Informatik VII - Theoretische Informatik und Grundlagen der künstlichen Intelligenz

Einsatz Neuronaler Netze für die Erkennung und Klassifizierung von Promotorstrukturen in genomischen DNA Sequenzen

Korbinian Grote

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzende: Univ.-Prof. Dr. A. Brüggemann-Klein
Prüfer der Dissertation:
1. Univ.-Prof. Dr. Dr. h.c. W. Brauer
2. Univ.-Prof. Dr. E. W. Mayr
3. Hon.-Prof. Dr. R. Balling, Technische Universität Braunschweig (schriftliche Beurteilung)

Die Dissertation wurde am 13. Dezember 2001 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 15. November 2005 angenommen.

Zusammenfassung

Neben den *Genen*, den Bauplänen für die zum Leben benötigten Proteine, enthält die DNA aller bekannten Organismen auch regulatorische Bereiche, die *Promotoren*, die für die Steuerung der Gen-Expression verantwortlich sind. Bisher bekannte Software-Methoden zur Klassifizierung bzw. Erkennung von Promotoren setzen darauf, anhand einiger weniger, zumeist direkt in der Nukleotidsequenz gefundener, gemeinsamer Merkmale zu beurteilen, welche Bereiche regulatorische Funktionen enthalten. Ihre Anwendbarkeit ist damit zum einen auf solche Promotoren beschränkt, die entsprechende Merkmale enthalten, zum anderen treffen sie nur eine generelle Entscheidung bezüglich der allgemeinen Promotoreigenschaft.

Mit *PromoterMap* wurde zum ersten Mal ein Verfahren bereitgestellt, das eine funktionelle Einteilung a priori unbekannter Promotoren ermöglicht. Zusätzlich erlaubt es die Generierung von Strukturen für die allgemeine genomweite Erkennung regulatorischer Bereiche. Damit steht der Molekularbiologie ein Werkzeug zur Verfügung, das ohne kostspielige Laborversuche einen schnellen und einfachen Einblick in biologisch relevante Zusammenhänge unbekannter Sequenzen erlaubt.

Für die Gruppierung der regulatorischen Sequenzen benötigt *PromoterMap* lediglich Informationen über die in ihnen enthaltenen Proteinbindungsstellen sowie deren Position im Verhältnis zueinander. Diese Daten können durch die Verwendung bereits existierender Software einfach verfügbar gemacht werden. Damit ist das Verfahren in der Lage, Sequenzen unabhängig von ihrer Nukleotidzusammensetzung in beliebiger Anzahl zu verarbeiten.

Die Methode basiert auf der hierarchischen Verknüpfung selbst-organisierender Karten. Die von Kohonen eingeführten Verfahren zur Visualisierung von Strukturen innerhalb merkmalsbehafteter Daten kommen dabei in verschiedenen Varianten zum Einsatz. Für die Anwendung dieser Algorithmen

war die Generierung vergleichbarer charakteristischer Merkmale sowohl für die Bindungsstellen wie auch für die Promotoren selbst unabdingbare Voraussetzung. Die Hierarchie der einzelnen Schichten des Verfahrens bildet in natürlicher Weise die biologischen Mechanismen bei der Promotorerkennung nach. Zunächst werden einzelne Proteinbindungsstellen erkannt und eingeordnet, um daraus ein Gesamtbild der Promotorstruktur zu erzeugen, das anschließend zum Vergleich mit anderen Sequenzen herangezogen werden kann. Der Aufbau des Verfahrens orientiert sich dabei streng an den natürlichen Organisationsprinzipien von Promotoren, wodurch eine hohe Kohärenz der Methode mit den biologischen Gegebenheiten erzielt wird.

Die im Rahmen dieser Arbeit mit *PromoterMap* erstellte Visualisierung der regulatorischen Sequenzen ist damit das einzige bekannte Verfahren, das eine funktionelle Anordnung der Promotoren auf der Basis von potenziell biologisch wirksamen Sequenz-Elementen erstellt. Es wird also nicht — wie sonst üblich — ein Vergleich der Einzelnukleotide der DNA verwendet, sondern eine biologisch sinnvolle Abstraktion, der die Kenntnisse über die Bedeutung der Promotorstruktur für die Gen-Regulation zugrundeliegen. Damit ist das Verfahren nicht nur auf im Training verwendete, isolierte Promotorgruppen anwendbar, sondern kann ohne alle Einschränkungen für die Gruppierung unbekannter Sequenzen herangezogen werden.

Darüberhinaus lassen sich bereits Ergebnisse aus den einzelnen Teilen des Verfahrens nutzbringend bei der Analyse von Promotoren einsetzen. So konnte mit der Erstellung von längenunabhängigen Merkmalsbeschreibungen für die Bindungsstellenmatrizen die Bildung von Gruppen ähnlicher Matrizen nahezu vollständig automatisiert werden. Dadurch wurde eine erhebliche Verminderung der bei der Suche nach diesen Motiven anfallenden redundanten Ergebnisse erreicht und so die Qualität der Analysen deutlich erhöht. Vergleiche mit unabhängigen Untersuchungen der Bindeproteine ergaben eine hohe biologische Relevanz der rein algorithmisch gefundenen Matrix-Gruppen. Weiterhin können aus dem Vergleich der Visualisierungen, die letztlich als Merkmale für die Promotorgruppierung dienen, für Gruppen von funktionell ähnlichen Promotoren automatisch hochspezifische Modelle erzeugt werden. Beide Teilverfahren basieren auf biologischen Prinzipien, jedoch nicht auf biologischem Vorwissen. Verwendet werden lediglich die aus den Daten gewonnenen Charakteristika.

Danksagung

An dieser Stelle möchte ich mich sehr herzlich bei allen bedanken, die zum Gelingen dieser Arbeit beigetragen haben.

Zunächst gilt mein Dank meinem Doktorvater an der Technischen Universität München, Herrn Prof. Dr. Dr. h.c. Wilfried Brauer, für die Ermöglichung der Durchführung dieser Arbeit an seinem Lehrstuhl und die Betreuung bei der Bearbeitung der Aufgabenstellung.

Bei Herrn Prof. Dr. Ernst Mayr bedanke ich mich für die Übernahme der Aufgabe des Zweitgutachters.

Herrn Prof. Dr. Rudi Balling danke ich sowohl für die Erstellung des Gutachtens für den biologischen Aspekt der Arbeit als auch für seine Unterstützung als Leiter des Instituts für Säugetiergenetik an der GSF.

Mein besonderer Dank gilt Herrn Dr. Thomas Werner. Er ermöglichte mir nicht nur die Mitarbeit in der AG BIODV, sondern war auch der Initiator der Arbeit und hat mich während der ganzen Zeit mit Anregungen, Ratschlägen, Diskussionen, offenen Ohren für alle meine Ideen, der nötigen Kritik und vor allem viel Geduld unterstützt.

Auch bei allen Kollegen der AG BIODV bzw. der Genomatix Software GmbH möchte ich mich für die gute Zusammenarbeit und die fortwährende Unterstützung bedanken. Inbesondere bei Kerstin Cartharius und Dr. Matthias Scherf auf der „informatischen“ und Dr. Ralf Schneider auf der „biologischen“ Seite für die vielen Tipps, Tricks und anregenden Diskussionen.

Die Arbeit wurde am Forschungszentrum für Umwelt und Gesundheit GmbH (GSF) durchgeführt und von dieser finanziert.

