguration entspricht das System einem fensterbasierten TFBS-Modul-Erkennungssystem.

Falls selbst eine Häufung von TFBS nicht erwartet werden kann, kann das MVBN auch völlig neutralisiert werden, indem alle Kontextausdrücke durch den einfachen Ausdruck `true` ersetzt werden. Dies entspricht einer einfachen Suche nach TFBS-Treffern mit den einzelnen Sequenzmodellen $\mathcal{C}_\kappa$

6.2 Ergebnisse

Um den hier vorgestellten Ansatz zur Modellierung von TFBS-Modulen zu analysieren, wurde es in verschiedenen Konfigurationen auf einen Datensatz von Promotorsequenzen aus Säugetiergenomen angewendet. Außerdem wurden künstliche Datensätze hergestellt, um das Verhalten des Systems in den verschiedenen Konfigurationen statistisch robuster zu untersuchen, als dies mit dem recht kleinen genomischen Datensatz möglich ist.

6.2.1 Genomische Daten

Bei Verwendung des *real-biologischen* Datensatz war das Ziel, Bindungsstellen für den Transkriptionsfaktor *HNF-4* in einer Menge von Promotoren korrekt vorherzusagen. HNF-4 (für englisch *Hepatocyte Nuclear Factor 4*) ist ein Transkriptionsfaktor, der an der Regulation von Leber-spezifischen Genen beteiligt ist. Im kleineren Umfang tritt er auch in Nieren sowie Dünn- und Dickdarm auf. Er ist evolutionär hochgradig konserviert, von Insekten bis Säugetieren. Seine bevorzugte TFBS besteht aus einer Wiederholung der Sequenz `AGGTCA`, unterbrochen von einigen wenigen Nukleotiden. Es ist bekannt, dass HNF-4 unter Anderen mit den Transkriptionsfaktoren *COUP-TF1*, *HNF-1*, *HNF-3*,

C/EBPα kooperiert sowie Homodimere ausbildet. Die öffentliche Version der Datenbank TRANSCOMPEL (siehe Unterabschnitt 3.1.4) enthält 17 TFBS-Paare, bestehend aus HNF-4 und einem dieser Faktoren. Eine Tendenz von HNF-4, in der Nähe von TFBS dieser Faktoren zu binden, wird auch aus TRANSFAC ersichtlich.

Alle Promotoren, die eine bekannte HNF4-Bindungssequenz enthalten, wurden aus diesen beiden Datenbanken über die referenzierten EMBL-Einträge gewonnen. Es handelt sich um 54 Sequenzen. Insgesamt 17 der 54 Sequenzen enthalten validierte kooperierende TFBS, bei den restlichen kann über die tatsächliche Funktion der HNF-4-Bindungsstelle nur gemutmaßt werden. Als Negativdatensatz wurden 54 beliebige Promotoren aus EPD heruntergeladen.

Das a priori Wissen über den Kontext einer HNF-4 Bindungssetelle wurde durch vier Fühlervariablen für jeden der möglichen kooperierenden Faktoren ausgedrückt. Der Kontextausdruck für HNF-4 war eine Disjunktion der vier Fühlervariablen.

Die Leistung des Systems wurde mit der einer einfachen Suche nach HNF-4 TFBS mittels eines normalen PWM-Modells verglichen. Jede der 2×54 Sequenzen wurde zunächst auf herkömmliche Weise mit einer PWM durchsucht, anschließend unter Anwendung des hier vorgestellten MVBN-Ansatzes. Dabei wurden Vorhersagen in den 54 Positivsequenzen als richtige Vorhersage gezählt, Vorhersagen in der Negativmenge galten als Falsch-positiv.

Es wurde eine ROC-Kurve gebildet, indem die a posteriori-Wahrscheinlichkeiten, welche die richtigen Treffer durch das jeweilige System erhalten haben, als Vorhersageschranke verwendet wurden, und für jede dieser Schranken die Anzahl der Falsch-Positiven bestimmt wurde. Sie ist in Abbildung 6.3 dargestellt und zeigt deutlich, wie die Verwertung des a priori Wissens die Erkennungsleistung verbessert. Für die in der Praxis bedeutsame Schranke, die 90% der TFBS richtig erkennt, lag die FP-Rate des hier vorgestellten Systems bei einem Zehntel der FP-Rate einer normalen PWM-Suche. Der Flächeninhalt unter der ROC-Kurve lag bei 0.981 im Vergleich zu 0.909 für die PWM-Suche. Wurde als Vorhersageschranke die jeweils wahrscheinlichste HNF-4 Bindungsstelle verwendet, schnitt die PWM-Suche etwas besser ab. Der Grund dafür ist, dass die HNF-4 Seite mit der besten PWM-Bewertung keinen günstigen Kontext besitzt und durch den a priori-Filter abgestraft wird.

6.2.2 Künstlicher Datensatz

Da die Anzahl verfügbarer regulatorischer Module, welche dieselben TFBS-Arten enthalten, sehr klein ist, ist es schwierig, die Leistung eines TFBS-Modul-Vorhersagesystems auf robuste Weise zu untersuchen. Aus diesem Grund wurde, wie im Folgenden beschrieben, ein künstlicher Datensatz entwickelt, der die beobachteten biologischen Verhältnisse möglichst gut widerspiegelt.

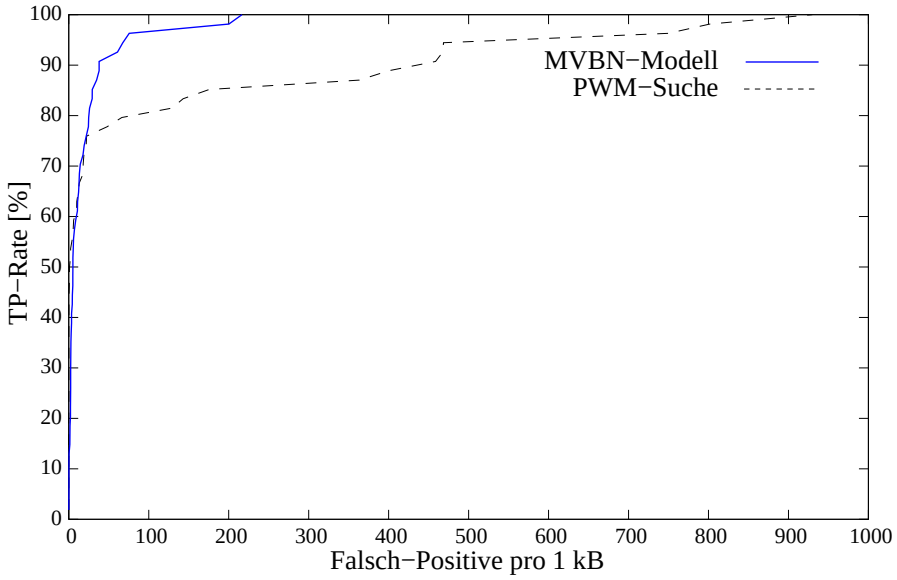


Abbildung 6.3: Figure 2 - ROC-Kurve der a priori Filterung im Vergleich zu einer Suche mit einem PWM-Modell, angewendet auf den HNF-4 Datensatz. Durchgehend blau: der a priori Filteransatz, AUC=0.981, gestrichelt: die PWM-Suche, AUC=0.909.

Zunächst wurde eine Markovkette der Ordnung 5 mit Hilfe aller 1871 menschlichen Promotorsequenzen aus EPD trainiert, wobei jeweils ein Sequenzausschnitt von 900 bp oberhalb bis 100 bp unterhalb der TSS verwendet wurde. Diese Markovkette wurde anschließend verwendet, um 1000 zufällige Sequenzen der Länge 1000 bp zu erzeugen.

Diesen zufälligen Hintergrundsequenzen wurden nun künstliche TFBS implantiert. Dazu wurden PWM-Modelle konstruiert, die $W = 10$ Spalten lang sind und einen Informationsgehalt von 10 Bit haben[7]. Im Folgenden seien die vier PWM-Modelle mit PWM_1, PWM_2, PWM_3 und PWM_4 bezeichnet. Sie wurden für zweierlei Aufgaben verwendet. Zum Einen wurden von diesen Sequenzmodellen künstliche Bindungsstellen erzeugt, die an einer zufälligen Position in den künstlichen Promotoren eingebaut werden. Zum Anderen dienten sie als Sequenzmodelle in dem a priori Filtersystem.

Jeder der 1000 künstlichen, *leeren* Promotoren wurde für drei verschiedene Datensätze verwendet:

- d_{valid}: eine Menge von TFBS-Modulen, deren Aufbau weiter unten beschrieben ist.
- d_{pseudo}: eine Menge zufälliger Anhäufungen von TFBS, die nicht die unten beschriebenen Bedingungen erfüllen. Hierfür wurden zufällig vier TFBS von jeweils einer zufälligen Wahl eines der vier PWM-Modelle erzeugt und an einer zufälligen Position der Sequenz (zwischen der 250. und 750. Position) eingebaut.

[7] 10 Bit ist ein üblicher Wert für PWM-Modelle aus TRANSFAC

- d_{single}: eine Menge von isolierten TFBS. Hierfür wurde eine zufällige TFBS erzeugt und zwischen der der 250. und 750. Position der Sequenz eingepflanzt.

d_{valid} ist der Datensatz der *echten* TFBS-Module. Das bedeutet, dass die TFBS dieser Module in einer bestimmten Weise strukturiert sind. Im Hinblick auf in der Natur auftretenden Situationen wurde die Modulstruktur folgendermaßen definiert:

1. eine PWM_1-Bindungsstelle im letzten Drittel eines künstlichen Promotors.
2. eine PWM_2-Bindungsstelle einige Hundert Basenpaare oberhalb der PWM_1-Bindungsstelle.
3. entweder eine PWM_3-Bindungsstelle in der Umgebung der PWM_2-Bindungsstelle oder eine PWM_4-Bindungsstelle unterhalb der PWM_2-Bindungsstelle, dann aber auf dem entgegengesetzten DNA-Strang.

Ein solches Szenario entspricht in etwa der Struktur eines Kernpromotors. Die PWM_1-Bindungsstelle erfüllt die Funktion einer Referenzstelle, wie sie häufig von einer TATA-Box übernommen wird. Die PWM_2-Bindungsstelle repräsentiert einen häufig im proximalen Promotorbereich bindenden Transkriptionsfaktor, wie z.B. Sp1. Die beiden letzten Bindungsstellen stehen für zwei Kooperationsalternativen, die der PWM_2-Faktor hat, und von denen zumindest eine erfüllt sein muss.

Die Abstände zwischen den implantierten TFBS wurden zufällig gemäß einer *negativen Binomialverteilung* ausgewählt [Boy04]. Diese hat für die Dauer- bzw. Abstandsmodellierung einen günstigeren Verlauf als die häufig verwendete geometrische Verteilung, da ihr maximaler Wert nicht bei Abstand 1 liegt. Die Parameter der Verteilung wurden anhand der Abstände von TFBS in der Gen-Relation der TRANSFAC-Datenbank geschätzt.

Die künstlichen Promotoren des d_{pseudo}-Datensatzes stehen für Fälle, in denen rein zufällig eine Häufung von TFBS-Vorhersagen vorliegt, die jedoch keine biologische Relevanz haben. Natürlich kann es auch unter den zufälligen Häufungen wohlgeformte TFBS-Module geben. Diese wurden nicht aus dem Datensatz entfernt.

Die interessante Frage des Versuchsaufsbau war, in welchem Maße der a priori Filteransatz in der Lage ist, wohlgeformte TFBS-Module aus dem Datensatz d_{valid} zu erkennen, und gleichzeitig isolierte TFBS (d_{single}) sowie falsche Häufungen (d_{pseudo}) abzulehnen. Dazu wurden drei Konfigurationen des hier vorgestellten Systems gegeneinander erprobt:

1. **DETAIL**: ein a priori Modell, dass das gesamte Wissen über wohlgeformte TFBS-Module enthält
2. **CLUSTER**: ein a priori Modell, dass jede Häufung von TFBS belohnt. Dieses System entspricht einem fensterbasierten TFBS-Modul-Erkennungssystem, wie MSCAN (siehe Seite 56)
3. **TRIVIAL**: ein völlig neutralisiertes a priori Modell, dass einer normalen PWM-Suche mit den vier Modellen entspricht, weil an jeder Position jedes PWM-Modell gleich wahrscheinlich ist.

Konfiguration	Vorhergesagte TFBS in Datensatz[%]		
	d_{valid}	d_{pseudo}	d_{single}
DETAIL	78.15	54.13	46.5
CLUSTER	60.72	65.85	42.1
TRIVIAL	88.44	88	87.8

Tabelle 6.1: Vorhersagen der drei verschiedenen MVBN-Konfigurationen in den drei künstlichen Datensätzen, angegeben in % aller implantierten TFBS.

Die drei Konfigurationen wurden auf die drei künstlichen Datensätze d_{valid}, d_{pseudo} und d_{single} angewendet. Gemessen wurde jeweils der Anteil der implantierten TFBS, die von den drei Konfigurationen erkannt wurden. Die Ergebnisse sind in Tabelle 6.1 dargestellt. Wie erwartet, erkannte das DETAIL Modell die meisten wohlgeformten TFBS-Module und wies etwa die Hälfte der zufälligen Häufungen aus d_{pseudo} und mehr als die Hälfte der isolierten TFBS zurück. Der Standardansatz, repräsentiert durch die Konfiguration CLUSTER, erkannte naturgemäß sowohl die echten, als auch die unechten Häufungen. Die isolierten TFBS wurden auch von diesem Modell zurückgewiesen. Während also die Berücksichtigung des Kontextes dazu führte, dass zufällige, mutmaßlich nicht funktionale TFBS-Module erheblich weniger häufig vorhergesagt wurden, kann der Standardansatz zur Erkennung von TFBS-Modulen beide Formen von Häufungen nicht unterscheiden. Es ist des Weiteren nicht überraschend, dass das einfachste Modell, TRIVIAL, die höchsten Erkennungsraten in allen drei Datensätzen hatte, da hier keinerlei Filterung erfolgte.

Eine weiterer Aspekt, unter dem die Filtereigenschaften einer a priori Modellverteilung untersucht werden kann, ist die Anzahl der Falschvorhersagen. Während die DETAIL-Konfiguration 0.3% Falschvorhersagen machte, waren es für die einfache PWM-Suche (TRIVIAL) schon 0.52%. Abbildung 6.4 illustriert ein Einfluss des DETAIL-Modells auf die Wahrscheinlichkeitskurve für eine beispielhafte Sequenz im Vergleich zur TRIVIAL-Konfiguration. Es ist zu erkennen, dass die meisten starken Falsch-Positiven Treffer der TRIVIAL-Konfiguration durch das MVBN bestraft werden, während der wahre Treffer weiterhin eine starke Bewertung erhält.

6.3 Diskussion und Ausblick

In diesem Kapitel wurde eine Möglichkeit vorgestellt, funktionell zusammenhängende TFBS-Module zu modellieren. Das Verfahren geht zunächst von den bekannten Erzeugungswahrscheinlichkeiten verschiedener Sequenzmodelle aus, wie sie z.B. von PWM-Modellen oder Markovketten ausgegeben werden. Anschließend werden diese sequenzbasierten Bewertungen von einer a priori Modellverteilung überlagert, die angibt, wie wahrscheinlich die verschiedenen Modelle unabhängig von der Sequenz sind. Diese a priori Modellverteilung wirkt als Filter, der Sequenzmodelltreffer belohnt, wenn diese

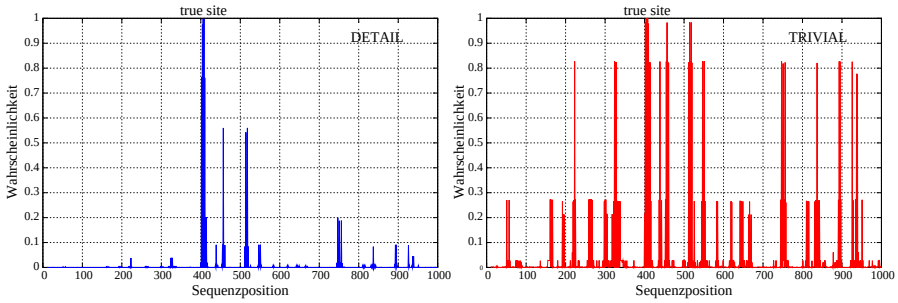


Abbildung 6.4: A posteriori Wahrscheinlichkeiten für die Positionen einer Sequenz aus dem d_{single}-Datensatz. links: mit der DETAIL-Konfiguration, rechts: mit einem normalen PWM-Modell.

aufgrund ihrer Umgebung sinnvoll erscheinen, und Vorhersagen bestraft, die aufgrund der Umgebung unwahrscheinlich sind.

Die Ableitung einer a priori Modellverteilung aus dem Kontext einer Sequenzposition übernimmt ein speziell konstruiertes Bayessches Netz, ein MVBN. Dieses wird aus einer Menge von logischen Ausdrücken, die das Hintergrundwissen über wohlgeformte Kontexte für die einzelnen Sequenzmodelle enthält, konstruiert.

Das MVBN ist eine neuartige Möglichkeit, logische Ausdrücke in probabilistische Bewertungen umzusetzen. Seinem Aufbau nach handelt es sich um einen BN-Klassifikator, während seine Funktionsweise und seine Intention Anleihen von probabilistischen Boolschen Netzen [Shm02] enthält, die für die Modellierung genregulatorischer oder metabolischer Netzwerke eingesetzt werden. Zudem teilt ein MVBN die Grundidee der *Fuzzy-Logik*, indem strikte, zweiwertige (Boolsche) Entscheidungen durch probabilistische Aufweichung verallgemeinert werden.

In Bezug auf die Anwendbarkeit der Filtereigenschaften der MVBN zur adäquateren Bewertung von TFBS muss jedoch festgehalten werden, dass diese nicht unproblematisch ist. Nur in seltenen Fällen kann Vorwissen zu Bindungsbedingungen eines Transkriptionsfaktors in einfachen, aber dennoch aussagekräftigen Ausdrücken formuliert werden. In der Evaluierungsphase des Ansatzes ergab es sich häufig, dass potenzielles Faktenwissen der Form "*Faktor A bindet immer oberhalb und gemeinsam mit Faktor B*", wie es in der Fachliteratur publiziert war, in entsprechenden Datensätzen von TFBS dieser Faktoren statistisch nicht bestätigt werden konnte und somit auch wenig geeignet für die Modellierung in MVBN erschien. Beispielsweise konnte trotz der weithin anerkannten Erkenntnis, dass der Transkriptionsfaktor Sp1 einige hundert Basenpaare oberhalb der TSS bindet und mit dem TATA-Box bindenden Protein ko-agiert, keine nennenswerte Häufung von TATA-Boxen unterhalb von Sp1 Bindungsstellen erkannt werden. Gemeinsamkeiten, die in ausreichend Beispielen tatsächlich statt fanden, beschränkten sich in den bekannten Benchmarkdatensätzen auf die klassischen Häufungstendenzen, die durch ein MVBN zwar ebenfalls modelliert werden können, jedoch ohne sichtbaren Nutzen gegenüber anderen Verfahren.