An letzter Stelle aber keinesfalls zuletzt möchte ich mich ganz besonders herzlich bei meinen Eltern und Geschwistern bedanken, die mich während der ganzen Zeit in jeder nur erdenklichen Hinsicht unterstützt haben.

Inhaltsverzeichnis

Kapitel 1

Einleitung

> „Das 1990 gestartete, international koordinierte Humane Genom Projekt (HUGO) ist mit einem Umfang von ca. 3 Milliarden US-Dollar das größte, das jemals in der Biologie begonnen wurde. Ziel dieser gewaltigen Anstrengung ist es, das komplette 'genetische Buch', also alle 3 Milliarden 'Buchstaben' mit denen dieser Grundplan geschrieben ist, bis zum Jahr 2005 zu entschlüsseln und alle ca. 100.000 menschlichen Gene [...] zu identifizieren. [...] Sobald die Sequenz des menschlichen Genoms vollständig bekannt ist, kann man die Aufmerksamkeit von der Suche nach den Genen, auf die viel spannendere Frage nach ihrer Funktion richten.“ [82]

Bereits jetzt, im Jahr 2001, vier Jahre vor dem angestrebten Termin für die vollständige Entschlüsselung, ist das Ziel erreicht. Ein erster Entwurf, der ca. 90% des menschlichen Genoms beinhaltet, ist verfügbar [30] und die Untersuchung der Funktion einzelner Abschnitte des Erbguts rückt in das Zentrum des Interesses. Die Menge der Daten und die Komplexität der funktionellen Abhängigkeiten macht den Computer dabei zu einem unersetzlichen Hilfsmittel für die Forschung. Auch wenn die Labormethoden der molekularen Biologie durch informatische Verfahren nicht zu ersetzen sind, können diese doch wertvolle Hilfestellungen leisten um die meist sehr zeitaufwändigen und teuren Prozeduren im Labor schneller und kostengünstiger zu machen. Die vorliegende Arbeit soll dazu beitragen, indem sie ein Verfahren bereitstellt, das die Klassifikation regulatorischer Bereiche des Genoms ermöglichen bzw. vereinfachen soll.

1.1 Problemstellung

Bei der Untersuchung unbekannter DNA-Sequenzen stellen sich im wesentlichen drei Fragen:

1. Wo in der Sequenz liegen biologisch funktionelle Einheiten?
2. Um welche Funktion handelt es sich dabei?
3. Mit welchen anderen funktionellen Einheiten gibt es Interaktionen?

Im Labor erfolgt die Beantwortung mit Hilfe molekularbiologischer Methoden, die sich in vielen Fällen bekannter Mechanismen aus der Natur bedient, um so zu Ergebnissen zu gelangen. In den allermeisten Fällen sind diese Verfahren zu komplex, um vollständig durch Algorithmen nachgebildet werden zu können. In der Bioinformatik macht man sich stattdessen eine der grundlegendsten Eigenschaften der Genome zu Nutze: ihre strukturelle Organisation in Untereinheiten. Diese hierarchische Aufteilung, die von vergleichsweise großen (z. B. den Chromosomen) zu immer kleineren (z. B. den Codons) funktionellen Einheiten gestaffelt ist, findet man in vergleichbarer Form im Erbmaterial fast aller Organismen. Ähnlichkeiten zwischen diesen Strukturen nicht nur in der Funktion, sondern eben auch in ihrer Zusammensetzung ermöglichen die Anwendung von Mustererkennungs- und Klassifikationsverfahren für die Erforschung dieser Bereiche.

Allerdings ist diese Form der Organisation auch die Ursache für viele der Schwierigkeiten, die bei der Suche nach biologisch gleichwertigen Bereichen im Genom auftreten. Während die kleineren funktionellen Elemente, wie beispielsweise Transkriptionsfaktoren, Splice-sites, o. ä. aufgrund ihrer vergleichsweise klar definierten Muster meist einfacher zu finden sind, sorgt die Variabilität bei größeren Strukturen dafür, dass die Anwendung herkömmlicher Verfahren oft zu keinen oder nur ungenügend genauen Ergebnissen führt. Einerseits geht die Natur oft unterschiedliche Wege, um dasselbe Ziel zu erreichen, andererseits werden viele Mechanismen immer wieder an unterschiedlichen Stellen für ähnliche Zwecke eingesetzt. Die Erkennung und Bewertung solcher Regelmäßigkeiten ist eines der Ziele der Bioinformatik, die sich dafür verschiedenster Methoden bedient, hauptsächlich aus der Mustererkennung und Statistik. Häufig werden solche Verfahren jedoch aus nicht-biologischen Umfeldern — Bild- und Spracherkennung, Zeitreihenanalyse, Textverarbeitung — übernommen, ohne zielgerichtet an die biologischen Sachverhalte angepasst zu werden. Die so gefundenen Modelle geben deshalb oft nur einen kleinen, sehr speziellen Ausschnitt der Biologie wieder, oder sie sind sehr

allgemein, so dass sie für die Suche nach distinkten funktionellen Einheiten in größerem Maßstab wenig geeignet sind. Beispielsweise liefern viele Verfahren bei der Untersuchung von einzelnen Genen oder kürzeren Sequenz-Abschnitten gute Ergebnisse, sind aber für die Erforschung ganzer Genome nicht geeignet, weil die Anzahl fehlerhafter Prognosen die der richtigen Vorhersagen um ein vielfaches übertrifft. Das Ziel, biologisch interessante Bereiche in großem Maßstab für die genauere Untersuchung im Labor zu markieren, ist damit nicht realisierbar.

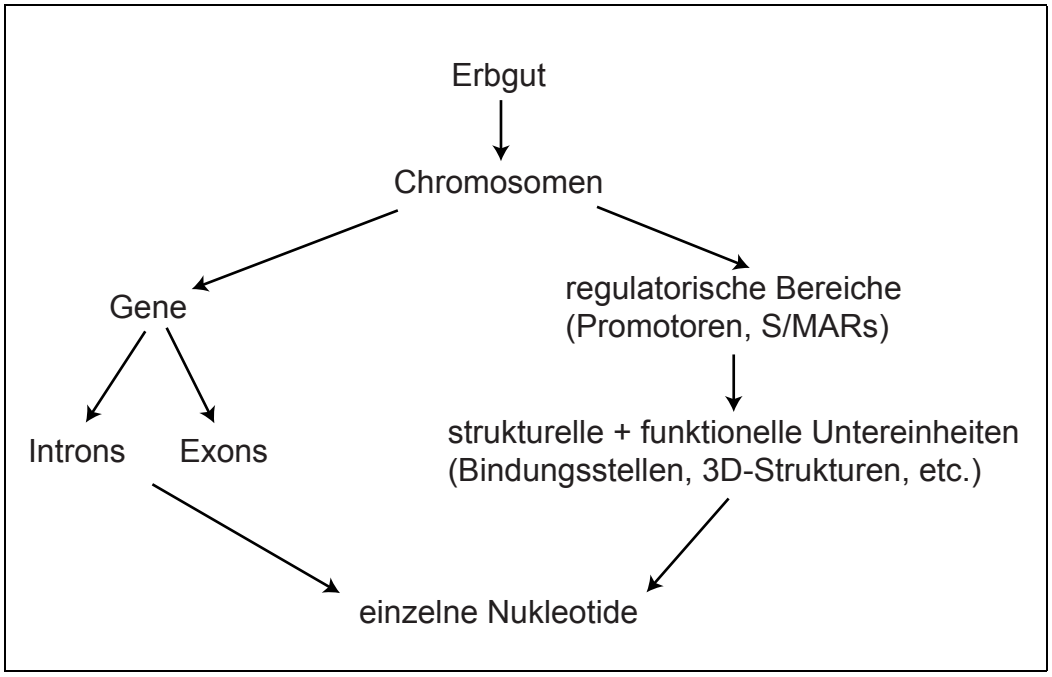

Abbildung 1.1: Exemplarische Darstellung funktionell bedeutender hierarchischer Strukturen innerhalb der Organisation molekularbiologischer Information. Das gesamte Erbgut ist (bei eukaryotischen Organismen) auf die Chromosomen verteilt. Innerhalb dieser lassen sich Gene und die dazugehörigen regulatorischen Bereiche ausmachen. Diese sind wiederum in verschiedenen, charakteristischen Untereinheiten organisiert. Die kleinste molekularbiologische übliche Einheit sind dann die einzelnen Basen der DNA.

Zusätzlich erschwert wird die algorithmische Analyse molekularbiologischer Fragestellungen durch die oft unzureichenden Datenmengen. Bei herkömmlichen Problemen der Mustererkennung stehen meist mehrere 100 oder 1.000 Datensätze zur Verfügung, die sich auf statistische Regelmäßigkeiten hin untersuchen oder für das Training einer Mustererkennungsmethode verwenden lassen. In der Molekularbiologie sind solche Datenmengen nur für Vorhersa-

gen zur Lokalisation allgemeinerer Strukturen, etwa der Gene, vorhanden. Präzise Aussagen über deren Funktion lassen sich damit nicht treffen; in der Spracherkennung wäre dies beispielsweise äquivalent zum Erkennen von Wortanfängen und -endungen. Möchte man zusätzlich noch die Bedeutung der Wörter, bzw. in der Biologie die Funktion erfassen, wird der Unterschied in der Anzahl der Trainingsdaten deutlich. Bei Strukturen mit präzisen Angaben über deren funktionelle Bedeutung muss man sich in der Bioinformatik derzeit häufig mit 10 bis 20 Datensätzen begnügen.

1.2 Zielsetzung

Biologisches Wissen sollte deshalb nicht erst bei der Anwendung, sondern schon beim Entwurf von Software berücksichtigt werden. Auf diese Weise werden grundlegende Eigenschaften der zu untersuchenden Sachverhalte von Anfang an miteinbezogen und können so zu einer verbesserten Modellierung der Biologie beitragen. Die Anwendung dieser Vorgehensweise bei der Entwicklung einer neuen Methodik ist eines der Hauptziele der vorliegenden Arbeit. Bei den zu erforschenden Objekten handelt es sich um spezielle *regulatorische Bereiche* im Genom, für deren Analyse bereits eine Reihe von Verfahren entwickelt wurden, die sowohl in der Anwendbarkeit und den Ergebnissen als auch in den verwendeten Algorithmen variieren. Ein Teil dieser Programme basiert auf der Anwendung *künstlicher neuronaler Netze*[1] für die Erkennung funktioneller Bereiche innerhalb der DNA-Sequenz. In den meisten Fällen handelt es sich um sogenannte „überwachte Verfahren“, die mit Hilfe von Trainingsdaten lernen, bestimmte Muster wiederzufinden. Leider eignen sich diese Methoden häufig nur für die Erkennung sehr genereller Muster und sind damit für die spezifische Untersuchung bestimmter Gruppen funktioneller DNA-Abschnitte nur bedingt geeignet.

Mit der Bereitstellung von immer mehr vollständig sequenzierten Genomen sind zwar in zunehmendem Maß Rohdaten verfügbar, die Funktion der einzelnen Sequenzen bleibt aber vorerst größtenteils unklar. Die Regionen im Genom, über die fundiertes Wissen vorhanden ist, nehmen nur einen kleinen Raum ein. Dementsprechend können vorhandene heuristische Verfahren das in ihnen enthaltene Wissen oft nur eingeschränkt auf die unbekannten Bereiche projizieren. Die automatische Generierung begründeter Hypothesen ohne Vorwissen über Struktur und Funktion bestimmter DNA-Bereiche ist deshalb ebenfalls ein wichtiger Aspekt bei der Entwicklung von Methoden

[1] Im folgenden kurz: *neuronale Netze*

für die Genom-Analyse.

Die Ziele dieser Arbeit sind daher:

- Die Entwicklung eines Verfahrens für die Klassifikation regulatorischer Gen-Bereiche. Als Grundlage sollen dabei Algorithmen aus dem Bereich der künstlichen neuronalen Netze dienen. Im Gegensatz zu vorhandenen Methoden soll der Schwerpunkt nicht in der Erkennung allgemeiner Strukturen bestehen, sondern im Vergleich der funktionellen Bedeutung einzelner Bereiche.

- Biologisches Vorwissen soll, soweit möglich, in die Entwicklung des Verfahrens mit eingehen. Dies schließt die Verwendung bereits existierender Software für die Bereitstellung von Teilergebnissen mit ein.

- Die Unterschiede zu anderen Verfahren auf der Grundlage neuronaler Netze sollen präsentiert werden.

- Das Verfahren sollte in der Lage sein, Hypothesen zu erstellen, die nicht auf vorher erworbenem Wissen basieren, aber durch die in den Daten gefundenen Ergebnisse begründet sind.

Zum Erreichen dieser Ziele muss zunächst vorhandenes biologisches Wissen über die zu untersuchenden DNA-Abschnitte gesammelt und auf seine Verwendbarkeit für ein algorithmisches Verfahren untersucht werden. Anschließend ist aus der Palette vorhandener Algorithmen aus dem Bereich der neuronalen Netze eine geeignete Methode auszuwählen oder entsprechend den Anforderungen zu modifizieren. Die biologischen Daten müssen in eine für die gewählte Netzwerkarchitektur verwendbare Form umgewandelt werden. Das entwickelte Verfahren ist dann anhand bekannter Ergebnisse auf seine Tauglichkeit zu überprüfen.

1.3 Überblick

Im folgenden Kapitel werden die benötigten biologischen Grundlagen zusammengefasst und aktuelle Labor- und Software-Methoden vorgestellt. Anschließend werden einige Varianten künstlicher neuronaler Netze präsentiert. Das Hauptaugenmerk richtet sich dabei auf Algorithmen, die entweder für das neu zu entwickelnde Verfahren in Betracht kommen, oder solche, die in bereits existierenden Methoden verwendet werden. Letztere werden am Ende des Kapitels untersucht. Die Architektur des neuen Verfahrens steht

im Mittelpunkt von Kapitel 4, seine Anwendung wird im darauffolgenden Abschnitt beschrieben. Die Zusammenfassung und der Ausblick auf weitere Entwicklungen in Kapitel 6 schließen diese Arbeit ab.

Kapitel 2

Biologische Grundlagen der Promotoranalyse

Eines der wichtigsten Gebiete der aktuellen molekularbiologischen Forschung ist die Analyse von Genomen, den Trägern der Erbinformation lebender Organismen. Im Mittelpunkt des Interesses steht dabei die Funktion der Gene, die - in der DNA kodiert - die Baupläne für die zum Leben benötigten Proteine enthalten. Sie bestimmen damit den Genotyp eines Lebewesens: seine Struktur, den jeweiligen Stoffwechselapparat; bei höheren Lebewesen die Differenzierung in verschiedene Gewebe und Organe und auch das äußere Erscheinungsbild, den Phänotyp.