Die Modellierung von Vorwissen in MVBN-Modellen hat nach Ansicht des Autors dennoch Potenzial, erfolgreich für Sequenzanalyseaufgaben einsetzbar zu sein. Zum Einen erlaubt die Verwendung von Bayesschen Netzen die Berücksichtigung weiterer Informationseinheiten, wie der Ergebnisse aus Microarray-Experimenten. Zum Anderen könnte eine Abkehr von rein boolschem Wissen zu einer Vergrößerung der Ausdrucksstärke des MVBN führen. Neben Ausdrucksvariablen könnten nicht-boolsche Fühlervariablen einen Beitrag zur a priori Modellverteilung liefern. Einen weiteren Ansatzpunkt für Verbesserungen ist das Verfahren zur automatischen Generierung logischer Ausdrücke aus Lernstichproben.

Kapitel 7

Motivsuche unter Einbeziehung von a priori Verteilungen

In den vorherigen Kapiteln wurden Modelle zur Beschreibung und Erkennung von TFBS und TFBS-Modulen entwickelt. Diese Modelle wurden mit Hilfe von etikettierten Lernstichproben gelernt, d.h., für jedes Lernbeispiel war bekannt, ob es sich um eine TFBS oder um eine nicht funktionale Sequenz handelt. Diese komfortable Situation liegt jedoch häufig nicht vor, z.B. bei Datensätzen, die mit den in Abschnitt 2.3 beschriebenen SELEX- oder ChIP-on-chip-Verfahren gewonnen wurden. In diesen Fällen werden Motivsucheverfahren eingesetzt.

Eine Auswahl verschiedener Ansätze zur Motivsuche findet sich in Abschnitt 3.2. Die meisten dieser Verfahren arbeiten ausschließlich auf der Sequenzebene. Deterministische Verfahren geben meist Consensussequenzen aus, probabilistische Verfahren PWM-Modelle. Die Suche der kurzen Teilsequenzen in den Eingabesequenzen, aus denen diese Sequenzmodelle berechnet werden, basiert auf der Sequenzähnlichkeit. Dabei wird davon ausgegangen, dass a priori jede Position mit gleicher Wahrscheinlichkeit Startpunkt einer Motivinstanz ist.

Vielfach liegen zusätzliche Informationen vor, die einige Positionen einer Sequenz mit hoher Wahrscheinlichkeit für eine Motivinstanz empfehlen, während es für andere Positionen unwahrscheinlich ist, dass dort eine Motivinstanz ist. Beispielsweise binden einige Transkriptionsfaktoren bevorzugt in der Nähe einer TSS. Die Motivsuche sollte dementsprechend bei der Identifizierung ähnlicher Teilsequenzen Sequenzbereiche in der Nähe einer TSS bevorzugen. Ein anderes Beispiel ist die Belegung der DNA durch Histonkomplexe. Befinden sich im Bereich eines Nukleosoms Teilsequenzen, die hinsichtlich ihrer Basenpaare eine optimale TFBS wären, ist es dennoch unwahrscheinlich, dass ein Transkriptionsfaktor diese Teilsequenz tatsächlich bindet. Wäre die gleiche Teilsequenz nicht Teil eines Nukleosoms, wäre sie mit höherer Wahrscheinlichkeit eine echte TFBS.

In diesem Kapitel wird ein Ansatz entwickelt, um a priori Wissen über potentielle Positionen von Motivinstanzen bei der Motivsuche einzusetzen [Hil06a]. Der Ansatz besteht in einer Erweiterung des bekannten Motivsucheverfahrens MEME [Bai95b]. Dieses Programm verwendet den EM-Algorithmus, um ein PWM-Modell aus einer Menge von Eingabesequenzen zu berechnen. In diesem Kapitel wird das stochastische Modell, dass in MEME verwendet wird, um eine a priori Verteilung über mögliche Startpositionen für

Motivinstanzen erweitert. Der Effekt der a priori Verteilung ist, dass die Motivsuche in probabilistischer Weise hin zu Positionen geleitet wird, die entsprechend dem Vorwissen vielversprechende Positionen für Motivinstanzen sind.

Der Ansatz wird für zwei Anwendungsgebiete erprobt. Zum einen werden mittels eines statistischen Modells von Segal et al. [Seg06] Nukleosompositionen vorhergesagt und die Vorhersagen als a priori Verteilung genutzt. Zum anderen wird die Tendenz von RNA bindenden Proteinen, auf einzelsträngigen Bereichen zu binden, als a priori Verteilung eingesetzt.

In Abschnitt 7.1 wird das mathematische Modell und der EM-Algorithmus eingeführt. Abschnitt 7.2 beschreibt die beiden Möglichkeiten, a priori Verteilungen zu konstruieren. Abschnitt 7.3 fasst die Ergebnisse der Untersuchungen zusammen und Abschnitt 7.4 diskutiert diese Ergebnisse.

7.1 EM-basierte Motivsuche

7.1.1 Das EM-Prinzip

Die Aufgabe des statistischen Lernens besteht darin, für eine Stichprobe d einer Grundgesamtheit D aus einem *Wahrscheinlichkeitsmodell* $\mathcal{P}$ von D [1] jene *Instanz* p auszuwählen, die d erzeugt haben könnte. Ziel ist es dabei, etwas über die Verteilung der Grundgesamtheit zu erfahren. Dieses Prinzip wurde beispielsweise im Abschnitt 4.4 angewendet. Dort war die Grundgesamtheit D ein Merkmalsraum, aufgespannt von einer Menge von Merkmalen, Stichprobe d eine endliche Menge von Merkmalsvektoren und das Wahrscheinlichkeitsmodell die Menge aller Bayesschen Netze, die auf der D aufspannenden Merkmalsmenge definiert sind.

Ein *Schätzer* ist eine Methode, die als Eingabe eine Stichprobe d und ein Wahrscheinlichkeitsmodell $\mathcal{P}$ erhält und eine Instanz $p \in \mathcal{P}$ ausgibt. Ein bekannter Schätzer ist der *Maximum-Likelihood-Schätzer* (ML). Dieser wählt aus $\mathcal{P}$ eine Instanz $\hat{p}$ aus, welche die logarithmische Wahrscheinlichkeit der Stichprobe d maximiert:

$$\begin{aligned}
\hat{p} &= \operatorname*{argmax}_{p \in \mathcal{P}} \log p(\mathsf{d}) && (7.1)\\
&= \operatorname*{argmax}_{p \in \mathcal{P}} \log \prod_{x \in \mathsf{d}} p(x) && (7.2)\\
&= \operatorname*{argmax}_{p \in \mathcal{P}} \log \prod_{x \in \mathsf{D}} p(x)^{f(x)} && (7.3)\\
&= \operatorname*{argmax}_{p \in \mathcal{P}} \sum_{x \in \mathsf{D}} f(x) \log p(x), && (7.4)
\end{aligned}$$

[1] Wahrscheinlichkeitsmodell einer Menge X: eine Menge von Wahrscheinlichkeitsverteilungen über X.

wobei $f(x)$ Häufigkeitsstatistiken der Stichprobe sind, reelle Zahlen [2], die angeben, wie häufig ein Element x der Grundgesamtheit in der Stichprobe vertreten ist [3].

Unvollständige Daten. In vielen Fällen kommt es vor, dass die vorhandene Stichprobe nur Komponenten der Elemente der Grundgesamtheit enthält, weil bestimmte Eigenschaften dieser Elemente verborgen sind. Diese Stichprobe $\breve{\mathsf{d}}$ heißt dann *unvollständig* und kann nicht verwendet werden, um via ML-Schätzer die ML-optimale Instanz des Wahrscheinlichkeitsmodells $\mathcal{P}$ der vollständigen Daten zu berechnen. Ein Beispiel ist eine unetikettierte Stichprobe zum Lernen eines Klassifikators. Hier fehlt in der unvollständigen Stichprobe die Klassenzugehörigkeit der Stichprobenelemente, so dass eine direkte Schätzung von Verteilungen der Klassengebiete nicht möglich ist.

Grundidee des EM. Dempster entwickelte 1977 ein iteratives Verfahren, dass es ermöglicht, in Situationen mit unvollständigen Daten dennoch eine ML-Schätzung durchzuführen [Dem77]. Die Grundidee besteht darin, 1.) in jeder Iteration einen Erwartungswert der vollständigen Daten zu schätzen (E-Schritt für englisch: *expectation*), wofür die gegebenen unvollständigen Daten und die ML-Modellinstanz $\hat{p}^{t-1}$ der vorherigen Iteration verwendet werden, und 2.) für diesen Erwartungswert eine normale ML-Schätzung mit Ergebnis $\hat{p}^t$ durchzuführen (M-Schritt für englisch: *maximization*).

Das Prinzip ist immer dann leicht anwendbar, wenn jedem Element y der unvollständigen Grundgesamtheit $\breve{\mathsf{D}}$ eine Menge $A(y)$ von vollständigen Elementen $x \in A(y) \subseteq \mathsf{D}$ zugeordnet werden kann, so dass durch die $A(y)$ eine Partition auf D definiert ist. Eine Modellinstanz $p \in \mathcal{P}$ für vollständige Daten definiert dann zum einen implizit eine Verteilung über den unvollständigen Daten vermöge

$$p(y) = \sum_{x \in A(y)} p(x) \tag{7.5}$$

zum anderen für den E-Schritt entscheidende bedingte Wahrscheinlichkeiten

$$p(x \,|\, y) = \frac{p(x)}{p(y)}. \tag{7.6}$$

Der EM-Algorithmus in seiner allgemeinsten Form benötigt als Eingabe:

1. eine unvollständige Stichprobe $\breve{\mathsf{d}} \subseteq \breve{\mathsf{D}}$
2. ein Wahrscheinlichkeitsmodell $\mathcal{P}$ für die vollständige Grundgesamtheit D
3. eine Funktion A zur Partitionierung von D
4. eine Startinstanz $p^0 \in \mathcal{P}$

[2] Für den EM-Algorithmus ist es nötig, reelle Zahlen als Zähler $f(x)$ zuzulassen.

[3] Die logarithmische Variante wird verwendet, da sie numerisch robuster ist. Die zu maximierende Funktion heißt *ML-Zielfunktion*.

und hat folgenden Aufbau:

ALGORITHMUS: EM

1. **FOR** $t = 1, 2, 3, \ldots$
   a) $q := p^{t-1}$
   b) **E-Schritt:** Berechne Erwartungswerte für die Häufigkeitsstatistiken $f(x)$ der vollständigen Daten unter Verwendung des aktuellen Modells q:

$$\mathbb{E}[f(x) \,|\, q] := f(y) \cdot q(x \,|\, y) \text{ für } y : x \in A(y)$$

   c) **M-Schritt:** Führe ML-Schätzung mit Hilfe dieser Erwartungswerte durch:

$$p^t := \operatorname*{argmax}_{p \in \mathcal{P}} \sum_{x \in \mathsf{D}} \mathbb{E}[f(x) \,|\, q] \log p(x)$$

2. **NEXT** t

Der EM-Algorithmus verbessert in jeder Iteration die ML-Zielfunktion und konvergiert in einem lokalen Maximum von $\ell_p(\cdot)$ [Dem77].

7.1.2 Anwendung auf die Aufgabe der Motivsuche

Die unvollständigen Eingabedaten bei der Motivsuche sind eine Menge $\boldsymbol{S} = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n\}$ von DNA- oder RNA-Sequenzen einer festen Länge L (o.B.d.A.). Vervollständigt würde sie durch Informationen, an welchen Positionen sich in diesen Sequenzen Motivinstanzen des gesuchten Sequenzmotivs befinden. MEME, und damit die hier entwickelte Erweiterung, unterscheidet drei verschiedene Suchmodi, die jeweils auf anderen Wahrscheinlichkeitsmodellen operieren und eine andere Repräsentation der unvollständigen Bestandteile besitzen:

- **OOPS:** jede Eingabesequenz enthält genau eine Instanz des gesuchten Sequenzmotivs (**o***ne* **o***ccurrence* **p***er* **s***equence*),
- **ZOOPS:** jede Eingabesequenz enthält höchstens eine Instanz des Sequenzmotivs (**z***ero or* **o***ne* **o***ccurrence* **p***er* **s***equence*),
- **TCM:** jede Eingabesequenz enthält eine beliebige Anzahl von Sequenzmotivinstanzen (**t***wo* **c***omponent* **m***ixture*).

OOPS. Im OOPS-Modus wird die verborgene Information durch Indikatorvariablen

$$Zij = \begin{cases} 1 & : \quad \boldsymbol{s}_i \text{ hat an Position } j \text{ eine Motivinstanz} \\ 0 & : \quad \text{sonst} \end{cases} \tag{7.7}$$

mit $1 \leq i \leq n$ und $1 \leq j \leq m := L - W + 1$ beschrieben[4]. Eine vollständige Stichprobe bestünde also aus den n Sequenzen $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$ gemeinsam mit Belegungen der Indikatorvariablen Z_{ij} , wobei für jedes $1 \leq i \leq n$ genau ein $Z_{ij} = 1$ sein muss.

Das Wahrscheinlichkeitsmodell in OOPS ist ein mehrstufiger Prozess, der wie folgt arbeitet, um die i-te Sequenz der Eingabe zu erzeugen:

- Wähle zufällig die Position der Motivinstanz j ($Z_{ij} = 1$) gemäß einer a priori Verteilung σ_i über $\{1, \ldots, m\}$. Es gilt:

$$\sigma_{ij} := \sigma_i(j) = P(Z_{ij} = 1). \tag{7.8}$$

  Es gibt für jede der n Eingabesequenzen $\boldsymbol{s}_i$ genau eine a priori Verteilung σ_i. Diese a priori Verteilungen sind der zentrale Punkt in diesem Kapitel, denn über sie wird das zu berücksichtigende Zusatzwissen in die Motivsuche eingebracht.

- Sei $Z_{ij} = 1$. Dann erzeuge die Hintergrundsequenzen $s_{i1} \ldots s_{ij-1}$ und $s_{ij+W} \ldots s_{iL}$ gemäß einer Hintergrundverteilung $\boldsymbol{\theta}_0$ über Σ_{DNA}.
- Erzeuge eine Motivinstanz $s_{ij} \ldots s_{ij+W-1}$ gemäß einer probabilistischen PWM $\boldsymbol{\theta}_1$.

Das Wahrscheinlichkeitsmodell $\mathcal{P}$ besteht aus allen Sequenzmodellen $\boldsymbol{\phi}$ mit einer Hintergrundverteilung $\boldsymbol{\theta}_0$, einem PWM-Modell $\boldsymbol{\theta}_1$ und fest vorgegebenen a priori-Verteilungen $\boldsymbol{\sigma}$. Die ML-Schätzung im M-Schritt des EM-Algorithmus soll in jeder Iteration jene $\boldsymbol{\phi} = (\boldsymbol{\theta}_0, \boldsymbol{\theta}_1 \,|\, \boldsymbol{\sigma})$ bei fest gewählten $\boldsymbol{\sigma}$ finden, welche die Wahrscheinlichkeit des gegenwärtigen Erwartungswerts der vollständigen Stichprobe maximieren. Der t-te E-Schritt besteht in der Berechnung eines Erwartungswerts für die Indikatorvariablen Z_{ij}, wobei die vorherige Modellinstanz $\boldsymbol{\phi}^{t-1} = (\boldsymbol{\theta}_0^{t-1}, \boldsymbol{\theta}_1^{t-1} \,|\, \boldsymbol{\sigma})$ und die unvollständigen Daten verwendet werden:

$$\begin{aligned}
\mathbb{E}^t_{Z_{ij}} &:= \mathbb{E}[Z_{ij} \,|\, \boldsymbol{\phi}^{t-1}, \boldsymbol{s}] && (7.9)\\
&= 0 \cdot P(Z_{ij} = 0 \,|\, \boldsymbol{\phi}^{t-1}, \boldsymbol{s_i}) + 1 \cdot P(Z_{ij} = 1 \,|\, \boldsymbol{\phi}^{t-1}, \boldsymbol{s_i}) && (7.10)\\
&= P(Z_{ij} = 1 \,|\, \boldsymbol{\phi}^{t-1}, \boldsymbol{s_i}) && (7.11)\\
&= \frac{P(\boldsymbol{s}_i \,|\, Z_{ij} = 1, \boldsymbol{\phi}^{t-1}) \cdot P(Z_{ij} = 1 \,|\, \boldsymbol{\phi}^{t-1})}{\sum_{k=1}^{m} P(\boldsymbol{s}_i \,|\, Z_{ik} = 1, \boldsymbol{\phi}^{t-1}) \cdot P(Z_{ik} = 1 \,|\, \boldsymbol{\phi}^{t-1})} && (7.12)\\
&= \frac{P(\boldsymbol{s}_i \,|\, Z_{ij} = 1, \boldsymbol{\phi}^{t-1}) \cdot \sigma_{ij}}{\sum_{k=1}^{m} P(\boldsymbol{s}_i \,|\, Z_{ik} = 1, \boldsymbol{\phi}^{t-1}) \cdot \sigma_{ik}}. && (7.13)
\end{aligned}$$

Der letzte Schritt ist möglich, da die a priori Wahrscheinlichkeit für $Z_{ij} = 1$ gerade durch den festen Teil $\boldsymbol{\sigma}$ der Bedingungsteile $\boldsymbol{\phi}$ definiert ist. Im M-Schritt werden die Erwartungswerte $\mathbb{E}^t_{Z_{ij}}$ als Platzhalter für die unbekannten Daten verwendet, um die ML-Zielfunktion $\log P(\boldsymbol{S}, \boldsymbol{Z} \,|\, \boldsymbol{\phi})$ zu maximieren:

$$\boldsymbol{\phi}^t = \operatorname*{argmax}_{\boldsymbol{\phi} \in \mathcal{P}} \log P(\boldsymbol{S}, \mathbb{E}^t_{\boldsymbol{Z}} \,|\, \boldsymbol{\phi}) \tag{7.14}$$

[4] m ist die maximale Position, an der eine Motivinstanz der Länge W beginnen kann.

$$= \operatorname*{argmax}_{\boldsymbol{\phi}\in\mathcal{P}} \log \prod_{i=1}^{n} P(\boldsymbol{s}_i, \mathbb{E}^t_{\boldsymbol{Z}_i} \mid \boldsymbol{\phi}) \tag{7.15}$$

$$= \operatorname*{argmax}_{\boldsymbol{\phi}\in\mathcal{P}} \sum_{i=1}^{n} \log P(\boldsymbol{s}_i, \mathbb{E}^t_{\boldsymbol{Z}_i} \mid \boldsymbol{\phi}) \tag{7.16}$$

$$= \operatorname*{argmax}_{\boldsymbol{\phi}\in\mathcal{P}} \sum_{i=1}^{n} \log \prod_{j=1}^{m} \left(P(\boldsymbol{s}_i \mid Z_{ij} = 1, \boldsymbol{\phi})\sigma_{ij}\right)^{\mathbb{E}_{Z_{ij}}} \tag{7.17}$$

$$= \operatorname*{argmax}_{\boldsymbol{\phi}\in\mathcal{P}} \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbb{E}^t_{Z_{ij}} \left(\log P(\boldsymbol{s}_i \mid Z_{ij} = 1, \boldsymbol{\phi}) + \log \sigma_{ij}\right) \tag{7.18}$$

$$= \operatorname*{argmax}_{\boldsymbol{\phi}\in\mathcal{P}} \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbb{E}^t_{Z_{ij}} \log P(\boldsymbol{s}_i \mid Z_{ij} = 1, \boldsymbol{\phi}) + \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbb{E}^t_{Z_{ij}} \log \sigma_{ij} \tag{7.19}$$

Der Übergang von Zeile 7.16 zu Zeile 7.17 entspricht dem Übergang von einer Summe über den Elementen einer konkreten Stichprobe d zu einer mit reellen Häufigkeitsstatistiken gewichteten Summe über alle Elemente der Grundgesamtheit, in den Gleichungen 7.2 und 7.3 auf Seite 168, wobei alle nicht in der Stichprobe vertretenen Sequenzen Gewicht Null haben. So erklärt sich der Exponent $\mathbb{E}_{Z_{ij}}$.