In unterschiedlichen Organismen variiert die Anzahl der Gene genauso wie der Anteil der den Genen zugeordneten Nukleotidsequenzen innerhalb der DNA des kompletten Genoms. Während bei der Hefe in den rund 12 Millionen Basen ca. 6.000 Gene liegen, erwartet man beim Menschen ca. 30.000 bis 80.000 Gene, deren protein-kodierende Bereiche zusammengenommen aber nur etwa 3% der 3 Milliarden Basen langen Sequenz in Anspruch nehmen [8]. Die Bereiche zwischen den Genen sind jedoch nicht ohne Funktion. Ein Teil von ihnen ist für die korrekte Umsetzung der in den Genen enthaltenen Information verantwortlich. Eine besonders wichtige Rolle bei der Genregulation spielen die Promotoren: vergleichsweise kurze DNA-Bereiche, die den Genen vorgelagert sind und diesen sowohl als Markierung wie auch als „Schalter“ dienen können. Die Analyse dieser Promotoren stellt daher einen wichtigen Aspekt der funktionellen Genomforschung dar. In diesem Kapitel sollen zunächst die biologischen Grundlagen der Genregulation erläutert werden, anschließend folgt ein Abriss über verschiedene Labor- und Software-Methoden zur Promotoranalyse.

2.1 Biologische Grundlagen

Der Träger des Genoms und damit der Erbinformation ist bei fast allen Lebewesen (mit Ausnahme einiger Viren) die Desoxyribonukleinsäure (*desoxyribo nucleic acid*, DNA). Bei den einfachsten einzelligen Organismen, den *Prokaryoten* (beispielweise Bakterien oder Blaualgen) liegt sie frei innerhalb der Zelle vor, bei höheren Organismen, den *Eukaryoten* befindet sie sich im Zellkern, der durch eine Membran vom Rest der Zelle getrennt ist. Die DNA hat die Form einer Doppelhelix. Sie ist zusammengesetzt aus den *Nukleotiden*, die aus einem Zuckermolekül (der *Desoxyribose*) bestehen, an das ein Phosphatrest und jeweils eine der vier Basen *Adenin*, *Guanin*, *Cytosin* oder *Thymin* angelagert sind. Paarweise Zusammensetzungen dieser Basen (Adenin ist immer kombiniert mit Thymin, Guanin mit Cytosin) sitzen innerhalb der Doppelhelix wie die Stufen einer Wendeltreppe. Das äußere Gerüst wird durch Phosphat-Brücken zwischen den einzelnen Nukleotiden gebildet [37].

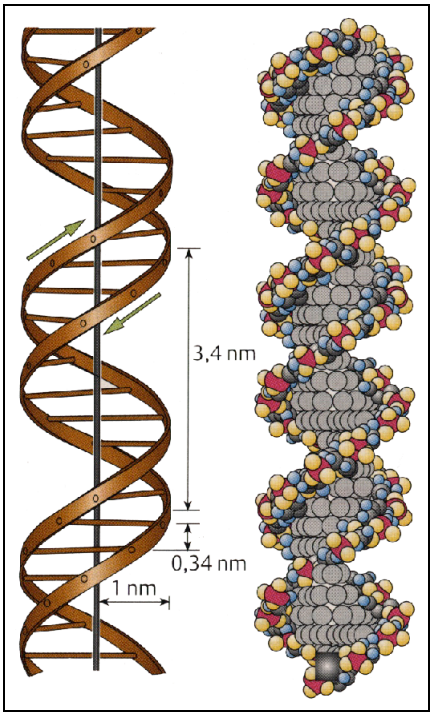

Abbildung 2.1: Die Doppelhelix der DNA. Links eine schematische Darstellung, rechts ein Kalottenmodell, bei der jedes am Aufbau der DNA beteiligte Atom durch eine Kugel dargestellt wird. (aus [37])

Durch die unterschiedliche Aneinanderreihung der Basen wird die enthaltene Information kodiert. Sie wird einerseits als „Bauplan“ für die vom Organis-

mus benötigten Proteine verwendet, andererseits auch zur Steuerung, wann und wo welches Protein hergestellt wird. Diejenigen Abschnitte der DNA, welche die Anleitung zum Aufbau der Proteine enthalten, bezeichnet man als *Gene*. Jedes Gen kodiert ein oder mehrere Proteine. Die im Gen enthaltene Information wird durch ein Enzym (die RNA-Polymerase[1]) abgelesen und kopiert; diesen Vorgang bezeichnet man als *Transkription*. Bei den Prokaryoten gibt es eine, bei den Eukaryoten drei RNA-Polymerasen. Von diesen dreien ist jedoch nur eine, die Polymerase II[2], an der Transkription der meisten proteinkodierenden Gene beteiligt. Bei der Abschrift der Geninformation handelt es sich um die *mRNA* ('messenger RNA'). Diese mRNA wird anschließend zu den in der Zelle enthaltenen Ribosomen transportiert. Dort entsteht dann durch die *Translation* das kodierte Protein.

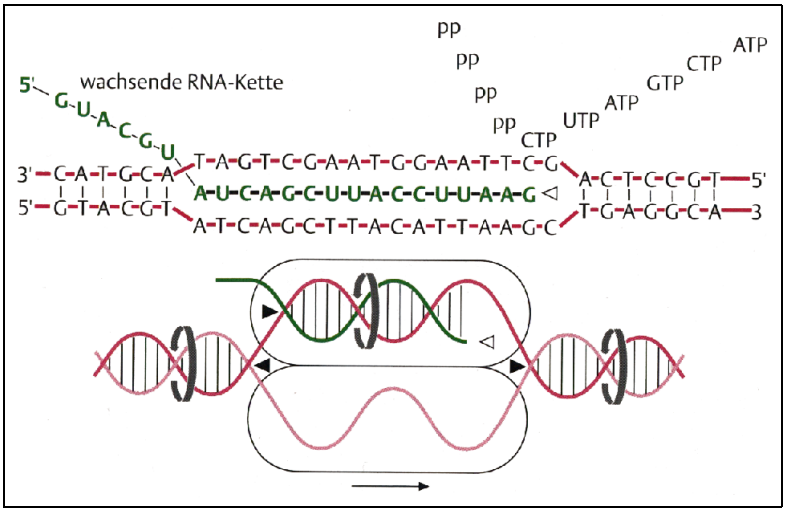

Abbildung 2.2: Schematische Darstellung der Transkription (aus [37]): Die Polymerase entwindet die Doppelhelix der DNA und erstellt dann eine Kopie der DNA, die mRNA. In der RNA wird der Baustein Thymin durch Uracil ersetzt.

Die Polymerasen I und III werden für die Herstellung von tRNA ('transfer RNA') und rRNA ('ribosomal RNA') benötigt, die Funktionen bei der Proteinsynthese erfüllen.

Ein weiterer Unterschied zwischen den Prokaryoten und den Eukaryoten besteht in der Organisation der Gene. Bei den meisten Lebewesen der ersten Gruppe liegen die Gensequenzen ununterbrochen an einem Stück vor, dage-

[1] *RNA* = 'ribonucleic acid'
[2] im folgenden oft einfach nur als Polymerase bezeichnet

gen findet man bei höheren Organismen innerhalb eines Gens sowohl Bereiche, die kodierende Funktion haben (die *Exons*), wie auch andere, die zwischen diese Regionen eingeschoben sind (die *Introns*), deren Funktion bis heute noch nicht vollständig geklärt ist. Die Introns werden bei der Transkription zunächst in die *prä-mRNA* übernommen, aber vor der Weiterverarbeitung in den Ribosomen durch das sogenannte *Spleißen* entfernt, wodurch dann die mRNA entsteht [37, 5].

2.1.1 Genregulation

Während der Organismus manche Proteine ständig und überall benötigt, werden andere nur zu bestimmten Zeitpunkten in der Entwicklung, in unterschiedlichen Geweben oder als Reaktion auf Umwelteinflüsse gebraucht. Die Differenzierung der Gliedmaßen findet beim Menschen z. B. nur einmal in der Embryonalphase statt, daran beteiligte Botenstoffe werden daher auch nur zu diesem Zeitpunkt benötigt. Andere Enzyme müssen hingegen stets in ausreichender Menge vorhanden sein. Dementsprechend kann man das Genom aufteilen in die Gruppe der sogenannten Haushalts-Gene (*housekeeping genes*) und die Gruppe der spezifischer regulierten Gene. Erstere sind immer aktiv und können daher permanent transkribiert werden. Sie „tragen beispielsweise die Information zur Herstellung von Enzymen des Stoffwechsels oder von Proteinen des Cytoskeletts.“ [37] Im Gegensatz dazu findet bei der zweiten Gruppe eine spezifische Regulation statt. Hier kann selektiv bestimmt werden, ob, wann oder wo ein individuelles Gen benutzt wird.

An der Steuerung der Genregulation sind verschiedene Mechanismen beteiligt. Die wichtigste Rolle spielt dabei ein Bereich, der am Anfang eines Gens vor der kodierenden Sequenz lokalisiert ist: der *Promotor*. Soll ein Gen transkribiert werden, bildet sich am Promotor der *Transkriptions-* oder auch *Initiationskomplex*, der als Signal für die Polymerase dient. Die Polymerase benutzt ihn, um an die DNA binden zu können und die Startposition für die Transkription zu finden.

Dieser Komplex setzt sich aus verschiedenen Proteinen, den *Transkriptionsfaktoren* zusammen. Diese binden sowohl direkt an die DNA als auch untereinander. Jeder Transkriptionsfaktor[3] besitzt spezielle funktionelle Untereinheiten für die Anlagerung an die DNA, die *Bindedomänen*. Diese Bereiche

[3]Im folgenden auch kurz als *Faktor* bezeichnet

im Protein haben meist eine charakteristische dreidimensionale Struktur, welche die Bindung an die Doppelhelix begünstigt. Anhand dieser Strukturen lassen sich die Transkriptionsfaktoren in Klassen einteilen. Man spricht beispielsweise von *bZIP*-Proteinen ('basic region/leucin zipper binding domain'), *bHLH*-Proteinen ('basic region/helix-loop-helix domain') oder Faktoren mit einer Homeobox-Bindedomäne. Während diese Strukturen bei allen Faktoren einer Klasse untereinander ähnlich sind, binden sie dennoch individuell verschieden an bestimmte Stellen der DNA.

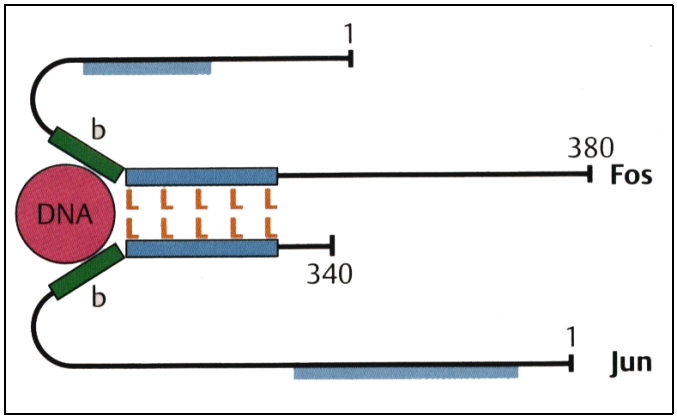

Abbildung 2.3: Schematische Darstellung des Transkriptionfaktors AP1 (aus [37]): die basischen Bereiche des aus den Proteinen *Fos* und *Jun* zusammengesetzten Faktors umfassen die DNA wie eine geöffnete Schere. Rechts davon die 'leucin zipper region', von der diese Protein-Klasse ihren Namen hat. Die Regionen ober- und unterhalb sind die Transaktivierungsdomänen, die für die Interaktion zwischen den einzelnen Bindeproteinen bei der Bildung des Initiationskomplexes wichtig sind.

Zusätzlich zu ihrer Bindedomäne besitzen viele Transkriptionsfaktoren einen *Transaktivierungsbereich*. Dabei handelt es sich um eine Untereinheit, über die ein Protein mit anderen in Wechselwirkung treten und ein Signal vermitteln kann. Diese wechselseitigen Interaktionen spielen bei der Ausbildung des Initiationskomplexes eine wichtige Rolle, indem sie Verbindungen zwischen den einzelnen Faktoren oder weiteren für die Transkription benötigten Proteinen herstellen.

Das passende Gegenstück zu den Bindedomänen der Transkriptionsfaktoren sind die *Bindungsstellen* innerhalb der DNA ('transcription factor binding sites'). Es handelt sich dabei um Motive von ca. 10-20 Nukleotiden Sequenzlänge, die von den Bindedomänen „erkannt“ werden. Entsprechend der

Individualität der Bindedomänen sind auch die Bindungsstellen spezifisch für die einzelnen Faktoren. Allerdings muss die Sequenz der Bindungsstelle nicht immer identisch sein. In beschränktem Maß sind Varianten in der Nukleotidanordnung innerhalb des Motivs möglich. Diese Unterschiede können sich auf die Qualität der Bindung des Proteins an die DNA auswirken. Dabei ist nicht unbedingt die Anzahl der Änderungen das entscheidende Kriterium, sondern der Anteil, den eine bestimmte Base zur Bindung beiträgt. Der Austausch eines unbedingt erforderlichen Nukleotids wirkt sich unter Umständen kritischer aus als mehrere Ersetzungen für die Bindung nicht zwingend benötigter Basen. Bezeichnen $b_1 \dots b_n$ die einzelnen Nukleotide der Bindestelle, sowie a_i den jeweiligen „Beitrag“, den ein Nukleotid zur Bindung leistet, kann man die Fähigkeit eines DNA-Stückes $d_1 \dots d_n$, ein Protein zu binden, formal durch die Erfüllung der Ungleichung

$$\sum_i a_i \delta_i > \rho, \qquad \delta_i = \begin{cases} 1 & : \quad b_i = d_i \\ 0 & : \quad \text{sonst} \end{cases}$$

beschreiben. Hierbei steht ρ für einen für die jeweilige Bindung individuellen Schwellwert.[4] Innerhalb dieses Rahmens gibt es eine Vielzahl an Variationen: zur Zeit sind mehrere hundert verschiedene Transkriptionsfaktoren, aber mehr als tausend unterschiedliche Bindungsstellenmotive bekannt [57].