Im M-Schritt zerfällt die Zielfunktion in zwei Doppelsummenterme, wobei nur der erste Term von den zu optimierenden $\boldsymbol{\theta}_0$ und $\boldsymbol{\theta}_1$ abhängt. Diese Doppelsumme ist im Übrigen identisch mit jener im originalen MEME-M-Schritt, da sich im M-Schritt der zusätzliche a priori Teil komplett in der zweiten Doppelsumme befindet. Für die erste Doppelsumme ergeben sich folgende ML-Schätzungen $\boldsymbol{\phi}^t$: Das PWM-Gewicht $\theta^t_{1k\mathtt{X}}$ für PWM-Spalte k und Nukleotid $\mathtt{X} \in \Sigma_{DNA}$ berechnet sich zu

$$\theta^t_{1k\mathtt{X}} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} \mathbb{E}^t_{Z_{ij}} \delta(s_{ij+k-1}, \mathtt{X})}{\sum_{\mathtt{Y}\in\Sigma_{DNA}}\sum_{i=1}^{n}\sum_{j=1}^{m} \mathbb{E}^t_{Z_{ij}} \delta(s_{ij+k-1}, \mathtt{Y})}, \tag{7.20}$$

wobei $\delta(s_{ij}, \mathtt{X}) = 1$ genau dann gilt, wenn an Position j in Sequenz $\boldsymbol{s}_i$ das Nukleotid $\mathtt{X}$ steht [Bai95b]. Die Hintergrundwahrscheinlichkeit $\theta^t_{0\mathtt{X}}$ für Nukleotid $\mathtt{X}$ berechnet sich durch

$$\theta^t_{0\mathtt{X}} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} \delta(s_{ij+k-1}, \mathtt{X}) - \sum_{w=1}^{W}\sum_{i=1}^{n}\sum_{j=1}^{m} \mathbb{E}^t_{Z_{ij}} \delta(s_{ij+w-1}, \mathtt{X})}{\sum_{\mathtt{Y}\in\Sigma_{DNA}} \left(\sum_{i=1}^{n}\sum_{j=1}^{m} \delta(s_{ij+k-1} = \mathtt{Y}) - \sum_{w=1}^{W}\sum_{i=1}^{n}\sum_{j=1}^{m} \mathbb{E}^t_{Z_{ij}} \delta(s_{ij+w-1}, \mathtt{Y})\right)}. \tag{7.21}$$

Diese Gleichung sieht komplizierter aus, als sie ist, handelt es sich doch um eine einfache stichprobenweite relative Häufigkeit eines Nukleotids, wovon noch gemäß $\mathbb{E}^t_{Z_{ij}}$ gewichtete Zählungen von Positionen abgezogen werden, die in den Bereich von Motivinstanzen fallen.

ZOOPS. Dieser Suchmodus besitzt ein ähnliches Wahrscheinlichkeitsmodell wie OOPS, jedoch mit dem Unterschied, dass die Möglichkeit eingeräumt wird, dass eine Eingabesequenz keine Motivinstanz besitzt. Diese Information gehört, wie die Startpositionen

der Motivinstanzen, zu den verborgenen Daten und wird durch Indikatorvariablen

$$Q_i = \begin{cases} 1 & : \quad \boldsymbol{s}_i \text{ enthält eine Motivinstanz} \\ 0 & : \quad \text{sonst} \end{cases} \tag{7.22}$$

beschrieben. Der Zufallsprozess wird um eine Stufe erweitert. Zunächst wird gemäß eines Bernoulli-Prozesses mit Wahrscheinlichkeit $\gamma = P(Q_i = 1)$ entschieden, ob eine Sequenz eine Motivinstanz enthalten soll[5]. Im Falle einer Entscheidung für eine Motivinstanz wird die Sequenz gemäß dem OOPS-Wahrscheinlichkeitsprozess erzeugt, anderenfalls wird jede Sequenzposition von der Hintergrundverteilung $\boldsymbol{\theta}_0$ erzeugt. Das Wahrscheinlichkeitsmodell besteht also aus drei variablen Komponenten und den festen a priori Wahrscheinlichkeiten: $\boldsymbol{\phi} = (\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \gamma \,|\, \boldsymbol{\sigma})$. Der E-Schritt ändert sich im ZOOPS-Modus zu:

$$\mathbb{E}^t_{Z_{ij}} \quad = \quad \frac{P(\boldsymbol{s}_i \,|\, Z_{ij} = 1, \boldsymbol{\phi}^{t-1}) \cdot \sigma_{ij} \cdot \gamma^{t-1}}{P(\boldsymbol{s}_i \,|\, Q_i = 0, \boldsymbol{\phi}^{t-1}) \cdot (1 - \gamma^{t-1}) + \sum_{k=1}^{m} P(\boldsymbol{s}_i \,|\, Z_{ik} = 1, \boldsymbol{\phi}^{t-1}) \cdot \sigma_{ik} \gamma^{t-1}}. \tag{7.23}$$

Im M-Schritt lassen sich in der ML-Zielfunktion die von $\boldsymbol{\theta}_0$ und $\boldsymbol{\theta}_1$ abhängigen Terme von denen trennen, die von γ abhängen (und wie schon im OOPS-Modus von den Teilen, die ausschließlich von $\boldsymbol{\sigma}$ abhängen). Die Maximierung der von $\boldsymbol{\theta}_0$ und $\boldsymbol{\theta}_1$ geschieht wieder über die Gleichungen 7.20 und 7.21. Der γ-Teil wird folgendermaßen maximiert [Bai95b]:

$$\gamma^t = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbb{E}^t_{Z_{ij}}. \tag{7.24}$$

TCM. In diesem Suchmodus kann es eine beliebige Anzahl von Motivinstanzen in jeder der Eingabesequenzen geben. In MEME wird dazu die unvollständige Eingabe, d.h. die n Sequenzen der Länge L, in eine Menge von Sequenzfenstern der Länge W zerlegt, wobei W die Länge des gesuchten Sequenzmotivs ist. Das Fenster, dass in Sequenz $\boldsymbol{s}_i$ an Position j beginnt, wird im Folgenden mit $\boldsymbol{s}_{ij}$ bezeichnet (nicht zu verwechseln mit dem einzelnen Zeichen s_{ij}).

Jedes dieser Fenster wird in MEME von einem Zwei-Komponenten-Mischverteilungsmodell mit Mischverteilungskoeffizient λ erzeugt. Die erste Komponente ist, wie schon zuvor, eine probabilistische PWM $\boldsymbol{\theta}_1$, die zweite Komponente die Hintergrundverteilung $\boldsymbol{\theta}_0$. Die den unvollständigen Daten fehlende Information, nämlich welches Fenster welcher Mischverteilungskomponente zugewiesen wird, wird erneut durch Indikatorvariablen Z_{ij} beschrieben. Während MEME diese als a priori gleichverteilt ansieht, wird in der hier entwickelten Erweiterung die a priori Verteilungen $\boldsymbol{\sigma}$ verwendet. Der EM-Algorithmus wird auf der abgeleiteten Stichprobe von Sequenzfenstern ausgeführt, um die Parameter der ML-Instanz $\hat{\boldsymbol{\phi}} = (\hat{\boldsymbol{\theta}}_0, \hat{\boldsymbol{\theta}}_1, \hat{\lambda} \,|\, \boldsymbol{\sigma})$ des Wahrscheinlichkeitsmodells $\mathcal{P}$ zu approximieren.

[5] Obwohl Variablen Q_i unterschieden werden, ist γ konstant für alle Sequenzen

Die Umwandlung der Daten in eine Menge von sich überlappenden Sequenzfenstern ist problematisch, da die einzelnen Stichprobenelemente nun nicht mehr unabhängig voneinander sind. So könnte der unerwünschte Effekt auftreten, dass zwei sich überlappende Fenster beide zu Motivinstanzen werden. MEME führt dazu nach jeder EM-Iteration einen Korrekturschritt durch, der einerseits die Schätzung für λ anpasst und andererseits die Erwartungswerte $\mathbb{E}^t_{Z_{ij}}$ überlappender Fenster in einer *Winner-takes-it-all*-Strategie verändert. Dieser Korrekturschritt ist unabhängig von der Erweiterung des stochastischen Modells und kann in [Bai95a] nachgeschlagen werden.

Der E-Schritt im TCM-Modus wird folgendermaßen formalisiert:

$$\mathbb{E}^t_{Z_{ij}} = P(Z_{ij} = 1 \mid \boldsymbol{\phi}^{t-1}, \boldsymbol{s}_{ij}) \tag{7.25}$$

$$= \frac{P(\boldsymbol{s}_{ij} \mid Z_{ij} = 1, \boldsymbol{\phi}^{t-1})\sigma_{ij}\lambda^{t-1}}{P(\boldsymbol{s}_{ij} \mid Z_{ik} = 1, \boldsymbol{\theta}_1^{t-1})\sigma_{ij}\lambda^{t-1} + P(\boldsymbol{s}_{ij} \mid Z_{ik} = 0, \boldsymbol{\theta}_0^{t-1})(1-\sigma_{ij})(1-\lambda^{t-1})} \tag{7.26}$$

Im Prinzip überlagern sich nun zwei a priori Verteilungen: 1.) der positionsunabhängige Koeffizient λ und 2.) die positionsspezifischen a priori Verteilungen $\boldsymbol{\sigma}$. Zunächst mag es nicht sinnvoll erscheinen, angesichts der eigenen a priori Verteilung das λ aus MEME weiter zu verwenden. Jedoch kann über die Tatsache, dass λ im EM-Algorithmus optimiert wird, während die a priori Verteilungen $\boldsymbol{\sigma}$ als Vorwissen festgelegt sind, im Falle einer hohen Anzahl *guter* Motivinstanzen λ in Richtung der $\boldsymbol{\theta}_1$-Komponente verschoben werden. Bei ausschließlicher Verwendung der $\boldsymbol{\sigma}$ würde sich nach einer EM-Iteration nichts mehr ändern können.

Der M-Schritt des TCM-Modus unterscheidet sich ebenfalls gegenüber den M-Schritten von OOPS und ZOOPS.

$$\boldsymbol{\phi}^t = \operatorname*{argmax}_{\boldsymbol{\phi} \in \mathcal{P}} \log P(\boldsymbol{S}, \mathbb{E}^t_{\boldsymbol{Z}} \mid \boldsymbol{\phi}) \tag{7.27}$$

$$= \operatorname*{argmax}_{\boldsymbol{\phi} \in \mathcal{P}} \sum_{i=1}^{n}\sum_{j=1}^{m} \log P(\boldsymbol{s}_{ij}, \mathbb{E}^t_{Z_{ij}} \mid \boldsymbol{\phi}) \tag{7.28}$$

$$= \operatorname*{argmax}_{\boldsymbol{\phi} \in \mathcal{P}} \sum_{i=1}^{n}\sum_{j=1}^{m} \log \Big[(P(\boldsymbol{s}_{ij} \mid Z_{ij} = 1, \boldsymbol{\theta}_1)\sigma_{ij}\lambda)^{\mathbb{E}^t_{Z_{ij}}} \tag{7.29}$$

$$+ (P(\boldsymbol{s}_{ij} \mid Z_{ij} = 0, \boldsymbol{\theta}_0)(1-\sigma_{ij})(1-\lambda))^{(1-\mathbb{E}^t_{Z_{ij}})} \Big]$$

$$= \operatorname*{argmax}_{\boldsymbol{\phi} \in \mathcal{P}} \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbb{E}^t_{Z_{ij}} (\log P(\boldsymbol{s}_{ij} \mid Z_{ij} = 1, \boldsymbol{\theta}_1) + \log\sigma_{ij} + \log\lambda) \tag{7.30}$$

$$+(1 - \mathbb{E}^t_{Z_{ij}}) (\log P(\boldsymbol{s}_{ij} \mid Z_{ij} = 0, \boldsymbol{\theta}_0) + \log(1-\sigma_{ij}) + \log(1-\lambda))$$

$$= \operatorname*{argmax}_{\boldsymbol{\phi} \in \mathcal{P}} \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbb{E}^t_{Z_{ij}} \log P(\boldsymbol{s}_{ij} \mid Z_{ij} = 1, \boldsymbol{\theta}_1) \tag{7.31}$$

$$+ \sum_{i=1}^{n}\sum_{j=1}^{m} (1 - \mathbb{E}^t_{Z_{ij}}) \log P(\boldsymbol{s}_{ij} \mid Z_{ij} = 0, \boldsymbol{\theta}_0)$$

$$+\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbb{E}_{Z_{ij}}^{t}\sigma_{ij}+(1-\mathbb{E}_{Z_{ij}}^{t})(1-\sigma_{ij})$$
$$+\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbb{E}_{Z_{ij}}^{t}\lambda+(1-\mathbb{E}_{Z_{ij}}^{t})(1-\lambda).$$

Wieder ergibt sich das erfreuliche Ergebnis, dass die freien Parameter des Modells getrennt optimiert werden können. Die erste beiden Doppelsummen sind nur abhängig von $\boldsymbol{\theta}_1$ und $\boldsymbol{\theta}_0$ und werden analog zu den Gleichungen 7.20 und 7.21 maximiert. Die dritte Doppelsumme enthält nur die konstanten a priori Verteilungen und kann bei der ML-Schätzung ignoriert werden. Die vierte Doppelsumme ist abhängig von dem zu schätzenden Mischverteilungskoeffizienten λ. Dieser Ausdruck wird durch

$$\lambda^t = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbb{E}_{Z_{ij}}^{t}}{nm} \tag{7.32}$$

maximiert [Bai95a].

7.2 Verwendung von a priori Verteilungen

Nachdem das mathematische Modell von MEME um a priori Verteilungen $\boldsymbol{\sigma}_i$ über Startpositionen j von Motivinstanzen in einer Sequenz $\boldsymbol{s}_i$ erweitert wurde, wird in diesem Abschnitt erläutert, wie solche a priori Verteilungen aus Vorwissen konstruiert werden können. Dieses Prinzip wird dann anhand zweier Beispiele demonstriert, die im Ergebnisteil dieses Kapitels wiederkehren werden: 1.) Vorhersagen von gebundenen Histonkomplexen im Zuge der TFBS-Motivsuche und 2.) Vorhersagen der lokalen Einzelsträngigkeit von RNA-Ketten im Zuge der Motivsuche für RNA bindende Faktoren.

Vorwissen über die Eignung einer Position j einer Sequenz $\boldsymbol{s}_{ij}$ als Motivinstanz lässt sich in vielen Fällen über einfache numerische Bewertungen $B_{ij} \in \mathbb{R}$ ausdrücken. Die beiden verwendeten Beispiele unterscheiden sich nur darin, gemäß welcher Vorgaben diese Bewertungen vergeben werden. Ziel ist es jedoch, eine Wahrscheinlichkeitsverteilung $\boldsymbol{\sigma}_i$ über alle möglichen Positionen j einer Sequenz $\boldsymbol{s}_i$ zu erhalten. Die Bewertungen B_{ij} selbst sind in der Regel keine gültige Verteilung, da sie sich nicht zu Eins summieren. Wahrscheinlichkeiten σ_{ij} werden deshalb wie folgt definiert:

$$\sigma_{ij} = \frac{B_{ij}+\pi}{\sum_{k=1}^{m}(B_{ik}+\pi)}. \tag{7.33}$$

Die Variable $\pi \geq 0$ dient als Glättungskoeffizient. Für die im Vergleich zu den den B_{ij} sehr großen Werte von π nähert sich die a priori Verteilung der Gleichverteilung an. In allen drei Suchmodi entspricht dies dem originalen MEME-Ansatz. Für den extremen Wert $\pi = 0$ kann unter Umständen die a priori Verteilung den EM-Algorithmus dominieren, wenn nämlich die B_{ij} sehr ungleich verteilt sind. Für ein $B_{ij} = 0$ würde das

bedeuten, dass Position j unter keinen Umständen Motivinstanz sein kann. Die richtige Wahl von π ist anwendungsabhängig, und hängt vom Werteniveau der B_{ij} und davon ab, wie viel Vertrauen der a priori Information entgegengebracht wird.

7.2.1 Nukleosombindungsstellen

Das in Abschnitt 2.3 beschriebene *ChIP on Chip*-Verfahren lokalisiert genomweit DNA-Regionen, die von einem bestimmten Transkriptionsfaktor gebunden werden. Die Ausgabe des Verfahrens ist eine Menge genomischer Sequenzen, die eine Länge von mehreren Hundert Basenpaaren haben. Klassische Motivsuchverfahren tun sich mit derart langen Sequenzen schwer, da die gesuchten TFBS recht kurz und variabel sind.

Häufig kommt es vor, dass vorhergesagte TFBS in *in vivo* Experimenten nicht bestätigt werden können. Ein Grund dafür ist, dass *in vivo* andere Bedingungen vorliegen, die eine Bindung verhindern. Besonders Sequenzregionen, die um einen Histonkomplex gewunden sind, sind für Transkriptionsfaktoren schlecht zugänglich [Nar07]. Wenn bekannt wäre, wo genau im Genom sich Nukleosomen befinden, wären dass brauchbare Informationen für die Motivsuche.

Bisher gibt es keine hochauflösenden, vollständigen und vor allem sicheren Daten zur Nukleosombesetzung irgendeines Genoms. Yuan et al. veröffentlichten Daten für Chromosom 3 der Hefe [Yua05], Lee et al. sogar für ein ganzes Genom, jedoch in einer sehr niedrigen Auflösung [Liu06]. Sie berechneten lediglich einen Wert für eine ganze Sequenzregion zwischen zwei Genen. Hinzu kommt, dass Nukleosomen ein dynamisches Verhalten besitzen, und durch spezielle Proteine verschoben oder aufgelöst werden können [Seg06].

Im Jahr 2006 veröffentlichten Segal et al. ein mathematisches Modell zur Vorhersage der durchschnittlichen Nukleosombesetzung einer beliebigen Sequenzposition [Seg06]. Dieses basiert auf einer statistischen Analyse hochauflösender Daten für Hefe und dem Huhngenom. Die Vorhersage geschieht mit einer Art PWM-Modell, wobei die optimale Ausrichtung von Treffern hinsichtlich der für Nukleosomen üblichen Periodizität (siehe Abschnitt 2.1.1) und hinsichtlich der Vermeidung von Überlappungen berechnet wird. Die Vorhersage der Nukleosombindungsstellen benötigt lediglich die Sequenzinformation, weswegen es sich als Zusatzinformation für die Motivsuche bei gegebenen Sequenzdatensatz leicht einsetzen lässt. Der sequenzabhängige Teil vernachlässigt natürlich alle dynamischen Veränderungen des Chromatins. Jedoch proklamieren Segal et al., dass ihr Modell über 50% der Nukleosombesetzung erklären kann [Seg06].

Das Modell von Segal et al. liefert Wahrscheinlichkeiten N_{ij} dafür, dass Position j in Sequenz $\boldsymbol{s}_i$ Teil eines Nukleosoms ist. Die Arbeitshypothese besteht darin, dass eine hohe Wahrscheinlichkeit N_{ij} die Wahrscheinlichkeit verringert, dass an Position j eine Motivinstanz ist. Dies wird durch folgende Positionsbewertungen ausgedrückt:

$$B_{ij} := 1 - N_{ij}. \tag{7.34}$$

7.2.2 RNA-Sekundärstruktur

Das zweite hier vorgestellte Beispiel für a priori Informationen bei der EM-basierten Motivsuche lieferte die Motivation für die Entwicklung der MEME-Erweiterung und ist im Zusammenhang mit der Motivsuche von DNA-bindenden Proteinen nicht sinnvoll einsetzbar.

Ausgangspunkt waren statistische Untersuchungen von Michael Hiller an regulierenden Proteinen für alternatives Splicing. Diese binden wie Transkriptionsfaktoren an für sie spezifischen Bindungsstellen, jedoch auf der mRNA eines transkribierten Gens. Länge und Komplexität[6] dieser Sequenzmotive sind vergleichbar mit denen von Transkriptionsfaktoren.

Wie im Unterabschnitt 2.1.3 auf Seite 13 bereits angedeutet wurde, können entfernt voneinander liegende, komplementäre Basen von RNA-Sequenzen eine Basenpaarung eingehen. Das hat zur Folge, dass RNA-Moleküle abhängig von ihrer Sequenz eine dreidimensionale Struktur annehmen. Existiert beispielsweise für eine ganze RNA-Teilsequenz in einigem Abstand eine komplementäre Sequenz, so besteht die Möglichkeit, dass diese beiden Teilsequenzen einen Doppelstrang (englisch: *stem*) und die Teilsequenz dazwischen eine einzelsträngige Schleife (englisch: *loop*) bilden.

Die Untersuchungen von Hiller ergaben, dass viele RNA-bindende Proteine einzelsträngige Sequenzbereiche zur Bindung bevorzugen [Hil06b]. Ziel war es, dieses Wissen bei der Motivsuche zu berücksichtigen. Sequenzbasierte Motivsuchverfahren wie MEME finden möglicherweise starke Sequenzmotive, wobei diese in doppelsträngigen Bereichen liegen. Zwar gibt es auch Algorithmen zur Vorhersage von RNA-Strukturelementen, diese suchen jedoch spezielle Strukturen und keine allgemeine Eigenschaft, wie die Einzelsträngigkeit einer RNA-Sequenz.

RNA faltet sich, genau wie Proteine, in eine energetisch möglichst günstige Struktur. Diese kann nicht mit Sicherheit vorausgesagt werden, denn möglicherweise gibt es viele alternative Strukturen mit optimaler Energie. Folglich kann für eine bestimmte RNA-Teilsequenz auch nur eine Wahrscheinlichkeit dafür berechnet werden, ob diese einzelsträngig ist oder Teil einer Basenpaarung.

Die Bewertung B_{ij} einer Position j einer RNA-Sequenz $\boldsymbol{s}_i$ ist die Wahrscheinlichkeit $\mathsf{PU}_{i,j,W}$ dafür, dass das Sequenzintervall $S_{ij} \ldots s_{j+W-1}$ einzelsträngig vorliegt. Diese Wahrscheinlichkeiten werden wie folgt berechnet:

$$\mathsf{PU}_{i,j,W} = e^{\frac{E^{\text{all}} - E^{\text{unpaired}}_{i,j,j+W-1}}{RT}}, \tag{7.35}$$

wobei E^{all} die Summe der freien Energien aller möglichen Strukturen einer RNA-Sequenz ist, $E^{\text{unpaired}}_{i,j,j+W-1}$ die Summe aller Strukturen, in denen die Teilsequenz $s_ij \ldots s_{ij+W-1}$ einzelsträngig ist, R die Gaskonstante und T die Temperatur ist. Die beiden Energiesummen werden mit dem Programm RNAfold berechnet, dass hierfür Algorithmen der

[6] Im informationstheoretischen Sinne

dynamischen Programmierung einsetzt [Hof06]. Diese Art der Berechnung bezieht alle Strukturen ein, und abstrahiert von speziellen Strukturelementen.

7.3 Ergebnisse

7.3.1 Nukleosombindungsstellen

Die Verwendung von Vorhersagen über das Vorhandensein von Nukleosomen an den jeweiligen Sequenzpositionen bei der Motivsuche wurde an einer Sammlung von *ChIP on chip*-Datensätzen aus Hefegenomen untersucht. Jeder dieser Datensätze besteht aus einer Menge von DNA-Segmenten, für die eine Bindung eines bestimmten Transkriptionsfaktors nachgewiesen wurde. Für jeden untersuchten Transkriptionsfaktor ist ein charakteristisches Sequenzmotiv aus der Literatur bekannt. Für eine Motivsuche kann deshalb entschieden werden, ob das richtige Motiv gefunden wurde. Der Nutzen der a priori Verteilung wurde untersucht, indem sowohl MEME als auch die MEME-Erweiterung auf die Sequenzdatensätze angewendet wurde. Dabei wurde protokolliert, in wie vielen Fällen die beiden Programme zuerst das bekannte Sequenzmotiv fanden. Ein ähnlicher Versuchsaufbau, der dieselben Datensätze verwendet, wurde von Narlikar et al. verwendet, um den Nutzen des a priori Wissens in ihrem *Gibbs-Sampling*-Verfahren *PRIORITY* zu bestimmen [Nar07].

Datensätze. Die Datensatzsammlung stammt aus einer Veröffentlichung von Harbison et al., in der sie in *ChIP on chip*-Experimenten die Bindungsregionen von 203 Hefe-Transkriptionsfaktoren auf insgesamt 6140 Sequenzbereichen zwischen Genen bestimmten [Har04]. Für jeden Transkriptionsfaktor wurden mindestens zwei *ChIP on chip*-Experimente für verschiedene physiologische Umgebungen der Hefezellen durchgeführt, jeweils ein Experiment in gesättigter Umgebung und mindestens ein weiteres in einer von zwölf möglichen Stressumgebungen durchgeführt[7]. Wie in [Nar07] wurden für die hier beschriebenen Experimente nur solche Datensätze verwendet, die aus mindestens 10 Sequenzen bestehen, für welche die Irrtumswahrscheinlichkeit [8] kleiner als 0.0001 war und für die ein Sequenzmotiv in der Literatur genannt wurde.

Übrig blieben 105 Datensätze für 81 Transkriptionsfaktoren. Ein Datensatz besteht aus durchschnittlich 36.6 Sequenzen. Die durchschnittliche Sequenzlänge beträgt 433 bp.

[7] Säure, Nährstoffmangel, Phosphatmangel, Vitaminmangel, Aminosäurenmangel, Galaktose, Raffinose, Hitze, Zellteilung, hohe/mittlere/niedrige Konzentration von Wasserstoffperoxid.

[8] Wahrscheinlichkeit dafür, dass die Sequenz von einem *ChIP on chip*-Experiment ausgegeben wird, ohne dass der Transkriptionsfaktor wirklich gebunden hat.

Motivsuche. Jeder der 105 Datensätze wird als Eingabe der originalen MEME-Software und für verschiedene Konfigurationen der hier vorgestellten Erweiterung verwendet. Beide Programme werden mit fest eingestellter Motivlänge $W = 8$ einmal im OOPS-Modus und einmal im ZOOPS-Modus ausgeführt. Das erweiterte MEME wurde mit verschiedenen Werten für den Glättungskoeffizienten π ausgeführt ($\pi \in \{0, 0.2, 0.3, 0.4\}$). Es ergaben sich also jeweils zehn Ergebnis-Sequenzmotive in Form einer probabilistischen PWM pro Datensatz (jeweils fünf Motive für OOPS und ZOOPS), die mit dem in der Literatur genannten Sequenzmotiven der jeweiligen Transkriptionsfaktoren verglichen wurden.

Für den Vergleich mit dem bekannten Sequenzmotiv wird ein Abstandsmaß für PWM verwendet, dass in [Har04] veröffentlicht wurde. Sie definieren den Abstand D zwischen zwei PWM $\boldsymbol{\theta}$ und $\boldsymbol{\theta}'$ durch

$$D(\boldsymbol{\theta}, \boldsymbol{\theta}') = \frac{1}{\omega} \sum_{i=1}^{\omega} \frac{1}{\sqrt{2}} \sum_{\mathtt{X} \in \Sigma_{DNA}} (\theta_{i\mathtt{X}} - \theta'_{i\mathtt{X}})^2, \tag{7.36}$$

wobei ω die Länge des überlappenden Bereiches beider PWM bezeichnet. Die beiden PWM werden gegeneinander verschoben, um den Abstand für jede mögliche Überlappung der Spalten zu berechnen, für die $\omega > 5$ gilt. Zusätzlich wird auch die reverskomplementäre PWM von $\boldsymbol{\theta}$ mit $\boldsymbol{\theta}'$ in dieser Weise verglichen. Der minimal mögliche Abstand D einer solchen Überlappung gilt hier als Maß der Übereinstimmung zwischen zwei Sequenzmotiven. Das Ergebnis einer Motivsuche wird als erfolgreich bewertet, wenn der minimale Abstand des ausgegebenen Sequenzmotivs und des bekannten Motivs kleiner als 0.25 ist. Da die bekannten Sequenzmotive in [Har04] nur in Form von Consensussequenzen angegeben sind, werden diese zuvor folgendermaßen in PWM-Modelle umgewandelt:

- Bezeichnet ein Zeichen der Consensussequenz ein konkretes Nukleotid, so erhält dieses Nukleotid in der entsprechenden PWM-Spalte die Wahrscheinlichkeit 0.964. Alle anderen Nukleotide erhalten die Wahrscheinlichkeit 0.012.
- Bezeichnet ein Zeichen der Consensussequenz zwei mögliche Nukleotide, so erhalten diese Nukleotide jeweils die Wahrscheinlichkeit 0.488, die übrigen Nukleotide 0.012.
- Im Falle eines `N` erhalten alle Nukleotide die Wahrscheinlichkeit 0.25.

Ergebnisse. Die absoluten Häufigkeiten der erfolgreichen Motivsuchen sind nach Programmkonfigurationen aufgeschlüsselt in Tabelle 7.1 dargestellt. Das originale MEME erkannte war im OOPS- bei 33 Datensätzen und im ZOOPS-Modus bei 40 Datensätzen erfolgreich. Bei 29 Datensätzen lag MEME in beiden Suchmodi gleichzeitig richtig. Unter Verwendung der a priori Verteilung mit Glättungskoeffizienten $\pi = 0.3$ konnten im ZOOPS-Modus bekannte Motive für 44 Datensätze erkannt werden. Darunter waren mit einer Ausnahme auch alle Datensätze, in denen das originale MEME erfolgreich war. Dadurch ergibt sich eine Steigerung an richtig erkannten Motiven von 10% gegenüber MEME. Im OOPS-Modus lag der optimale Wert für π bei 0.2. Unter Verwendung dieses

Suchmodus	**MEME**	$\pi = 0$	$\pi = 0.2$	$\pi = 0.3$	$\pi = 0.4$
OOPS	33	32	38	34	34
ZOOPS	40	31	43	44	43

Tabelle 7.1: Ergebnisse der Motivsuche in 105 Datensätzen genomischer Sequenzen der Hefe. Angegeben ist die Anzahl der Datensätze, in denen die einzelnen Programme erfolgreich waren. Die erste Spalte bezieht sich auf das originale MEME mit uniformer Positionsverteilung, die weiteren Spalten beziehen sich auf die hier vorgestellte Erweiterung für verschiedene Werte des Glättungskoeffizienten π.

Suchmodus	**MEME**	$\pi = 0$	$\pi = 0.2$	$\pi = 0.3$	$\pi = 0.4$
OOPS	0.384	0.589	0.481	0.467	0.453
ZOOPS	0.324	0.547	0.398	0.372	0.367

Tabelle 7.2: Durchschnittliche Wahrscheinlichkeit der gefundenen Motivinstanzen, kein Teil eines Nukleosoms zu sein.

Werts konnten in 38 Datensätzen bekannte Motive entdeckt werden. Auch für die anderen Werte von $\pi > 0$ zeigte sich eine leichte Steigerung der Erkennungsrate gegenüber dem originalem MEME. Ohne Glättung der Verteilung ($\pi = 0$) wurde hingegen eine Verringerung der Anzahl der richtig erkannten Motive beobachtet. Dies deutet darauf hin, dass die Vorhersagen über Nukleosompositionen durch das Modell von Segal et al. sehr strikt sind, die ungeglättet häufig ganze Abschnitte der Eingabesequenzen für den EM-Algorithmus unerreichbar machen. Da Segal et al. einräumen, dass ihre Vorhersagen nur etwa 50% der tatsächlichen Verteilung von Nukleosomen im Genom erklären können, wäre eine allzu große Gewichtung der Motivsuche auf die a priori Information, wie sie durch sehr niedrige π geschieht, ohnehin nicht empfehlenswert.

Die Instanzen der gefundenen Motive wiesen für alle Konfigurationen, welche die a priori Informationen nutzten, eine höhere Wahrscheinlichkeit auf, kein Teil eines Nukleosoms zu sein. Die durchschnittlichen Wahrscheinlichkeiten sind in Tabelle 7.2 zusammengefasst. Bei der strikten Anwendung der a priori Verteilungen (ohne Glättung) waren die Motivinstanzen, wie zu erwarten war, besonders unbesetzt von Nukleosomen. Jedoch muss eingeräumt werden, dass die in dieser Konfiguration gefundenen Motive eine niedrige statistische Signifikanz besaßen. Zudem ist die Tendenz zu beobachten, dass sich im OOPS-Modus die gewünschte Eigenschaft, kein Teil eines Nukleosoms zu sein, stärker durchsetzte als im ZOOPS-Modus.

7.3.2 RNA-Sekundärstuktur

Die Einbeziehung der lokalen Einzelsträngigkeit von RNA-Sequenzen bei der Motivsuche wurde an künstlichen Sequenzen und biologischen Datensätzen untersucht.

Künstliche Datensätze. Intention der künstlichen Sequenzdatensätze ist es, kontrolliert sowohl Motivinstanzen in sehr wahrscheinlich einzelsträngige Teilsequenzen als auch in konstruiert doppelsträngige Strukturen einzubauen, und zu untersuchen, wie die Verwendung der a priori Informationen das Ergebnis im Vergleich zu einem Lauf des originalen MEME verändert.

Für acht Datensätze wurden verschiedene Szenarien entworfen. Jeder der Datensätze besteht aus 20 Sequenzen. Jede der Sequenzen enthält in ihrem Zentrum eine *Stem-Loop*-Struktur, d.h., einen doppelsträngigen Teil von 12 bp, an dessen Ende sich blasenartig eine einzelsträngige Schleife befindet (siehe z.B. die Strukturen in Abbildung 7.2). Dieses Strukturelement wird beiderseits von zufälligen Sequenzen flankiert. Neben den bekannten Basenpaarungen zwischen `A` und `T` sowie `C` und `G` werden auch `G-T`-Basenpaare zugelassen, die bei der Ausbildung von RNA-Strukturen ebenfalls auftreten können.

Motivinstanzen wurden entweder als feste Sequenz oder gemäß eines PWM-Modells zufällig gezogen in die Sequenzen eingebaut, entweder in den doppelsträngigen *stem*-Teil ($\mathsf{MOTIV_{DS}}$) oder in den einzelsträngigen *loop*-Teil ($\mathsf{MOTIV_{ES}}$). Durch die Zulassung von `G-T`-Paarungen wurde verhindert, dass sich in einer Sequenz eine perfekt komplementäre Instanz von $\mathsf{MOTIV_{DS}}$befindet. Die Motivlänge W lag bei 6 bp. Die unterschiedlichen Szenarien waren wie folgt aufgebaut:

- **Datensätze 1-4:** Jede Sequenz eines solchen Datensatzes enthält sowohl eine $\mathsf{MOTIV_{DS}}$-Instanz als auch eine $\mathsf{MOTIV_{ES}}$-Instanz. Im Datensatz 1 sind die Motivinstanzen feste Sequenzen, wobei die $\mathsf{MOTIV_{DS}}$-Instanz aus der $\mathsf{MOTIV_{ES}}$-Instanz durch Vertauschung der Positionen entsteht. In den Datensätzen 2 und 4 wurden für jede Sequenz die Instanzen gemäß PWM zufällig erzeugt, wobei die $\mathsf{MOTIV_{DS}}$-PWM aus der $\mathsf{MOTIV_{ES}}$-PWM durch Vertauschung der Spalten entstand. Datensatz drei enthält wieder feste Sequenzen, wobei 25% der Positionen der $\mathsf{MOTIV_{ES}}$-Instanzen mutiert wurden, um $\mathsf{MOTIV_{ES}}$ gegenüber $\mathsf{MOTIV_{DS}}$zu schwächen.
- **Datensatz 5:** Jede Sequenz enthält nur eine $\mathsf{MOTIV_{DS}}$-Instanz, die zufällig gemäß einer PWM gezogen wurde.
- **Datensatz 6:** Die Sequenzen enthalten Instanzen eines Motivs der Länge 12, wobei nur 6 bp der Instanzen im einzelsträngigen *loop* liegen.
- **Datensatz 7:** 50% der Sequenzen haben eine $\mathsf{MOTIV_{ES}}$-Instanz, 40% der Sequenzen eine $\mathsf{MOTIV_{DS}}$-Instanz und 10% keine Instanz.
- **Datensatz 8:** 30% der Sequenzen haben $\mathsf{MOTIV_{DS}}$- und $\mathsf{MOTIV_{ES}}$-Instanzen, 20% haben zwei $\mathsf{MOTIV_{ES}}$-Instanzen, je 20 haben eine $\mathsf{MOTIV_{ES}}$-Instanz oder eine $\mathsf{MOTIV_{DS}}$-Instanz und 10% haben keine Instanz.

Untersuchungen fanden stets als Vergleich des originalen MEME und der hier vorgestellten Erweiterung statt, wobei im letzteren Fall zuvor a priori Verteilungen für jede Sequenz eines Datensatzes gemäß Unterabschnitt 7.2.2 berechnet wurden. Als Glättungskoeffizient π wurde für die Datensätze 1-6 der Wert 0.1 verwendet, für die Datensätze 7 und 8 der Wert 0.01.

Die Datensätze 1 bis 6 wurden verwendet, um beide Programme im OOPS-Modus zu vergleichen. Eine Heuristik in MEME zur Startmodellauswahl hat zur Folge, dass MEME bei gleichstarken Motiven stets dasjenige ausgibt, das in der ersten Sequenz zuerst eine Instanz besitzt. Die a priori-Erweiterung fand dagegen in jedem Fall (d.h. bei jeder untersuchten Reihenfolge der Eingabesequenzen) das einzelsträngige $\mathsf{MOTIV}_{\mathsf{ES}}$. Auch im Datensatz 2, in dem die Instanzen von einem PWM-Modell erzeugt wurden und demzufolge nicht mehr perfekt übereinstimmen, wurde bei Verwendung der a priori Verteilungen stets $\mathsf{MOTIV}_{\mathsf{ES}}$ als Ergebnis ausgegeben, während MEME bei verschiedenen Sortierungen der Sequenzen zu gleichen Teilen $\mathsf{MOTIV}_{\mathsf{DS}}$und $\mathsf{MOTIV}_{\mathsf{ES}}$ ausgab. Im Datensatz 3, bei dem $\mathsf{MOTIV}_{\mathsf{ES}}$ durch Mutationen abgeschwächt ist, fand MEME ausschließlich das stärkere, doppelsträngige Motiv, während die a priori-Erweiterung das schwächere $\mathsf{MOTIV}_{\mathsf{ES}}$ fand.