Eng verwandt mit den Promotoren sind die *Enhancer*. Sie gleichen ihnen in der Struktur und tragen ebenfalls zur Bildung des Initiationskomplexes bei. Der Unterschied besteht in der Positionierung. Während die Promotoren unmittelbar vor dem Anfang der kodierenden Sequenz lokalisiert sind, findet man Enhancer in größerer Distanz oder auch hinter dem Transkriptionsstart innerhalb von Introns [74]. Die DNA ist kein starres Molekül, sondern lässt sich, innerhalb bestimmter Grenzen, verbiegen oder verwinden. So kann der räumliche Abstand zwischen innerhalb des DNA-Stranges eigentlich weit auseinanderliegenden Bereichen durch das sogenannte '*DNA-looping*' deutlich verkürzt werden. Dabei bildet ein Teil der dazwischenliegenden DNA eine Schlaufe, und verkürzt so die Entfernung auf dem Strang.

Weitere für die Gen-Regulation wichtige Elemente sind 'locus control regions' (*LCR*s) und 'scaffold/matrix attachment regions' (*S/MAR*s). Da ihre

[4] Insgesamt handelt es sich hierbei jedoch um eine grobe Vereinfachung, denn für die tatsächliche Proteinbindung spielen noch andere Faktoren, wie beispielsweise die schon erwähnten wechselseitigen Interaktionen eine Rolle. Das Auffinden einer geeigneten Transkriptionsfaktorbindungsstelle stellt daher lediglich eine Mindestanforderung für die Anlagerung eines Proteins dar.

Funktion sich grundlegend von der von Promotoren und Enhancern unterscheidet - beide spielen bei der Ausbildung des Initiationskomplexes keine unmittelbare Rolle - ist ihre Untersuchung nicht Gegenstand dieser Arbeit.

Im folgenden wird nun genauer auf die Struktur von Promotoren und die Rolle der Transkriptionsfaktorbindungsstellen eingegangen.

2.1.2 Merkmale regulatorischer Gen-Bereiche

Die Aufgabe eines Promotors ist die Steuerung der Transkription eines Gens. Dazu leitet er die Polymerase an den Anfang der kodierenden Sequenz. Die dabei ablaufenden Vorgänge und die daran beteiligten Elemente eines Promotors sollen nun für eukaryotische Gene und demzufolge die RNA Polymerase II näher beschrieben werden.

Die Polymerase benötigt für das Anlagern an die DNA mehrere *generelle Transkriptionsfaktoren* ('general transcription factors', GTFs), die im allgemeinen einfach als TFII-A bis TFII-J bezeichnet werden.[5] Diese bilden zusammen mit der Polymerase an der DNA den Initiationskomplex. So wird die Polymerase exakt am Startpunkt der mRNA-Synthese (*TSS*[6]) positioniert. Durch Phosphorylierung löst sie sich schließlich aus dem Komplex und beginnt mit der Transkription (sh. Abb. 2.2). Dabei spielt eine Bindungsstelle in der DNA, die sogenannte *TATA-Box* eine wichtige Rolle. An dieses ca. 30 Basen vor dem Transkriptionsstart gelegene Element bindet zunächst der Faktor TFII-D, an den dann alle weiteren GTFs binden. Die TATA-Box findet man in verschiedenen Varianten in den Promotoren; in den allermeisten Fällen beinhaltet sie das Sequenz-Motiv '`TATA`', von dem sie ihren Namen hat. Allerdings enthalten nur schätzungsweise 50% aller Promotoren dieses Motiv. Die übrigen verwenden verschiedene andere Mechanismen zur Positionierung des Initiationskomplexes.

Der Transkriptionsstart zusammen mit der ihn umgebenden Region ist ein weiteres wichtiges Promotorelement. Da hier die Transkription „initiiert" wird, spricht man von der 'initiatior region' ('initiator', INR). Im Gegensatz zur TATA-Box findet man hier kein übereinstimmendes Sequenz-Motiv. Am Transkriptionsstart findet man häufig eine Adenin-Base, der meist ein Cytosin vorangestellt ist [37]. Da diese Kombination aus zwei Basen stati-

[5] **t**ranscription **f**actor for polymerase **II**
[6] 'transcription start site'

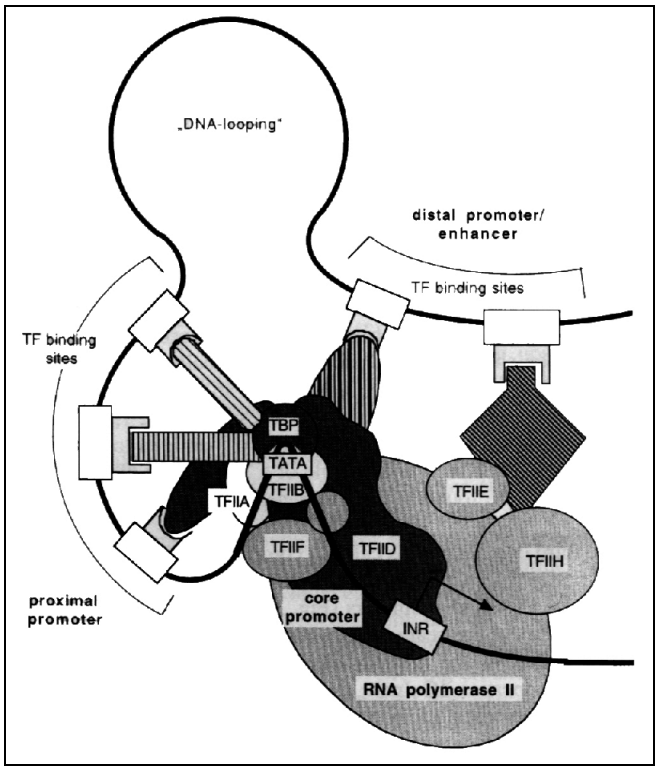

Abbildung 2.4: Schematische Darstellung eines Transkriptionkomplexes (aus [79]). In der Mitte die TATA-Box und der Verbund aus GTFs und der Polymerase. Weitere spezifische Transkriptionsfaktoren sind ebenfalls beteiligt.

stisch betrachtet allerdings alle 16 Nukleotide einmal vorkommt, ist sie als Muster für das Auffinden der INR nicht geeignet. Abgesehen von diesem '`CA`'-Signal sind kaum weitere Gemeinsamkeiten zu finden. Die Kombination aus TATA-Box und INR wird oft als 'core promoter' bezeichnet (sh. Abb. 2.5). Für die Funktionalität eines Promotors ist keines der beiden Elemente zwingend erforderlich [79]. Das Fehlen der TATA-Box kann beispielsweise durch den generellen Transkriptionsfaktor TFII-I ausgeglichen werden. Dieser bindet einerseits an den INR Bereich, andererseits an den Faktor TFII-D und steuert so die Positionierung des Initiationskomplexes [37].