Datensatz 5, der nur $\mathsf{MOTIV}_{\mathsf{DS}}$-Instanzen enthält, wurde verwendet, um den Einfluss des Glättungskoeffizienten näher zu untersuchen. Da kein weiteres starkes Motiv in diesem Sequenzdatensatz auftritt, war die Frage zu beantworten, bei welcher Schwelle die a priori Verteilungen das Ergebnis so sehr dominieren, dass das einzige Motiv, dass eben doppelsträngig ist, zugunsten eines schwachen Pseudomotivs abgelehnt wird. Abbildung 7.1 zeigt die Ergebnisse dieses Versuchs. Bis zu einem Wert $\pi = 0.22$ gibt die MEME-Erweiterung schwächere, aber einzelsträngige Motive aus, bei Werten über 0.22 nähert sich die a priori Verteilung so sehr der Gleichverteilung an, dass $\mathsf{MOTIV}_{\mathsf{DS}}$als Ergebnis ausgegeben wird.

Bei Datensatz 6 bestand die Herausforderung an die Motivsuchealgorithmen darin, ein nur teilweise einzelsträngiges Motiv zu identifizieren. Während MEME jedoch als Motiv die ersten sechs Positionen (von denen 3 doppelsträngig sind) als Motiv ausgab, stieß die Erweiterung richtig auf die zentralen, aber doppelsträngigen Bereiche der Motivinstanzen.

Datensatz 7 und 8 wurden verwendet, um die beiden Motivsucheprogramme im ZOOPS- und TCM-Modus miteinander zu vergleichen. Es zeigte sich in beiden Fällen, dass auch hier die Verwendung der a priori Verteilungen dazu führte, dass etwas schwächere Motive ausgegeben werden, wenn deren Instanzen die gewünschte Einzelsträngigkeit besitzen, während das originale MEME stets das stärkere $\mathsf{MOTIV}_{\mathsf{DS}}$ ausgaben.

Biologische Datensätze. Nachdem sich die Verwendung von a priori Verteilungen bei den künstlichen Daten bewährt hatte, sollten diese auch auf echte Sequenzdaten angewendet werden.

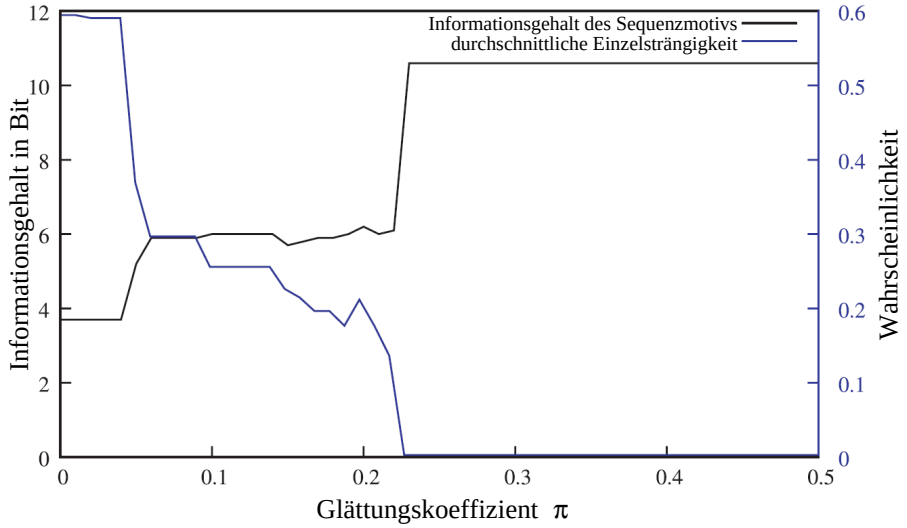


Abbildung 7.1: Einfluss des Glättungskoeffizienten π auf die Stärke des ausgegebenen Sequenzmotivs und auf die Einzelsträngigkeit der Motivinstanzen.

Zunächst wurden SELEX-Daten (siehe Unterabschnitt 2.3.4) für den neuronspezifischen Splicing-Faktor *Nova-1* mit Hilfe des erweiterten MEME untersucht. Die Sequenzen dieses Datensatzes enthalten 33 Bindungsstellen dieses Faktors, die in einzelsträngigen *Loops* liegen [Buc97]. Das originale MEME und die um a priori Verteilungen erweiterte Version wurden im TCM-Modus auf den Datensatz angewendet, wobei beide Programme mit dem Parameter $maxsites = 33$ ausgeführt wurden. Die erweiterte Version erkannte die 33 Motivinstanzen völlig korrekt, während das originale MEME zwar ein nahezu identisches Motiv als Ergebnis ausgab, dazu aber zwei mutmaßliche Motivinstanzen verwendete, die in doppelsträngigen Sequenzbereichen liegen.

Im 5′-UTR (siehe Unterabschnitt 2.1.2 auf Seite 12) von bestimmten Genen befinden sich konservierte Sequenzelemente, die bei mRNA-Sequenzen von regulierenden Proteinen gebunden werden. Häufig sind diese Elemente nicht nur auf Sequenzebene, sondern auch auf Strukturebene konserviert. Um die Anwendbarkeit der a priori Verteilungen bei der Suche dieser Elemente in einem größeren Sequenzbereich zu untersuchen, wurden in der Datenbank Pfam Sequenzen ausgewählt, die solche Sequenzelemente mit bekannter Struktur und bekannter Bindungssequenz enthalten.

Eines dieser Elemente ist das IRE (*iron responsive element*), dass bei der Expression von Genen benötigt wird, die mit dem Eisenhaushalt im Zusammenhang stehen [Hen96]. Das IRE besteht aus einer *stem-loop*-Struktur, wobei die einzelsträngige Schleife von regulierenden Proteinen gebunden wird. Unter Verwendung der a priori Erweiterung von MEME wird ein Motiv ausgegeben, dessen Instanzen in den Schleifen der IRE liegen. Das originale MEME liefert ein etwas stärkeres Motiv, dass jedoch gegenüber dem richtigen Bindungsmotiv um eine Position verschoben ist.

Ein weiteres Beispiel ist das Element PIE (*polyadenylation inhibition element*), dass Bindungsstellen für Proteine enthält. Die PIE bestehen aus einer *stem*-Struktur mit asymmetrischen internen Schleifen, in denen die Bindungsstellen dieser Proteine liegen. Während das originale MEME ein Motiv ausgab, dessen Instanzen in den doppelsträngigen *stems* zu finden sind, entdeckte die Erweiterung korrekt die Motive in den einzelsträngigen Schleifen (siehe Abbildung 7.2a)).

Eine ähnliche Verbesserung konnte bei dem TAR-Element in HIV-1-Viren festgestellt werden. Dieses Element besitzt eine Schleife, in der sich eine Bindungsstelle für ein regulierendes Heterodimer befindet. Die strukturbasierten a priori Verteilungen identifizierten diese Bindungsstellen korrekt, während das originale MEME ein stärkeres, jedoch doppelsträngiges Motiv fand (siehe Abbildung 7.2b)).

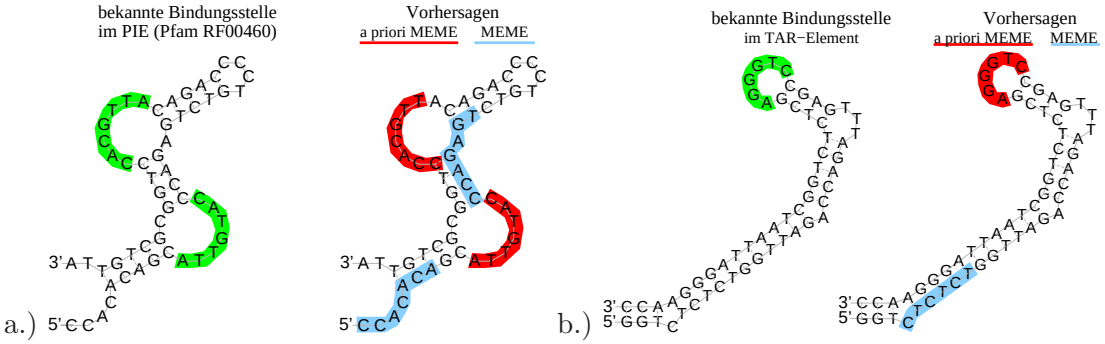


Abbildung 7.2: Schematische Darstellung von der RNA-Sekundärstrukturen in der Umgebung des PIE- und des TAR-Elements. Markiert sind jeweils das bekannte, einzelsträngige Motiv sowie die Teilsequenzen, die vom originalen MEME und der a priori Erweiterung ausgegeben wurden. In beiden Fällen bevorzugte das originale MEME ein konservierteres Sequenzmotiv.

7.4 Diskussion und Ausblick

In diesem Kapitel wurde ein Motivsucheverfahren entwickelt, dass auf dem bekannten EM-basierten MEME aufsetzt. Im Unterschied zu MEME können a priori Verteilungen über mögliche Startpositionen von Motivinstanzen verwendet werden, um die Motivsuche probabilistisch in Richtung vielversprechender Sequenzbereiche zu leiten. Auf diese Weise lässt sich zusätzliches biologisches Wissen bei der Motivsuche nutzen.

Es wurden zwei Anwendungen des Systems vorgeschlagen: 1.) Informationen über mögliche Nukleosomen in den Eingabesequenzen und 2.) Informationen über die Einzelsträngigkeit von RNA-Sequenzpositionen. In beiden Fällen konnte durch Verwendung der a priori Informationen die Motivsuche hinsichtlich der geforderten Eigenschaft (frei von Nukleosomen oder Einzelsträngigkeit) verbessert werden.

In den beiden Anwendungen der MEME-Erweiterung wurde jedoch recht ungesichertes Wissen verwendet. Die sequenzbasierte Vorhersage von Nukleosomen kann nur einen Teil der *in vivo* tatsächlich zu einem bestimmten Zeitpunkt auftretenden Nukleosome erklären. Die Vorhersage der RNA-Sekundärstruktur führt in vielen Fällen ebenfalls zu zweifelhaften Ergebnissen, vor allem, wenn es konkurrierende Strukturen mit minimaler Bindungsenergie gibt. In beiden Anwendungen sollte der a priori Verteilung nicht allzu großer Einfluss eingeräumt werden. Ein Anwender kann den Einfluss über den Glättungskoeffizienten π regulieren. Besitzt er großes Vertrauen in die Richtigkeit und Wichtigkeit der a priori Information, so ist ein kleiner π-Wert nahe liegend. Bei recht unsicherem Vorwissen kann der Einfluss mit einem großen π abgeschwächt werden. Da eine Wahl von π stark anwendungsabhängig ist, wird folgende Verfahrensweise bei der Verwendung der a priori Erweiterung von MEME vorgeschlagen:

1. Führe eine Motivsuche mit einem sehr großen π durch (z.B. $\pi = 10000$). Diese Motivsuche simuliert das originale MEME.
2. Zähle die gefundenen Motivinstanzen und untersuche bei ihnen die gewünschte a priori Eigenschaft.
3. Führe eine zweite Motivsuche mit einem niedrigen π durch, bei dem die minimale Anzahl der zu findenden Motivinstanzen festgelegt wird (MEME-Parameter *-minsites* und *-maxsites*).

Die entwickelte MEME-Erweiterung empfiehlt sich für weitere Anwendungen. So könnten im Bereich der TFBS-Motivsuche solche Positionen hervorgehoben werden, die evolutionär konserviert sind. Ein weiteres Beispiel wäre die Verwendung des Abstandes einer Position zur nächstgelegenen TSS.

Kapitel 8

Zusammenfassung

Diese Dissertation beschäftigte sich mit der stochastischen Modellierung und Vorhersage von regulativen DNA-Sequenzen. Dabei handelt es sich um Bindungsstellen für Transkriptionsfaktoren, die meist im Bereich des 5′-Endes von proteinkodierenden Genen liegen und die Transkription dieser Gene beeinflussen, beispielsweise durch Wechselwirkungen mit dem RNA-PolymeraseII-Komplex. Das Aufdecken dieser regulativen DNA-Sequenzen erlaubt Rückschlüsse darauf, in welchen Situationen ein Gen aktiviert oder deaktiviert ist. Daraus ergeben sich wiederum Folgerungen für die Funktion eines Gens.

Die Modellierung und Vorhersage von regulativen DNA-Sequenzen ist ein Forschungszweig der Bioinformatik. In dieser Dissertation wurden drei verschiedene Aspekte dieses Forschungszweigs bearbeitet: 1.) die Modellierung einzelner Transkriptionsfaktorbindungsstellen (TFBS), 2.) die gemeinsame Modellierung funktionell zusammengehörender TFBS und 3.) das unüberwachte Lernen von Sequenzmotiven. In allen drei Bereichen wurden Modelle und Verfahren entwickelt, die sich von bisherigen Standardlösungen dadurch unterscheiden, dass sie neben der Sequenzinformation zusätzliches biologisches Wissen berücksichtigen können. Ziel war es in allen drei Bereichen, über adäquatere Beschreibungen der regulativen DNA-Sequenzen die für diese Erkennungsausgabe allgemein hohen Klassifikationsfehlerraten zu verringern. In allen drei Bereichen konnte dieses Ziel erreicht werden.

8.1 Modellierung von Transkriptionsfaktorbindungsstellen

Die Modellierung einzelner TFBS konzentrierte sich bisher meist auf die positionsweise Sequenzähnlichkeit von TFBS eines Transkriptionsfaktors. Bekannte Vertreter, wie Consensussequenzen oder PWM-Modelle, sind einfache, fehlertolerante Repräsentationen eines Sequenzmotivs. Problematisch ist bei ihrer Verwendung die hohe Zahl an Falsch-Positiv-Vorhersagen.

Der in dieser Arbeit vorgestellte Modellierungsansatz ermöglicht eine flexiblere Beschreibung der charakteristischen Eigenschaften von TFBS eines Transkriptionsfaktors und berücksichtigt zusätzlich statistische Zusammenhänge zwischen diesen Eigenschaften. Beispiele für solche Eigenschaften sind sequenzabhängige lokale Schwankungen von

strukturellen Eigenschaften der DNA, Anfangspositionen von Treffern kurzer Muster innerhalb der TFBS oder Überrepräsentationen von Nukleotiden in einer Umgebung der TFBS.

Individuelle TFBS wurden in dieser Arbeit nicht mehr als Zeichenkette über dem Alphabet der DNA-Nukleotide aufgefasst, sondern allgemeiner als Merkmalsvektor einer Menge von Merkmalen. Über Merkmalsauswahlverfahren wurden Teilmengen möglicher Merkmale ausgewählt, die besonders gut zwischen TFBS und anderen DNA-Sequenzen diskriminieren können. Die Merkmale einer solchen Teilmenge wurden als diskrete Zufallsvariablen in BN-Klassifikatoren für dieses Zwei-Klassenproblem modelliert.

Der Modellierungsansatz wurde für 86 Sequenzdatensätze über einen Vergleich mit PWM-Modellen evaluiert. In mehr zwei Dritteln der Versuche konnten mit BN-Klassifikatoren bessere Ergebnisse erzielt werden als mit PWM-Modellen. Mit der Webanwendung *BioBayesNet* wurde der Modellierungsansatz auf beliebige Klassifikationsaufgaben der biologischen Sequenzanalyse erweitert und der Forschungsgemeinde frei zugänglich gemacht.

Trotz weitreichender Verbesserungen kann das Problem der TFBS-Modellierung nicht als gelöst bezeichnet werden. Auch BN-Klassifikatoren finden weit mehr mutmaßliche TFBS, als biologisch relevant sein können. Der gegenwärtige Forschungsstand lässt jedoch für das Problem der rechnergestützten Modellierung einzelner TFBS in naher Zukunft keine nennenswerten Verbesserungen erhoffen, vor allem bei alleiniger Verwendung von Sequenzinformationen. Beinahe jedes Modellierungsparadigma wurde bereits auf die TFBS-Vorhersage angewendet, und mit marginalen Erfolgen veröffentlicht. Gerade in den letzten Jahren ist zu beobachten, dass neue Veröffentlichungen meist Modelle vorstellen, die eine Kombination bisher erfolgreicher Ansätze darstellen oder bisherige Modelle durch Optimierungen kleiner Details zu verbessern versuchen. Die Klassifikationsleistungen dieser Verfahren gegenüber dem Status-Quo sind jedoch meist zu gering, um ihren Einsatz anstelle von Gewichtsmatrizen zu rechtfertigen. Zusätzliche Daten, beispielsweise von komplexen dreidimensionalen Analysen der DNA-Protein-Bindung oder des Chromatins sowie Expressionsdaten der Transkriptionsfaktoren versprechen ein tieferes Verständnis der transkriptionellen Genregulation im Allgemeinen und dem DNA-Protein-Bindungsprozess im Speziellen. Solche Daten liegen im erforderlichen Umfang derzeit nicht vor. Die hier entwickelten BN-Klassifikatoren besitzen den Vorteil, dass relativ einfach neue Daten in die Modellierung einbezogen werden können.

Neben der wünschenswerten Verbesserung der Datenlage sind auch auf Modellierungsseite Weiterentwicklungen denkbar. Nach Einschätzung des Autors liegt ein Schlüssel für weitere Verbesserungen in der direkten Verwendung kontinuierlicher Merkmale anstatt einer vorherigen Diskretisierung. Diesbezügliche Bayessche Netze existieren. Ihre Lernalgorithmen sind jedoch komplexer als für diskrete Bayessche Netze. Des weiteren können zusätzliche Merkmalsklassen entwickelt werden, die neue Informationsquellen abbilden können. Beispielsweise ließen sich die im dritten Teil der Arbeit verwendeten Nukleosomvorhersagen als Merkmale in BN-Klassifikatoren einsetzen.

8.2 Modellierung von TFBS-Modulen

Im zweiten Teil wurde der Tatsache Rechnung getragen, dass TFBS aufgrund kooperierender Faktoren oft gehäuft auf der DNA auftreten. Die gemeinsame Modellierung von TFBS-Modulen, d.h. einer Gruppe funktionell zusammenhängender TFBS, verspricht, unter den unzähligen Einzel-TFBS-Vorhersagen jene hervorzuheben, die in einem Umfeld auftreten, dass biologisch plausibel ist.

In dieser Dissertation wurden zwei Modellierungsansätze für solche TFBS-Module entwickelt. Die Modelle beider Ansätze haben gemeinsam, dass sie sich in modularer Weise aus stochastischen Modellen einzelner TFBS zusammensetzen. Sie unterscheiden sich darin, wie der Zusammenhang zwischen kooperierenden TFBS hergestellt wird.

Der erste Ansatz ist eine Weiterentwicklung eines existierenden Modellierungsansatzes, bei dem einfache TFBS-Sequenzmodelle in ein Hidden Markov Modell (HMM) integriert werden. Das HMM modelliert den stochastischen Prozess der Erzeugung einer gesamten DNA-Region. Dieses Vorgehen wurde in der Vergangenheit erfolgreich unter Verwendung von PWM-Modellen eingesetzt. In dieser Arbeit wurde ein Ansatz vorgestellt, der die im ersten Teil entwickelten BN-Klassifikatoren in einem HMM für TFBS-Module verwendet. Die Grundidee bestand darin, Bayessche Netze für TFBS oder nicht funktionellen Sequenzen als Ausgabeverteilungen von HMM-Zuständen einzusetzen. Ein solches HMM erzeugt eine Sequenz von Merkmalsvektoren.

Die Kombination von Bayesschen Netzen und Hidden-Markov-Modellen ist zum gegenwärtigen Zeitpunkt neu in der Bioinformatik. Obwohl erste Ergebnisse nur marginale Verbesserungen gegenüber den bisherigen Modul-HMM zeigen, empfiehlt sich diese Modellierung für die Verwendung in weiteren Bereichen der Bioinformatik, etwa für multiple Alignments mit Profile-HMM oder zur Erkennung von alternativen Splicestellen.

Der zweite Ansatz zur Modellierung von TFBS-Modulen bestand in einer positionsweisen a priori Verteilung über alle berücksichtigten Einzel-TFBS-Modelle. Diese a priori Verteilungen überlagern die Vorhersagenn der TFBS-Modelle und wirken als Filter, der TFBS-Vorhersagen bestraft, wenn sie in einem ungünstigen Kontext auftreten und TFBS-Vorhersagen verstärkt, wenn der Kontext Hinweise auf die Relevanz einer Vorhersage bietet. Zur Berechnung der a priori Verteilungen werden für jedes TFBS-Modell Bedingungen in Form aussagenlogischer Ausdrücke definiert, die erfüllt sein müssen, damit eine Vorhersage biologisch relevant sein kann. Solche Bedingungen betreffen die Lage und Orientierung von notwendigen TFBS kooperierender Faktoren oder Sequenzannotationen, die Informationen über Gewebespezifität oder Transkriptionsstartstellen des regulierten Gens enthalten können. Die Auswertung solcher aussagenlogischen Ausdrücke wurden in speziell konstruierten Bayesschen Netzen probabilistisch umgesetzt. Diese Netze erzeugen für jede Sequenzposition die gewünschte a priori Verteilung.

Bei der Evaluierung dieses Ansatzes anhand künstlicher Sequenzdaten und einem genomischen Datensatz konnten starke Verbesserungen gegenüber der isolierten Suche nach TFBS gezeigt werden.

8.3 Verwendung von a priori Wissen bei der EM-basierten Motivsuche

Der dritte Teil der Arbeit widmete sich der Motivsuche, dem Aufspüren ähnlicher Teilsequenzen in einer Menge von Eingabesequenzen. In der Musteranalyse entspricht diese Aufgabe dem unüberwachten Lernen.

In dieser Arbeit wurde eine Erweiterung für den bekannten Motivsuche-Algorithmus MEME entwickelt, die es ermöglicht, zusätzliches Wissen bei der Suche nach Motivinstanzen einzusetzen. Das zusätzliche Wissen wurde über Bewertungen der einzelnen Positionen der Eingabe eingebracht, die ausdrücken, wie wahrscheinlich eine Position Startpunkt einer Instanz des gesuchten Sequenzmotivs ist (unabhängig von der Nukleotidfolge an dieser Stelle). Mit Hilfe dieser positionsspezifischen Bewertungen wurden a priori Verteilungen über mögliche Startpositionen entwickelt. Diese Verteilungen wurden in dem EM-Algorithmus für die Motivsuche verwendet. Dazu mussten die probabilistischen Modelle von MEME erweitert werden, um a priori Verteilungen berücksichtigen zu können.

Es wurden zwei Anwendungen vorgestellt, in denen erfolgreich a priori Wissen eingesetzt werden konnte. Die erste Anwendung verwendet ein mathematisches Modell, um allein aus Sequenzinformationen Bereiche vorherzusagen, an denen sich Nukleosome befinden. Die Intention dahinter ist, dass Transkriptionsfaktoren durch das Vorhandensein von Histonkomplexen daran gehindert werden können, an Sequenzbereichen eines Nukleosoms zu binden. Die von den Vorhersagen abgeleiteten a priori Verteilungen weisen nukleosombesetzen Positionen niedrige Wahrscheinlichkeiten dafür zu, Startpunkt einer Motivinstanz zu sein. Dieses Zusatzwissen wurde auf 105 *ChIP on chip*-Datensätze aus der Hefe angewendet. Gegenüber dem originalen MEME konnten Verbesserungen erreicht werden. Es war zu beobachten, dass die durchschnittliche Wahrscheinlichkeit der Motivinstanzen, Teil eines Nukleosoms zu sein, niedriger ist als für die Motivinstanzen, die das originale MEME fand. Zudem stimmte das gefundene Sequenzmotiv bei Verwendung des a priori Wissens um 10% häufiger mit dem aus der Literatur bekannten Motiv überein als das bei MEME der Fall war.

Die zweite Anwendung beschäftigte sich mit der Suche nach Sequenzmotiven für Splicefaktoren auf RNA-Sequenzen. Für eine Reihe dieser Faktoren ist bekannt, dass sie bevorzugt einzelsträngige RNA-Sequenzen binden. Mit Hilfe von Algorithmen zur RNA-Strukturvorhersage wurden aus diesem Grund a priori Verteilungen abgeleitet, welche die Einzelsträngigkeit von Sequenzpositionen quantifizieren. In verschiedenen Versuchen mit künstlichen Datensätzen konnte demonstriert werden, dass bei Verwendung dieser a priori Verteilungen trotz Vorhandensein eines starken Sequenzmotivs bevorzugt schwächere Sequenzmotive ausgegeben werden, wenn diese die wichtige Eigenschaft der Einzelsträngigkeit besitzen. Das originale MEME wählte hingegen stets das stärkere, aber doppelsträngige Sequenzmotiv. Diese Beobachtung konnte auch für alle untersuchten SELEX-Datensätze gemacht werden.

Es gibt eine Reihe weiterer Möglichkeiten, zusätzliches Wissen in der hier vorgestellten MEME-Erweiterung zu berücksichtigen. Im Bereich der TFBS-Motivsuche könnte beispielsweise eine komparative Analyse der durchsuchten Sequenzen mit orthologen Sequenzen betrachtet werden. Eine weitere Möglichkeit ist die Verwendung des Abstandes einer Position zu einer nachfolgenden TSS, da viele Transkriptionsfaktoren bevorzugt dort binden.

Literaturverzeichnis

[Aer03] Aerts, S.; Van Loo, P.; Thijs, G.; Moreau, Y.; De Moor, B.: *Computational detection of cis -regulatory modules*, *Bioinformatics*, Bd. 19 Suppl 2, 2003, S. II5–II14.

[Agr94] Agrawal, R.; Srikant, R.: *Fast Algorithms for Mining Association Rules*, in *Proceedings of the 20th VLDB Conference*, 1994, S. 1–13.

[Ahm04] Ahmad, S.; Gromiha, M. M.; Sarai, A.: *Analysis and prediction of DNA-binding proteins and their binding residues based on composition, sequence and structural information*, *Bioinformatics*, Bd. 20, Nr. 4, 2004, S. 477–486.

[Alb02] Alberts, B.; Johnson, A.; Lewis, J.; Raff, M.; Roberts, K.; Walter, P.: *Molecular Biology of the Cell*, Garland Science, New York, 4. Ausg., 2002.

[Alv03] Alvarez, M.; Rhodes, S. J.; Bidwell, J. P.: *Context-dependent transcription: all politics is local*, *Gene*, Bd. 313, 2003, S. 43–57.

[Ant93] Antequera, F.; Bird, A.: *Number of CpG islands and genes in human and mouse*, *Proc. Natl. Acad. Sci.*, Bd. 90, 1993, S. 11995–11999.

[Arn97] Arnone, M. I.; Davidson, E. H.: *The hardwiring of development: organization and function of genomic regulatory systems*, *Development*, Bd. 124, Nr. 10, 1997, S. 1851–1864.

[Arn02] Arnosti, D. N.: *Design and function of transcriptional switches in Drosophila*, *Insect Biochem Mol Biol*, Bd. 32, Nr. 10, 2002, S. 1257–1273.

[Arn03] Arnosti, D. N.: *Analysis and function of transcriptional regulatory elements: insights from Drosophila*, *Annual review of entomology*, Bd. 48, 2003, S. 579–602.

[Ash04] Ashmore, D. C.: *The J2EE Architect's Handbook: How to be a successful technical architect for J2EE applications*, DVT Press, Lombard, IL, 2004.

[Bai95a] Bailey, T. L.: *Discovering motifs in DNA and protein sequences: The approximate common substring problem*, PhD thesis, University of California, San Diego, 1995.

[Bai95b] Bailey, T. L.; Elkan, C.: *Unsupervised Learning of Multiple Motifs in Biopolymers using EM*, *Machine Learning*, Bd. 21, Nr. 1-2, 1995, S. 51–80.

[Bai03] Bailey, T. L.; Noble, W. S.: *Searching for statistically significant regulatory modules*, *Bioinformatics*, Bd. 19 Suppl 2, 2003, S. II16–II25.

[Bal98] Baldi, P.; Chauvin, Y.; Brunak, S.; Gorodkin, J.; Pedersen, A. G.: *Computational applications of DNA structural scales*, in *Proc. ISMB 1998*, Bd. 6, 1998, S. 35–42.

[Bar03] Barash, Y.; Elidan, G.; Friedman, N.; Kaplan, T.: *Modeling Dependencies in Protein-DNA Binding Sites*, in *Research in Computational Molecular Biology, 7th Annual International Conference*, 2003, S. 28–37.

[Bar04] Barash, Y.; Elidan, G.; Kaplan, T.; Friedman, N.: *CIS: Compound Importance Sampling Method for Protein-DNA Binding Site p-value Estimation*, in *ECCB*, 2004, S. 0–0, short-paper.

[Bea97] Beato, M.; Eisfeld, K.: *Transcription factor access to chromatin*, *Nucleic Acids Research*, Bd. 25, Nr. 18, 1997, S. 3559–3563.

[Bec06] Beckstette, M.; Homann, R.; Giegerich, R.; Kurtz, S.: *Fast index based algorithms and software for matching position specific scoring matrices*, *BMC Bioinformatics*, Bd. 7, 2006, S. 389.

[Bec07] Beckstette, M.: *Index-based algorithms for motif search and their integration in a system for differential genome analysis*, PhD thesis, University of Bielefeld, 2007.

[Ben02a] Benos, P. V.; Bulyk, M. L.; Stormo, G. D.: *Additivity in protein-DNA interactions: how good an approximation is it?*, *Nucleic Acids Research*, Bd. 30, Nr. 20, 2002, S. 4442–4451.

[Ben02b] Benos, P. V.; Lapedes, A. S.; Stormo, G. D.: *Probabilistic code for DNA recognition by proteins of the EGR family*, *Journal of Molecular Biology*, Bd. 323, Nr. 4, 2002, S. 701–727.

[Ber02] Berman, B. P.; Nibu, Y.; Pfeiffer, B. D.; Tomancak, P.; Celniker, S. E.; Levine, M.; Rubin, G. M.; Eisen, M. B.: *Exploiting transcription factor binding site clustering to identify cis-regulatory modules involved in pattern formation in the Drosophila genome*, *Proc. Natl. Acad. Sci. USA*, Bd. 99, Nr. 2, 2002, S. 757–762.

[BG05] Ben-Gal, I.; Shani, A.; Gohr, A.; Grau, J.; Arviv, S.; Shmilovici, A.; Posch, S.; Grosse, I.: *Identification of transcription factor binding sites with variable-order Bayesian networks*, *Bioinformatics*, Bd. 21, Nr. 11, 2005, S. 2657–66.

[Bil05] Bilu, Y.; Barkai, N.: *The design of transcription-factor binding sites is affected by combinatorial regulation*, *Genome Biol*, Bd. 6, Nr. 12, 2005, S. R103.

[Bla98] Blackwood, E. M.; Kadonaga, J. T.: *Going the distance: a current view of enhancer action*, *Science*, Bd. 281, Nr. 5373, 1998, S. 60–63.

[Bla06] Blanco, E.; Messeguer, X.; Smith, T. F.; Guigo, R.: *Transcription factor map alignment of promoter regions*, *PLoS Comput Biol*, Bd. 2, Nr. 5, 2006, S. e49.

[Boy04] Boys, R. J.; Henderson, D. A.: *A Bayesian approach to DNA sequence segmentation (with discussion)*, *Biometrics*, Bd. 60, 2004, S. 573–588.

[Boy05] Boyer, L. A.; Lee, T. I.; Cole, M. F.; Johnstone, S. E.; Levine, S. S.; Zucker, J. P.; Guenther, M. G.; Kumar, R. M.; Murray, H. L.; Jenner, R. G.; Gifford, D. K.; Melton, D. A.; Jaenisch, R.; Young, R. A.: *Core transcriptional regulatory circuitry in human embryonic stem cells*, *Cell*, Bd. 122, Nr. 6, 2005, S. 947–956.

[Bra98] Brazma, A.; Jonassen, I.; Eidhammer, I.; Gilbert, D.: *Approaches to the automatic discovery of patterns in biosequences*, *Journal of Computational Biology*, Bd. 5, Nr. 2, 1998, S. 279–305.

[Bra04] Bray, N.; Pachter, L.: *MAVID: constrained ancestral alignment of multiple sequences*, *Genome Res*, Bd. 14, Nr. 4, 2004, S. 693–699.

[Bre02] Brett, D.; Pospisil, H.; Valcarcel, J.; Reich, J.; Bork, P.: *Alternative splicing and genome complexity*, *Nat Genet*, Bd. 30, Nr. 1, 2002, S. 29–30.

[Bru95] Brukner, I.; Sanchez, R.; Suck, D.; Pongor, S.: *Sequence-dependent bending propensity of DNA as revealed by DNase I: parameters for trinucleotides*, *EMBO J*, Bd. 14, Nr. 8, 1995, S. 1812–1818.

[Buc94] Bucher, P.; Bairoch, A.: *A generalized profile syntax for biomolecular sequence motifs and its function in automatic sequence interpretation*, in *ISMB94*, Bd. 2, 1994, S. 53–61.

[Buc97] Buckanovich, R. J.; Darnell, R. B.: *The neuronal RNA binding protein Nova-1 recognizes specific RNA targets in vitro and in vivo*, *Mol Cell Biol*, Bd. 17, Nr. 6, 1997, S. 3194–3201.

[Bul02] Bulyk, M. L.; Johnson, P. L. F.; Church, G. M.: *Nucleotides of transcription factor binding sites exert interdependent effects on the binding affinities of transcription factors*, *Nucleic Acids Research*, Bd. 30, Nr. 5, 2002, S. 1255–1261.

[Bus00] Bussemaker, H. J.; Li, H.; Siggia, E. D.: *Building a dictionary for genomes: identification of presumptive regulatory sites by statistical analysis*, *Proc. Natl. Acad. Sci. USA*, Bd. 97, Nr. 18, 2000, S. 10096–10100.

[Cai00] Cai, D.; Delcher, A.; Kao, B.; Kasif, S.: *Modeling splice sites with Bayes networks*, *Bioinformatics*, Bd. 16, Nr. 2, 2000, S. 152–158.

[Cav87] Cavener, D. R.: *Comparison of the consensus sequence flanking translational start sites in Drosophila and vertebrates*, *Nucleic Acids Research*, Bd. 15, Nr. 4, 1987, S. 1353–1361.

[Che03] Chenna, R.; Sugawara, H.; Koike, T.; Lopez, R.; Gibson, T. J.; Higgins, D. G.; Thompson, J. D.: *Multiple sequence alignment with the Clustal series of programs*, *Nucleic Acids Research*, Bd. 31, Nr. 13, 2003, S. 3497–3500.

[Chi96] Chickering, D.: *Learning Bayesian Networks is NP-complete*, Kap. 3, Springer, New York, 1996, S. 125–150.

[Cho68] Chow, C. K.; Liu, C. N.: *Approximating discrete probability distributions with dependence drees*, *IEEE Transactions on Information Theory*, Bd. 14, 1968, S. 462–467.

[Cho93] Choy, B.; Green, M. R.: *Eukaryotic activators function during multiple steps of preinitiation complex assembly*, *Nature*, Bd. 366, Nr. 6455, 1993, S. 531–536.

[Cla96] Claverie, J. M.; Audic, S.: *The statistical significance of nucleotide position-weight matrix matches*, *Comput Appl Biosci*, Bd. 12, Nr. 5, 1996, S. 431–439.

[Coo90] Cooper, G.: *The Computational Complexity of Probabilistic Inference Using Bayesian Networks*, *Artificial Intelligence*, Bd. 33, 1990.

[Coo92] Cooper, G. F.; Herskovits, E.: *A Bayesian Method for the Induction of Probabilistic Networks from Data*, *Machine Learning*, Bd. 9, 1992.

[Cro97] Crowley, E. M.; Roeder, K.; Bina, M.: *A statistical model for locating regulatory regions in genomic DNA*, *Journal of Molecular Biology*, Bd. 268, Nr. 1, 1997, S. 8–14.

[Das97] Dash, M.; Liu, H.: *Feature Selection for Classification*, *Intelligent Data Analysis*, Bd. 1, Nr. 3, 1997, S. 131–156.

[Day92] Day, W. H.; McMorris, F. R.: *Critical comparison of consensus methods for molecular sequences*, *Nucleic Acids Research*, Bd. 20, Nr. 5, 1992, S. 1093–1099.

[Dec99] Dechter, R.: *Bucket Elimination: A Unifying Framework for Reasoning*, *Artificial Intelligence*, Bd. 113, Nr. 1-2, 1999, S. 41–85.

[Dem77] Dempster, A.; Laird, N.; Rubin, D.: *Maximum-likelihood from incomplete data via the EM algorithm*, *Journal of Royal Statistical Society*, Bd. 39, 1977, S. 1 – 38.

[Dic92] Dickerson, R. E.: *DNA structure from A to Z*, *Methods Enzymol*, Bd. 211, 1992, S. 67–111.

[Djo03] Djordjevic, M.; Sengupta, A. M.; Shraiman, B. I.: *A biophysical approach to transcription factor binding site discovery*, *Genome Research*, Bd. 13, Nr. 11, 2003, S. 2381–2390.

[Dre81] Drew, H. R.; Wing, R. M.; Takano, T.; Broka, C.; Tanaka, S.; Itakura, K.; Dickerson, R.: *Structure of a B-DNA dodecamer: conformation and dynamics*, in *Natn. Acad. Sci. USA*, 1981, S. 2179–2183.

[Dro04] Dror, G.; Sorek, R.; Shamir, R.: *Accurate identification of alternatively spliced exons using support vector machine*, *Bioinformatics*, Bd. 21, Nr. 7, 2004, S. 897–901.

[Dur98] Durbin, R.; Eddy, S.; Krogh, A.; Mitchison, G.: *Biological sequence analysis - Probabilistic models of proteins and nucleic acids*, Cambridge University Press, Cambridge, UK, 1998.