Zusätzlich zu den Bestandteilen des core promoters gibt es weitere Bindungsstellen für Transkriptionsfaktoren. Diese verteilen sich auf die Abschnitte, die in Abb. 2.5 als 'proximal promoter' und 'distal promoter' bezeichnet sind.[7] Der 'proximal promoter' Bereich erstreckt sich bis zu einigen

[7]Der Begriff *Promotor* wird in der Literatur nicht einheitlich verwendet. Teilweise wird

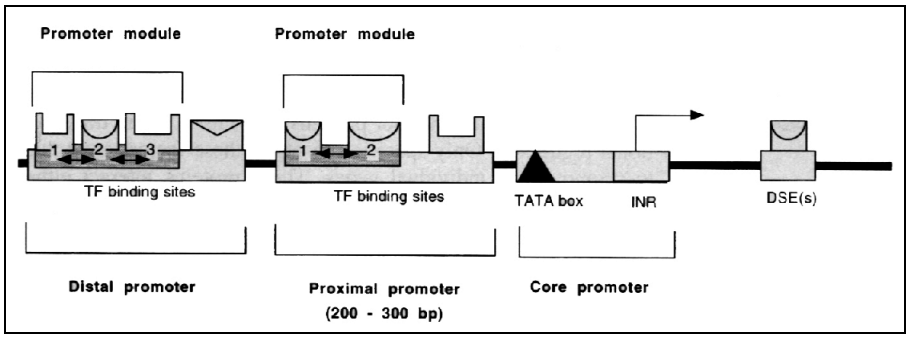

Abbildung 2.5: Schematische Darstellung des Aufbaus eines Promotors (aus [79])

hundert Basen vor dem Transkriptionsstart. Oft enthält er eine — neben der TATA-Box ebenfalls häufig zu findende — Bindungsstelle, die *CCAAT-Box*, die ihren Namen genau wie die TATA-Box von der ihr zugrundeliegenden Nukleotid-Sequenz hat. An diese Box bindet eine Gruppe verschiedener Transkriptionsfaktoren, über deren genaue Beteiligung am Aufbau des Transkriptions-Komplexes noch keine vollständigen Kenntnisse vorliegen. Eines der gebundenen Proteine fördert die Replikation von Adenoviren, indem es dort Replikations-Proteine an ihre Positionen dirigiert [37]. Dementsprechend könnte es bei der Transkription ebenfalls dafür verantwortlich sein, dass andere Proteine korrekt an die DNA angelagert werden. Die 'distal promoter region' ist der variabelste Bereich des Promotors. Sie kann mehrere hundert, aber auch bis zu 2000 Nukleotide [35] lang sein. In diesen beiden Abschnitten lassen sich spezifische Bindungsstellen für alle möglichen Transkriptionsfaktoren finden [79]. Auch nach dem Transkriptionsstart finden sich manchmal noch Bindungsstellen, sogenannte 'Downstream elements'. Diese zusätzlich von der DNA gebundenen Proteine sind entweder selbst ebenfalls Bestandteile des Initiationskomplexes oder sie tragen indirekt zu seiner Bildung bei. Sie interagieren über ihre Transaktivierungsdomänen untereinander und mit den generellen Transkriptionsfaktoren. Bei Promotoren ohne TATA-Box und klar definierter Initiator-Region stellen sie eine Alternative für die korrekte Positionierung der GTFs dar. Weitere positive Einflüsse sind die Stabilisierung des Transkriptions-Komplexes oder auch die Verformung der DNA, um weiteren Proteinen die Bindung zu erleichtern (sogenanntes 'cooperative binding'). Allerdings gibt es auch Faktoren, welche die Transkription negativ beeinflussen. Sie verhindern etwa die Anlagerung anderer

damit nur der hier als 'core promoter' definierte Bereich bezeichnet, teilweise 'core' und 'proximal promoter'. Hier soll die Definition aus [79] Anwendung finden, die alle drei der in Abb. 2.5 dargestellten Bereiche umfasst.

Proteine, indem sie deren Bindungsstelle überdecken oder sie verändern die dreidimensionale Struktur der DNA, um so die Bildung des Initiationskomplexes zu stören. Auf diese Weise kann die Transkription beendet oder verhindert werden; die Transkriptionsfaktoren ermöglichen also eine spezifische Regulation der Gene.

Für die Steuerung der Regulation ist dabei sowohl entscheidend, welche Faktoren beteiligt sind, als auch welche Organisation die Bindungsstellen innerhalb des Promotors besitzen. Während die generellen Transkriptionsfaktoren in fast allen Zellen in ausreichender Menge zu finden sind [37], ist das Vorhandensein der übrigen Proteine von einer Vielzahl unterschiedlicher Voraussetzungen abhängig. Manche kommen nur in bestimmten Zelltypen vor, andere werden nur aufgrund von äußeren Einflüssen wie zum Beispiel Hunger, Krankheit, Verletzungen oder anderer Umwelteinflüsse erzeugt. Durch eine geeignete Auswahl der entsprechenden Faktoren läßt sich so die *Spezifität* eines Promotors und damit die Regulation des zugehörigen Gens beeinflussen. Ein Beispiel sind die sogenannten 'heat shock factors', eine Gruppe von Proteinen, die bei erhöhter Temperatur in den Zellen freigesetzt werden. Bindungsstellen für diese Faktoren lassen sich dementsprechend in den Promotoren von Genen nachweisen, die Proteine kodieren, die für die Fiebersenkung benötigt werden [43].

Für das Zustandekommen der Transkription ist also nicht nur die Existenz bestimmter Bindungsstellen sondern auch deren Positionierung innerhalb des Promotors bzw. relativ zueinander maßgeblich. Sie wirkt sich auf das Zustandekommen der oben beschriebenen Protein-Interaktionen und deren Einflussnahme auf die Genregulation aus. Beispielsweise kann eine Reihe von Proteinen sogenannte *COMPELS* ('composite elements') bilden, die die Genregulation maßgeblich beeinflussen. Dabei binden beide Faktoren in relativ kurzem Abstand an die DNA und aneinander [33]. Für Gene, die funktionell ähnliche Proteine kodieren, ist zu erwarten, dass sich die Zusammensetzung ihrer Promotoren ähnelt, da auf diese Weise verwandte Proteine mit denselben Mechanismen reguliert werden können. Ein Beispiel dafür sind die Aktin-Gene. Bei dieser Familie von Proteinen, die wichtige Funktionen in der Zellstruktur und beim Muskelaufbau haben, lassen sich in den jeweiligen Promotoren immer wieder die gleichen Bindungsstellen in der gleichen Reihenfolge und ähnlichen Abständen finden. Bei Aktinen, die im Muskelgewebe exprimiert werden, findet man noch eine zusätzliche Bindungsstelle, über die offensichtlich die Gewebespezifität geregelt wird. Ein weiteres Beispiel ist der Interleukin-2 Promotor. Hier findet man sowohl in der DNA von

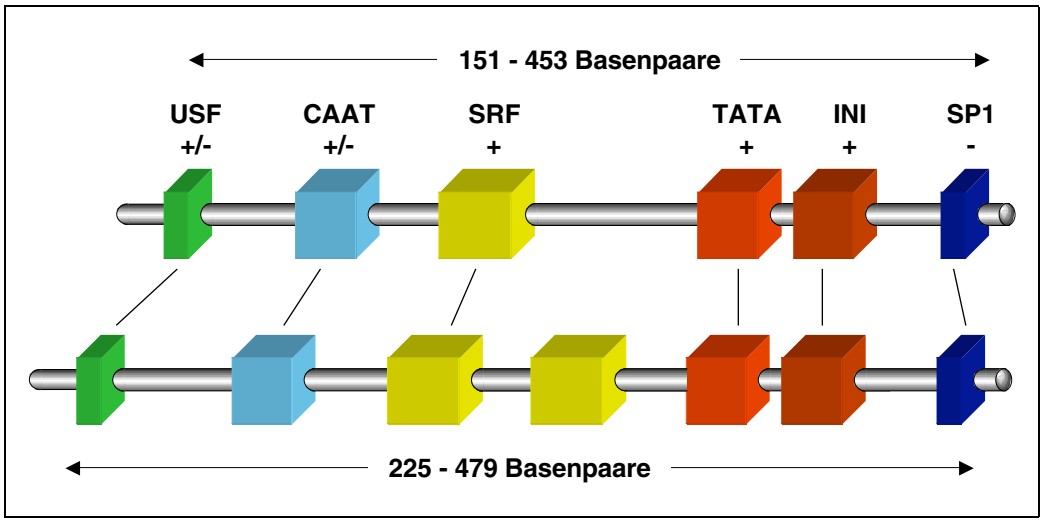

Abbildung 2.6: Schematische Darstellung des Aufbaus der Aktinpromotoren. Oben der generelle Promotor, unten der muskelspezifische, bei dem noch eine SRF ('serum response factor') Bindungsstelle eingefügt ist.