[EH97] El Hassan, M. A.; Calladine, C. R.: *Conformational characteristics of DNA: Empirical classifications and a hypothesis for the conformational behaviour of dinucleotide steps*, *Phil. Trans. R. Soc. Lond. A*, Bd. 355, 1997, S. 43–100.

[Ell92] Ellington, A. D.; Szostak, J. W.: *Selection in vitro of single-stranded DNA molecules that fold into specific ligand-binding structures*, *Nature*, Bd. 355, Nr. 6363, 1992, S. 850–852.

[Esk02] Eskin, E.; Pevzner, P. A.: *Finding composite regulatory patterns in DNA sequences*, *Bioinformatics*, Bd. 18 Suppl 1, 2002, S. S354–S363.

[Eva88] Evans, R. M.; Hollenberg, S. M.: *Zinc fingers: gilt by association*, *Cell*, Bd. 52, 1988, S. 1–3.

[Fay93] Fayyad, U. M.; Irani, K. B.: *Multi-interval discretization of continuousvalued attributes for classification learning*, in *IJCAI-93*, Bd. 2, 1993.

[Fri97] Friedman, N.; Geiger, D.; Goldszmidt, M.: *Bayesian Network Classifiers*, *Machine Learning*, Bd. 29, Nr. 2-3, 1997, S. 131–163.

[Fri01] Frith, M. C.; Hansen, U.; Weng, Z.: *Detection of cis-element clusters in higher eukaryotic DNA*, *Bioinformatics*, Bd. 17, Nr. 10, 2001, S. 878–889.

[Fri02] Frith, M. C.; Spouge, J. L.; Hansen, U.; Weng, Z.: *Statistical significance of clusters of motifs represented by position specific scoring matrices in nucleotide sequences*, *Nucleic Acids Research*, Bd. 30, Nr. 14, 2002, S. 3214–3224.

[Geo06] Georgi, B.; Schliep, A.: *Context-specific independence mixture modeling for positional weight matrices*, *Bioinformatics*, Bd. 22, Nr. 14, 2006, S. e166–e173.

[Ger05] Gershenzon, N. I.; Stormo, G. D.; Ioshikhes, I. P.: *Computational technique for improvement of the position-weight matrices for the DNA/protein binding sites*, *Nucleic Acids Research*, Bd. 33, Nr. 7, 2005, S. 2290–2301.

[GG87a] Gardiner-Garden, M.; Frommer, M.: *CpG islands in vertebrate genomes*, *Journal of Molecular Biology*, Bd. 196, 1987, S. 261–282.

[GG87b] Gardiner-Garden, M.; Frommer, M.: *CpG islands in vertebrate genomes*, *Journal of Molecular Biology*, Bd. 196, 1987, S. 261–282.

[Gol94] Goldstein, L.; Waterman, M. S.: *Approximations to profile score distributions*, *Journal of Computational Biology*, Bd. 1, Nr. 2, 1994, S. 93–104.

[Gru96] Grundy, W. N.; Bailey, T. L.; Elkan, C. P.: *ParaMEME: a parallel implementation and a web interface for a DNA and protein motif discovery tool*, *Comput Appl Biosci*, Bd. 12, Nr. 4, 1996, S. 303–310.

[Gua88] Guarente, L.: *USAs and Enhancers: common mechanisms of transcriptional activation in yeast and mammals*, *Cell*, Bd. 52, 1988, S. 303–305.

[Guh01] GuhaThakurta, D.; Stormo, G. D.: *Identifying target sites for cooperatively binding factors*, *Bioinformatics*, Bd. 17, Nr. 7, 2001, S. 608–621.

[Han05] Hannenhalli, S.; Wang, L.-S.: *Enhanced position weight matrices using mixture models*, *Bioinformatics*, Bd. 21 Suppl 1, 2005, S. i204–i212.

[Har04] 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.; Young, R. A.: *Transcriptional regulatory code of a eukaryotic genome*, *Nature*, Bd. 431, Nr. 7004, 2004, S. 99–104.

[Heg07] Heger, A.; Ponting, C. P.: *Variable strength of translational selection among 12 Drosophila species*, *Genetics*, Bd. 177, Nr. 3, 2007, S. 1337–1348.

[Hen88] Henricon, M.: *Propagating Uncertainty in Bayesian Networks by Logic Sampling*, in Lemmer, J. F.; Kanal, L. N. (Hrsg.): *Uncertainty in Artificial Intelligence; Proceedings of the Second Conference*, Amsterdam, NL, 1988.

[Hen96] Hentze, M. W.; Kuhn, L. C.: *Molecular control of vertebrate iron metabolism: mRNA-based regulatory circuits operated by iron, nitric oxide, and oxidative stress*, *Proc. Natl. Acad. Sci. USA*, Bd. 93, Nr. 16, 1996, S. 8175–8182.

[Her99] Hertz, G. Z.; Stormo, G. D.: *Identifying DNA and protein patterns with statistically significant alignments of multiple sequences*, *Bioinformatics*, Bd. 15, Nr. 7-8, 1999, S. 563–577.

[Her05] Hertzberg, L.; Zuk, O.; Getz, G.; Domany, E.: *Finding motifs in promoter regions*, *Journal of Computational Biology*, Bd. 12, Nr. 3, 2005, S. 314–330.

[Hil06a] Hiller, M.; Pudimat, R.; Busch, A.; Backofen, R.: *Using RNA secondary structures to guide sequence motif finding towards single-stranded regions*, *Nucleic Acids Research*, Bd. 34, Nr. 17, 2006, S. e117.

[Hil06b] Hiller, M.: *Bioinformatics Analyses of Alternative Splicing: Non-EST based Prediction, Influence of Secondary Structures and Tandem Splice Sites*, PhD thesis, Albert-Ludwigs University Freiburg, December 2006.

[Hof06] Hofacker, I. L.; Stadler, P. F.: *Memory efficient folding algorithms for circular RNA secondary structures*, *Bioinformatics*, Bd. 22, Nr. 10, 2006, S. 1172–1176.

[Hog87] Hogan, M. E.; Austin, R. H.: *Importance of DNA stiffness in protein-DNA binding specificity*, *Nature*, Bd. 329, 1987, S. 263–266.

[Hor92] Horikoshi, M.; Bertuccioli, C.; Takada, R.; Wang, J.; Yamamoto, T.; Roeder, R. G.: *Transcription factor TFIID induces DNA bending upon binding to the TATA element*, *Proc. Natl. Acad. Sci. USA*, Bd. 89, Nr. 3, 1992, S. 1060–1064.

[Hua96] Huang, C.; Darwiche, A.: *Inference in belief networks: A procedural guide*, *International Journal of Approximate Reasoning*, Bd. 15, Nr. 3, 1996, S. 225–263.

[Jen90] Jensen, F. V.; Lauritzen, S. L.; Olesen, K. G.: *Bayesian Updating in Causal Probabilistic Networks by Local Computation*, *Computational Statistical Quarterly*, Bd. 4, 1990.

[Joh03] Johansson, O.; Alkema, W.; Wasserman, W. W.; Lagergren, J.: *Identification of functional clusters of transcription factor binding motifs in genome sequences: the MSCAN algorithm*, *Bioinformatics*, Bd. 19 Suppl 1, 2003, S. I169–I176.

[Jol05] Jolly, E.; Chin, C. S.; Herskowitz, I.; Li, H.: *Genome-wide identification of the regulatory targets of a transcription factor using biochemical characterization and computational genomic analysis*, *BMC Bioinformatics*, Bd. 6, Nr. 1, 2005, S. 275.

[Jon95] Jonassen, I.; Collins, J. F.; Higgins, D. G.: *Finding flexible patterns in unaligned protein sequences*, *Protein Sci*, Bd. 4, Nr. 8, 1995, S. 1587–1595.

[Kar90] Karlin, S.; Altschul, S. F.: *Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes*, *Proc. Natl. Acad. Sci. USA*, Bd. 87, Nr. 6, 1990, S. 2264–2268.

[Kar96] Karas, H.; Knuppel, R.; Schulz, W.; Sklenar, H.; Wingender, E.: *Combining structural analysis of DNA with search routines for the detection of transcription regulatory elements*, *Comput Appl Biosci*, Bd. 12, Nr. 5, 1996, S. 441–446.

[Kas01] Kask, K.; Dechter, R.; Larrosa, J.; Cozman, F.: *Bucket-Tree Elimination for Automated Reasoning*, 2001.

[Kei02] Keich, U.; Pevzner, P. A.: *Finding motifs in the twilight zone*, *Bioinformatics*, Bd. 18, Nr. 10, 2002, S. 1374–1381.

[Kel03] Kel, A. E.; Gossling, E.; Reuter, I.; Cheremushkin, E.; Kel-Margoulis, O. V.; Wingender, E.: *MATCH: A tool for searching transcription factor binding sites in DNA sequences*, *Nucleic Acids Research*, Bd. 31, Nr. 13, 2003, S. 3576–3579.

[Kel06] Kel, A.; Konovalova, T.; Waleev, T.; Cheremushkin, E.; Kel-Margoulis, O.; Wingender, E.: *Composite Module Analyst: a fitness-based tool for identification of transcription factor binding site combinations*, *Bioinformatics*, Bd. 22, Nr. 10, 2006, S. 1190–1197.

[Kie01] Kielbasa, S. M.; Korbel, J. O.; Beule, D.; Schuchhardt, J.; Herzel, H.: *Combining frequency and positional information to predict transcription factor binding sites*, *Bioinformatics*, Bd. 17, Nr. 11, 2001, S. 1019–1026.

[Kim94] Kim, J. L.; Burley, S. K.: *1.9 A resolution refined structure of TBP recognizing the minor groove of TATAAAAG*, *Nat Struct Biol*, Bd. 1, Nr. 9, 1994, S. 638–653.

[Kit78] Kittler, J.: *Feature set search algorithms*, *Pattern Recognition and Signal Processing*, Bd. 5, 1978, S. 41–60.

[Kli99] Klingenhoff, A.; Frech, K.; Quandt, K.; Werner, T.: *Functional promoter modules can be detected by formal models independent of overall nucleotide sequence similarity*, *Bioinformatics*, Bd. 15, Nr. 3, 1999, S. 180–186.

[Klu94] Klug, S.; Famulok, M.: *All you wanted to know about SELEX*, *Mol Biol Rep*, Bd. 20, Nr. 2, 1994, S. 97–107.

[Kol02] Kolchanov, N. A.; Ignatieva, E. V.; Ananko, E. A.; Podkolodnaya, O. A.; Stepanenko, I. L.; Merkulova, T. I.; Pozdnyakov, M. A.; Podkolodny, N. L.; Naumochkin, A. N.; Romashchenko, A. G.: *Transcription Regulatory Regions Database (TRRD): its status in 2002*, *Nucleic Acids Research*, Bd. 30, Nr. 1, 2002, S. 312–317.

[Kon99] Kono, H.; Sarai, A.: *Structure-based prediction of DNA target sites by regulatory proteins*, *Proteins*, Bd. 35, Nr. 1, 1999, S. 114–131.

[Kor93] Kornberg, T. B.: *Understanding the homeodomain*, *Journal of Biological Chemistry*, Bd. 268, Nr. 36, 1993, S. 26813–26816.

[Kud00] Kudo, M.; Sklansky, J.: *Comparison of Algorithms that Select Features for Pattern Classifiers*, *Pattern Recognition*, Bd. 33, Nr. 1, 2000, S. 25–41.

[Lam91] Lamb, P.; McKnight, S. L.: *Diversity and specificity in transcriptional regulation: the benefits of heterotypic dimerization*, *Trends in Biochemical Sciences*, Bd. 16, Nr. 11, 1991, S. 417–422.

[Lan94] Langley, P.: *Selection of Relevant Features in Machine Learning*, in *AAAI Fall Symposium on Relevance*, AAAI Press, New Orleans, LA, 1994.

[Lan97] Lania, L.; Majello, B.; De Luca, P.: *Transcriptional regulation by the Sp family proteins*, *International Journal of Biochemical Cell Biology*, Bd. 29, Nr. 12, 1997, S. 1313–1323.

[Las89] Lassar, A. B.; Buskin, J. N.; Lockshun, D.; Davis, R. L.; Apone, S.; Hanaschka, S. D.; Weintraub, H.: *Myo D is a sequence-specific DNA-binding protein requiring a region of myc homology to bind to the muscle creatine kinase enhancer*, *Cell*, Bd. 58, 1989, S. 823–831.

[Lat98] Latchman, D. S.: *Eukaryotic Transcription Factors*, Academic Press, San Diego, CA, 3. Ausg., 1998.

[Lau88] Lauritzen, S. L.; Spiegelhalter, D. J.: *Local Computation with Probabilities in Graphical Structures and Their Applications to Expert Systems*, *Journal of the Royal Statistical Society B*, Bd. 50, Nr. 2, 1988.

[Law90] Lawrence, C. E.; Reilly, A. A.: *An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences*, *Proteins*, Bd. 7, Nr. 1, 1990, S. 41–51.

[Law93] Lawrence, C.; Altschul, S.; Boguski, M.; Liu, J.; Neuwald, A.; Wootton, J.: *Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment*, *Science*, Bd. 262, Nr. 5131, 1993, S. 208–214.

[Lev02] Levy, S.; Hannenhalli, S.: *Identification of transcription factor binding sites in the human genome sequence*, *Mamm Genome*, Bd. 13, Nr. 9, 2002, S. 510–514.

[Lev03] Levine, M.; Tjian, R.: *Transcription regulation and animal diversity*, *Nature*, Bd. 424, Nr. 6945, 2003, S. 147–151.

[Li94] Li, Z.; D'Ambrosio, B.: *Efficient Inference in Bayes' Networks as a Combinatorial Optimization Problem*, *International Journal of Approximate Inference*, Bd. 11, 1994.

[Liu01] Liu, X.; Brutlag, D. L.; Liu, J. S.: *BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes*, in *Proc. Pacific Symposium on Biocomputing 2001*, 2001, S. 127–138.

[Liu04] Liu, Y.; Liu, X. S.; Wei, L.; Altman, R. B.; Batzoglou, S.: *Eukaryotic regulatory element conservation analysis and identification using comparative genomics*, *Genome Res*, Bd. 14, Nr. 3, 2004, S. 451–458.

[Liu05] Liu, Y.; Vincenti, M. P.; Yokota, H.: *Principal component analysis for predicting transcription-factor binding motifs from array-derived data*, *BMC Bioinformatics*, Bd. 6, Nr. 1, 2005, S. 276.

[Liu06] Liu, X.; Lee, C.-K.; Granek, J. A.; Clarke, N. D.; Lieb, J. D.: *Whole-genome comparison of Leu3 binding in vitro and in vivo reveals the importance of nucleosome occupancy in target site selection*, *Genome Res*, Bd. 16, Nr. 12, 2006, S. 1517–1528.

[Loo02] Loots, G. G.; Ovcharenko, I.; Pachter, L.; Dubchak, I.; Rubin, E. M.: *rVista for comparative sequence-based discovery of functional transcription factor binding sites*, *Genome Res*, Bd. 12, Nr. 5, 2002, S. 832–839.

[Mac94] Macleod, D.; Charlton, J.; Mullins, J.; Bird, A.: *Sp1 sites in the mouse aprt gene promoter are required to prevent methylation of the CpG islands*, *Gene Development*, Bd. 8, 1994, S. 2282–2292.

[Man01] Man, T. K.; Stormo, G. D.: *Non-independence of Mnt repressor-operator interaction determined by a new quantitative multiple fluorescence relative affinity (QuMFRA) assay*, *Nucleic Acids Research*, Bd. 29, Nr. 12, 2001, S. 2471–2478.

[Mar00] Marsan, L.; Sagot, M. F.: *Algorithms for extracting structured motifs using a suffix tree with an application to promoter and regulatory site consensus identification*, *Journal of Computational Biology*, Bd. 7, Nr. 3-4, 2000, S. 345–362.

[Mar02] Markstein, M.; Markstein, P.; Markstein, V.; Levine, M. S.: *Genome-wide analysis of clustered Dorsal binding sites identifies putative target genes in the Drosophila embryo*, *Proc. Natl. Acad. Sci. USA*, Bd. 99, Nr. 2, 2002, S. 763–768.

[Mar06] Markov, K.; Nakamura, S.: *Using Hybrid HMM/BN Acoustic Models: Design and Implementation Issues*, *IEICE Trans Inf Syst*, Bd. E89-D, Nr. 3, 2006, S. 981–988.

[Mat06] Matys, V.; Kel-Margoulis, O. V.; Fricke, E.; Liebich, I.; Land, S.; Barre-Dirrie, A.; Reuter, I.; Chekmenev, D.; Krull, M.; Hornischer, K.; Voss, N.; Stegmaier, P.; Lewicki-Potapov, B.; Saxel, H.; Kel, A. E.; Wingender, E.: *TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes*, *Nucleic Acids Research*, Bd. 34, Nr. Database issue, 2006, S. D108–D110.

[Men06] Meng, H.; Banerjee, A.; Zhou, L.: *BLISS: binding site level identification of shared signal-modules in DNA regulatory sequences*, *BMC Bioinformatics*, Bd. 7, 2006, S. 287–293.

[MG01] Mandel-Gutfreund, Y.; Baron, A.; Margalit, H.: *A structure-based approach for prediction of protein binding sites in gene upstream regions*, in *Proc. Pacific Symposium on Biocomputing 2001*, 2001, S. 139–150.

[Min88] Minsky, M. L.; Papert, S. A.: *Perceptrons*, MIT-Press, 2. Ausg., 1988.

[Mor98] Morgenstern, B.; Frech, K.; Dress, A.; Werner, T.: *DIALIGN: finding local similarities by multiple sequence alignment*, *Bioinformatics*, Bd. 14, Nr. 3, 1998, S. 290–294.

[Mur04] Murakami, K.; Kojima, T.; Sakaki, Y.: *Assessment of clusters of transcription factor binding sites in relationship to human promoter, CpG islands and gene expression*, *BMC Genomics*, Bd. 5, Nr. 1, 2004, S. 16.

[Mü04] Müller, H.-J.; Röder, T.: *Der Experimentator: Microarrays*, Spektrum Akademischer Verlag, Heidelberg, 2004.

[Nar06a] Narlikar, L.; Gordan, R.; Ohler, U.; Hartemink, A. J.: *Informative priors based on transcription factor structural class improve de novo motif discovery*, *Bioinformatics*, Bd. 22, Nr. 14, 2006, S. e384–392.

[Nar06b] Narlikar, L.; Hartemink, A. J.: *Sequence features of DNA binding sites reveal structural class of associated transcription factor*, *Bioinformatics*, Bd. 22, Nr. 2, 2006, S. 157–163.

[Nar07] Narlikar, L.; Gordân, R.; Hartemink, A. J.: *Nucleosome Occupancy Information Improves e novo Motif Discovery*, in Speed, T. P.; Huang, H. (Hrsg.): *Research in Computational Molecular Biology, 11th Annual International Conference*, Bd. 4453 von *Lecture Notes in Computer Science*, Springer, 2007, S. 32–46.

[Nea03] Neapolitan, R. E.: *Learning Bayesian Network*, Prentice Hall, New Jersey, U.S., 2003.

[Neu94] Neuwald, A. F.; Green, P.: *Detecting patterns in protein sequences*, *Journal of Molecular Biology*, Bd. 239, Nr. 5, 1994, S. 698–712.

[Neu95] Neuwald, A. F.; Liu, J. S.; Lawrence, C. E.: *Gibbs motif sampling: detection of bacterial outer membrane protein repeats*, *Protein Sci*, Bd. 4, Nr. 8, 1995, S. 1618–1632.

[Nik07] Nikolajewa, S.; Pudimat, R.; Hiller, M.; Platzer, M.; Backofen, R.: *BioBayesNet: a web server for feature extraction and Bayesian network modeling of biological sequence data*, *Nucleic Acids Research*, Bd. 35, 2007, S. W688–693.

[Ogb98] Ogbourne, S.; Antalis, T. M.: *Transcriptional control and the role of silencers in transcriptional regulation in eukaryotes*, *Biochemical Journal*, Bd. 331 (Pt 1), 1998, S. 1–14.

[Ohl00] Ohler, U.; Stemmer, G.; Harbeck, S.; Niemann, H.: *Stochastic segment models of eukaryotic promoter regions*, in *Proc. Pacific Symposium on Biocomputing 2000*, 2000, S. 380–391.

[Ohl01] Ohler, U.; Niemann, H.; Rubin, G. M.: *Joint modeling of DNA sequence and physical properties to improve eukaryotic promoter recognition*, *Bioinformatics*, Bd. 17 Suppl 1, 2001, S. S199–S206.

[Olm03] Olman, V.; Xu, D.; Xu, Y.: *Identification of regulatory binding sites using minimum spanning trees*, in *Pacific Symposium on Biocomputing 2003*, 2003, S. 327–38.

[Pab00] Pabo, C. O.; Nekludova, L.: *Geometric analysis and comparison of protein-DNA interfaces: why is there no simple code for recognition?*, *Journal of Molecular Biology*, Bd. 301, Nr. 3, 2000, S. 597–624.

[Pav01] Pavesi, G.; Mauri, G.; Pesole, G.: *An algorithm for finding signals of unknown length in DNA sequences*, *Bioinformatics*, Bd. 17 Suppl 1, 2001, S. S207–S214.

[Pea88] Pearl, J.: *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Francisco, U.S., 1988.

[Ped99] Pedersen, A. G.; Baldi, P.; Chauvin, Y.; Brunak, S.: *The biology of eukaryotic promoter prediction–a review*, *Comput Chem*, Bd. 23, Nr. 3-4, 1999, S. 191–207.

[Pen01] Pennacchio, L. A.; Rubin, E. M.: *Genomic strategies to identify mammalian regulatory sequences*, *Nat Rev Genet*, Bd. 2, Nr. 2, 2001, S. 100–109.

[Pev89] Pevzner, P. A.; Mironov, A. A.: *Linguistics of nucleotide sequences. II: Stationary words in genetic texts and the zonal structure of DNA*, *J Biomol Struct Dyn*, Bd. 6, Nr. 5, 1989, S. 1027–1038.

[Pev00] Pevzner, P.; Sze, S.: *Combinatorial approaches to finding subtle signals in DNA sequences*, *Proc Int Conf Intell Syst Mol Biol*, Bd. 8, 2000, S. 269–278.

[Pic98] Pickert, L.; Reuter, I.; Klawonn, F.; Wingender, E.: *Transcription regulatory region analysis using signal detection and fuzzy clustering*, *Bioinformatics*, Bd. 14, Nr. 3, 1998, S. 244–251.

[Plo04] Plotz, T.; Fink, G. A.: *Feature extraction for improved profile hmm based biological sequence analysis*, in *Proceedings of the 17th International Conference on Pattern Recognition*, Bielefeld University, 2004, S. 10.

[Pon97] Ponomarenko, M. P.; Ponomarenko, J. V.; Kel, A. E.; Kolchanov, N. A.: *Search for DNA conformational features for functional sites. Investigation of the TATA box*, in *Proc. Pacific Symposium on Biocomputing 1997*, 1997, S. 340–351.

[Pon99] Ponomarenko, J. V.; Ponomarenko, M. P.; Frolov, A. S.; Vorobyev, D. G.; Overton, G. C.; Kolchanov, N. A.: *Conformational and physicochemical DNA features specific for transcription factor binding sites*, *Bioinformatics*, Bd. 15, Nr. 7-8, 1999, S. 654–668.

[Pra02] Praz, V.; Perier, R.; Bonnard, C.; Bucher, P.: *The Eukaryotic Promoter Database, EPD: new entry types and links to gene expression data*, *Nucleic Acids Research*, Bd. 30, Nr. 1, 2002, S. 322–324.

[Pra04] Prakash, A.; Blanchette, M.; Sinha, S.; Tompa, M.: *Motif discovery in heterogeneous sequence data*, in *PSB04*, 2004, S. 348–359.

[Pra05] Prakash, A.; Tompa, M.: *Discovery of regulatory elements in vertebrates through comparative genomics*, *Nat Biotechnol*, Bd. 23, Nr. 10, 2005, S. 1249–1256.

[Pud94] Pudil, P.; Novovicova, J.; Kittler, J.: *Floating Search Methods In Feature-Selection*, *Pattern Recognition Letters*, Bd. 15, Nr. 11, 1994, S. 1119–1125.

[Pud05] Pudimat, R.; Schukat-Talamazzini, E.; Backofen, R.: *A multiple-feature framework for modelling and predicting transcription factor binding sites*, *Bioinformatics*, Bd. 21, Nr. 14, 2005, S. 3082–3088.

[Pud08] Pudimat, R.; Backofen, R.; Schukat-Talamazzini, E.-G.: *Fast Feature Subset Selection in Biological Sequence Analysis*, *International Journal of Pattern Recognition and Artificial Intelligence*, 2008, S. Accepted.

[Rab89] Rabiner, L.: *A tutorial on hidden Markov models and selected applications in speech recognition*, *Proceedings of the IEEE*, Bd. 77, Nr. 2, Feb 1989, S. 257–286.

[Rah04] Rahmann, S.; Mueller, T.; Vingron, M.: *On the Power of Profiles for Transcription Factor Binding Site Detection*, preprint, MPI Berlin, 2004.

[Raj02] Rajewsky, N.; Vergassola, M.; Gaul, U.; Siggia, E. D.: *Computational detection of genomic cis-regulatory modules applied to body patterning in the early Drosophila embryo*, *BMC Bioinformatics*, Bd. 3, Nr. 1, 2002, S. 30.

[Ren00] 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.; Young, R. A.: *Genome-wide location and function of DNA binding proteins*, *Science*, Bd. 290, Nr. 5500, 2000, S. 2306–2309.

[Rig98] Rigoutsos, I.; Floratos, A.: *Combinatorial pattern discovery in biological sequences: The TEIRESIAS algorithm*, *Bioinformatics*, Bd. 14, Nr. 1, 1998, S. 55–67.

[Rok07] Rokach, L.; Chizi, B.; Maimon, O.: *A methodology for improving the performance of non-ranker feature selection filters*, *IJPRAI*, Bd. 21, Nr. 5, 2007, S. 809–830.

[Rot98] Roth, F. P.; Hughes, J. D.; Estep, P. W.; Church, G. M.: *Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation*, *Nat Biotechnol*, Bd. 16, Nr. 10, 1998, S. 939–945.

[Rou00] Roulet, E.; Bucher, P.; Schneider, R.; Wingender, E.; Dusserre, Y.; Werner, T.; Mermod, N.: *Experimental analysis and computer prediction of CTF/NFI transcription factor DNA binding sites*, *Journal of Molecular Biology*, Bd. 297, Nr. 4, 2000, S. 833–848.

[Sab04] Sabatti, C.; Rohlin, L.; Lange, K.; Liao, J. C.: *Vocabulon: a dictionary model approach for reconstruction and localization of transcription factor binding sites*, *Bioinformatics*, Bd. 21, Nr. 7, 2004, S. 922–931.

[Sag95] Sagot, M. F.; Viari, A.; Pothier, J.; Soldano, H.: *Finding flexible patterns in a text: an application to three-dimensional molecular matching*, *Comput Appl Biosci*, Bd. 11, Nr. 1, 1995, S. 59–70.

[Sag98] Sagot, M.-F.: *Spelling Approximate Repeated or Common Motifs Using a Suffix Tree*, *Lecture Notes in Computer Science*, Bd. 1380, 1998, S. 374–380.

[San04] Sandelin, A.; Alkema, W.; Engstrom, P.; Wasserman, W. W.; Lenhard, B.: *JASPAR: an open-access database for eukaryotic transcription factor binding profiles*, *Nucleic Acids Research*, Bd. 32, Nr. Database issue, 2004, S. D91–D94.

[Sch90] Schneider, T. D.; Stephens, R. M.: *Sequence logos: a new way to display consensus sequences*, *Nucleic Acids Research*, Bd. 18, Nr. 20, 1990, S. 6097–6100.

[Sch98] Schneider, T. D.; Stormo, G. D.; Ehrenfeucht, A.: *Information content of binding sites on nucleotide sequences*, *Journal of Molecular Biology*, Bd. 188, 1998, S. 415–431.

[Sch07] Schones, D. E.; Smith, A. D.; Zhang, M. Q.: *Statistical significance of cis-regulatory modules*, *BMC Bioinformatics*, Bd. 8, 2007, S. 19.

[See74] Seeman, N. C.; Rosenberg, J. M.; Rich, A.: *Sequence-specific recognition of double helical nucleic acids by proteins*, in *Proc. Natl Acad. Sci. USA*, 1974, S. 804–808.

[Seg06] Segal, E.; Fondufe-Mittendorf, Y.; Chen, L.; Thastrom, A.; Field, Y.; Moore, I. K.; Wang, J.-P. Z.; Widom, J.: *A genomic code for nucleosome positioning*, *Nature*, Bd. 442, Nr. 7104, 2006, S. 772–778.

[Sei92] Seipel, K.; Georgiev, O.; Schaffner, W.: *Different activation domains stimulate transcription from remote ('enhancer') and proximal ('promoter') positions*, *EMBO J*, Bd. 11, Nr. 13, 1992, S. 4961–4968.

[Shm02] Shmulevich, I.; Dougherty, E. R.; Kim, S.; Zhang, W.: *Probabilistic Boolean Networks: a rule-based uncertainty model for gene regulatory networks*, *Bioinformatics*, Bd. 18, Nr. 2, 2002, S. 261–274.

[Sin03] Sinha, S.; van Nimwegen, E.; Siggia, E.: *A Probabilistic Method to Detect Regulatory Modules*, *Bioinformatics*, Bd. 19, 2003, Proceedings of the 11th International Conference on Intelligent Systems for Molecular Biology (ISMB 2003).

[Smi90] Smith, R. F.; Smith, T. F.: *Automatic generation of primary sequence patterns from sets of related protein sequences*, *Proc Natl Acad Sci*, Bd. 87, Nr. 1, 1990, S. 118–122.

[Spi98] Spiro, C.; McMurray, C. T.: *Footprint analysis of DNA-protein complexes in vitro and in vivo*, IRL Press, New York, 2. Ausg., 1998.

[ST95] Schukat-Talamazzini, E. G.: *Automatische Spracherkennung. Grundlagen, statistische Modelle und effiziente Algorithmen*, Vieweg, Wiesbaden, 1995.

[Sta84] Staden, R.: *Computer methods to locate signals in nucleic acid sequences*, *Nucleic Acids Research*, Bd. 12, Nr. 1 Pt 2, 1984, S. 505–519.

[Sta89a] Staden, R.: *Methods for calculating the probabilities of finding patterns in sequences*, *Comput Appl Biosci*, Bd. 5, Nr. 2, 1989, S. 89–96.

[Sta89b] Staden, R.: *Methods for discovering novel motifs in nucleic acid sequences*, *Comput Appl Biosci*, Bd. 5, Nr. 4, 1989, S. 293–298.

[Sta95] Starr, D. B.; Hoopes, B. C.; Hawley, D. K.: *DNA bending is an important component of site-specific recognition by the TATA binding protein*, *Journal of Molecular Biology*, Bd. 250, Nr. 4, 1995, S. 434–446.

[Ste76] Stearns, S. D.: *On selecting features for pattern classifiers*, in *Third International Conference on Pattern Recognition*, 1976, S. 71–75.

[Sto82] Stormo, G. D.; Schneider, T. D.; Gold, L. M.: *Characterization of translational initiation sites in E. coli*, *Nucleic Acids Research*, Bd. 10, Nr. 9, 1982, S. 2971–2996.

[Sto98] Stormo, G. D.; Fields, D. S.: *Specificity, free energy and information content in protein-DNA interactions*, *TiBS*, Bd. 23, Nr. 3, 1998, S. 109–113.

[Sto00] Stormo, G. D.: *DNA binding sites: representation and discovery*, *Bioinformatics*, Bd. 16, Nr. 1, 2000, S. 16–23.

[Sty04] Styczynski, M. P.; Jensen, K. L.; Rigoutsos, I.; Stephanopoulos, G. N.: *An extension and novel solution to the (l,d)-motif challenge problem*, *Genome Inform Ser Workshop Genome Inform*, Bd. 15, Nr. 2, 2004, S. 63–71.

[Sue90] Suermondt, H. J.; Cooper, G. F.: *Probabilistic Inference in Multiply Connected Belief Networks Using Loop Cutsets*, *International Journal of Approximate Inference*, Bd. 4, 1990.

[Sug96] Sugimoto, N.; Nakano, S.; Yoneyama, M.; Honda, K.: *Improved thermodynamic parameters and helix initiation factor to predict stability of DNA duplexes*, *Nucleic Acids Research*, Bd. 24, 1996, S. 4501–4505.

[Suz97] Suzuki, M.; Amano, N.; Kakinuma, J.; Tateno, M.: *Use of a 3D structure data base for understanding sequence-dependent conformational aspects of DNA*, *Journal of Molecular Biology*, Bd. 274, Nr. 3, 1997, S. 421–435.

[Thi02] Thijs, G.; Marchal, K.; Lescot, M.; Rombauts, S.; De Moor, B.; Rouze, P.; Moreau, Y.: *A Gibbs sampling method to detect overrepresented motifs in the upstream regions of coexpressed genes*, *Journal of Computational Biology*, Bd. 9, Nr. 2, 2002, S. 447–464.

[Tho06] Thomas, A.; Hara, B. O.; Ligges, U.; Sturtz, S.: *Making BUGS Open*, *R News*, Bd. 6, 2006, S. 12–17.

[Tom05] Tompa, M.; Li, N.; Bailey, T. L.; Church, G. M.; De Moor, B.; Eskin, E.; Favorov, A. V.; Frith, M. C.; Fu, Y.; Kent, W. J.; Makeev, V. J.; Mironov, A. A.; Noble, W. S.; Pavesi, G.; Pesole, G.; Regnier, M.; Simonis, N.; Sinha, S.; Thijs, G.; van Helden, J.; Vandenbogaert, M.; Weng, Z.; Workman, C.; Ye, C.; Zhu, Z.: *Assessing computational tools for the discovery of transcription factor binding sites*, *Nat Biotechnol*, Bd. 23, Nr. 1, 2005, S. 137–144.

[Uch07] Uchyigit, G.; Clark, K.: *A new feature selection method for text classification*, *IJPRAI*, Bd. 21, Nr. 2, 2007, S. 423–438.

[UV03] Ureta-Vidal, A.; Ettwiller, L.; Birney, E.: *Comparative genomics: genome-wide analysis in metazoan eukaryotes*, *Nat Rev Genet*, Bd. 4, Nr. 4, 2003, S. 251–262.

[Vis05] Vishnevsky, O. V.; Kolchanov, N. A.: *ARGO: a web system for the detection of degenerate motifs and large-scale recognition of eukaryotic promoters*, *Nucleic Acids Research*, Bd. 33, Nr. Web Server issue, 2005, S. W417–W422.

[Wag99] Wagner, A.: *Genes regulated cooperatively by one or more transcription factors and their identification in whole eukaryotic genomes*, *Bioinformatics*, Bd. 15, Nr. 10, 1999, S. 776–784.

[Was98] Wasserman, W. W.; Fickett, J. W.: *Identification of regulatory regions which confer muscle-specific gene expression*, *Journal of Molecular Biology*, Bd. 278, Nr. 1, 1998, S. 167–181.

[Wat53] Watson, J. D.; Crick, F. H.: *Molecular structure of nucleic acids. A structure for deoxyribose nucleic acid*, *Nature*, Bd. 171, 1953, S. 737–741.

[Wat84] Waterman, M.; Arratia, R.; Galas, D.: *Pattern recognition in several sequences: consensus and alignment*, *Bull. Math. Biol.*, Bd. 46, 1984, S. 515–527.

[Wer99] Werner, T.: *Models for prediction and recognition of eukaryotic promoters*, *Mamm Genome*, Bd. 10, Nr. 2, 1999, S. 168–175.

[Whi71] Whitney, A. W.: *A direct method of nonparametric measurement selection*, *IEEE Transactions on Computers*, Bd. 20, Nr. 9, 1971, S. 1100–1103.

[Wol96] Wolfertstetter, F.; Frech, K.; Herrmann, G.; Werner, T.: *Identification of functional elements in unaligned nucleic acid sequences by a novel tuple search algorithm*, *Comput Appl Biosci*, Bd. 12, Nr. 1, 1996, S. 71–80.

[Wol99] Wolfe, S. A.; Greisman, H. A.; Ramm, E. I.; Pabo, C. O.: *Analysis of zinc fingers optimized via phage display: evaluating the utility of a recognition code*, *Journal of Molecular Biology*, Bd. 285, Nr. 5, 1999, S. 1917–1934.

[Wu00] Wu, T. D.; Nevill-Manning, C. G.; Brutlag, D. L.: *Fast probabilistic analysis of sequence function using scoring matrices*, *Bioinformatics*, Bd. 16, Nr. 3, 2000, S. 233–244.

[Yad98] Yada, T.; Totoki, Y.; Ishikawa, M.; Asai, K.; Nakai, K.: *Automatic extraction of motifs represented in the hidden Markov model from a number of DNA sequences*, *Bioinformatics*, Bd. 14, Nr. 4, 1998, S. 317–325.

[Yu03] Yu, L.; Liu, H.: *Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution*, in *ICML*, 2003, S. 856–863.

[Yu06] Yu, X.; Lin, J.; Masuda, T.; Esumi, N.; Zack, D. J.; Qian, J.: *Genome-wide prediction and characterization of interactions between transcription factors in Saccharomyces cerevisiae*, *Nucleic Acids Research*, Bd. 34, Nr. 3, 2006, S. 917–927.

[Yua05] Yuan, G.-C.; Liu, Y.-J.; Dion, M. F.; Slack, M. D.; Wu, L. F.; Altschuler, S. J.; Rando, O. J.: *Genome-Scale Identification of Nucleosome Positions in S. cerevisiae*, *Science*, Bd. 309, Nr. 5734, 2005, S. 626–630.

[Zha93] Zhang, M. Q.; Marr, T. G.: *A weight array method for splicing signal analysis*, *Comput Appl Biosci*, Bd. 9, Nr. 5, 1993, S. 499–509.

[Zha96] Zhang, N. L.; Poole, D.: *Exploiting Causal Independence in Bayesian Network Inference*, *Journal of Artificial Intelligence Research*, Bd. 5, 1996, S. 301–328.

[Zho04] Zhou, Q.; Liu, J. S.: *Modeling within-motif dependence for transcription factor binding site predictions*, *Bioinformatics*, Bd. 20, Nr. 6, 2004, S. 909–916.