Mäusen als auch in der von Menschen dieselben Proteinbindungsstellen in der Promotorsequenz. Die generelle Ähnlichkeit bezüglich der Nukleotidanordnung ist jedoch nicht besonders ausgeprägt [37]. Auch bei 'heat shock' und muskelspezifischen Genen sind in den jeweiligen Promotoren paarweise auftretende Bindungsstellen identifiziert worden [9, 17]. Solche Kombinationen von Bindungsstellen, die in funktionell ähnlichen Promotoren auftauchen werden auch als *Promotor-Module* bezeichnet [79]. Im Unterschied zu den COMPELS können die daran beteiligten Bindungsstellen auch weiter voneinander entfernt sein.

Funktionell verwandte eukaryotische Promotoren lassen sich also durch die in ihnen enthaltenen Transkriptionsfaktorbindungsstellen und deren relativen Abstände charakterisieren. Die formale Beschreibung einer solchen Struktur nennt man *Promotor-Modell*[8]. Bezeichnet man mit el_i ein *Element* des Modells (beispielsweise eine Transkriptionsfaktorbindungsstelle) und mit d_i die Distanz zwischen den Elementen el_i und el_{i+1}, erhält man mit

$$\mathcal{M} := \{(el_1, \ldots, el_n), (d_1, \ldots, d_{n-1})\}$$

eine Definition für ein Modell mit n Elementen. Da die Abstände zwischen den einzelnen Elementen jedoch meistens in gewissem Rahmen variabel sind,

[8] im folgenden kurz *Modell*

ist es sinnvoll, die Definition von $\mathcal{M}$ wie folgt zu erweitern:

$$\mathcal{M} := \{(el_1, \ldots, el_n), (d_1^{min}, \ldots, d_{n-1}^{min}), (d_1^{max}, \ldots, d_{n-1}^{max})\} \tag{2.1}$$

Dabei bezeichnet d_i^{min} die Minimal-, d_i^{max} die Maximaldistanz zwischen den Elementen el_i und el_{i+1}. Für die Elemente kommen sowohl einzelne Bindungsstellen als auch andere Modelle in Frage. So können beispielsweise die oben beschriebenen COMPELS durch einfache Modelle beschrieben werden, die sich dann wiederum in komplexere Strukturen einbetten lassen. Da sowohl die Bindungsstellen, als auch die Modelle über eine räumliche Ausdehnung verfügen, muss außerdem festgelegt sein, wo an den einzelnen Elementen die Messpunkte für die Abstände liegen.

Rein statistisch betrachtet, ließen sich mit den bekannten Transkriptionsfaktoren und ihnen zugeordneten Abständen beliebig viele Promotor-Modelle konstruieren. Theoretisch könnte jedem Gen ein individueller Promotor zugeordnet werden. Tatsächlich wird in der Natur diese Vielzahl der Möglichkeiten jedoch nicht ausgenutzt. Wie man beispielsweise an den Aktin-Promotoren sieht, wird durch eine kleine Variation des Grundgerüsts die Transkription in unterschiedlichen Geweben realisiert.

In Abb. 2.6 sind die Modelle für die Aktin-Promotoren grafisch dargestellt. Sie enthalten sowohl die TATA- als auch die CCAAT-Box. Hinter dem TSS findet sich als 'Downstream element' eine SP1 ('stimulating protein 1') Bindungsstelle. In der 'proximal promoter region' liegen Bindungsstellen für USF ('upstream stimulating factor') und SRF ('serum response factor'), die beim muskelspezifischen Modell noch einmal zusätzlich enthalten ist. Formal betrachtet erhält man

$$\begin{aligned} \mathcal{M}_1 \;=\; & \{(\mathsf{USF},\mathsf{CAAT},\mathsf{SRF},\mathsf{TATA},\mathsf{INI},\mathsf{SP1}), (18, 31, 31, 20, 7), \\ & (161, 99, 193, 49, 90)\} \end{aligned}$$

für das normale, und

$$\begin{aligned} \mathcal{M}_2 \;=\; & \{(\mathsf{USF},\mathsf{CAAT},\mathsf{SRF},\mathsf{SRF},\mathsf{TATA},\mathsf{INI},\mathsf{SP1}), \\ & (24, 34, 31, 18, 20, 0), (218, 100, 74, 167, 49, 90)\} \end{aligned}$$

für das muskelspezifische Aktin-Modell. Definiert man für den hier enthaltenen core-Promotor Bereich das Modell

$$\mathcal{M}_{CP} = \{\mathsf{TATA},\mathsf{INI}\}, (20), (49)\},$$

kann man sie auch wie folgt beschreiben:

$$\begin{aligned}\mathcal{M}_{1a} &= \{(\mathsf{USF},\mathsf{CAAT},\mathsf{SRF},\mathcal{M}_{CP},\mathsf{SP1}),(18,31,31,27),\\ &\quad (161,99,193,139)\},\end{aligned}$$

bzw.

$$\begin{aligned}\mathcal{M}_{2a} &= \{(\mathsf{USF},\mathsf{CAAT},\mathsf{SRF},\mathsf{SRF},\mathcal{M}_{CP},\mathsf{SP1}),(24,34,31,18,20),\\ &\quad (218,100,74,167,139)\}.\end{aligned}$$

Auch bei den Promotoren gibt es wesentliche Unterschiede zwischen den Eukaryoten und den Prokaryoten. Erstere sind, wie im vorangegangenen Abschnitt deutlich wurde, komplexe Gebilde, bei denen sich funktionelle Verwandtschaft nicht unbedingt in der globalen Ähnlichkeit ihrer Nukleotid-Sequenzen sondern vielmehr in der Übereinstimmung einzelner Strukturelemente und deren Arrangement widerspiegelt. Dagegen sind die Promotoren von Bakterien sehr viel einfacher aufgebaut und zeichnen sich durch vergleichsweise große Ähnlichkeit auf der Sequenz-Ebene vor allem im 'core promoter' Bereich aus. Beispielsweise lassen sich bei etwa 60% der bekannten Promotoren des Bakteriums Escherichia coli die der TATA-Box ähnelnde Pribnow-Box und das Sequenzmotiv '`TTGACA`' 10 bzw. 35 Basen vor dem TSS finden [37].

Den Promotoren sehr ähnlich sind, wie bereits erwähnt, die *Enhancer*. Sie enthalten genau wie 'proximal' und 'distal promoter' Bindungsstellen für spezielle Transkriptionsfaktoren, oftmals mehrere des gleichen Typs in kurzem Abstand hintereinander. Im allgemeinen fehlt ihnen der 'core promoter' Bereich und sie befinden sich in weitem Abstand vom Transkriptionsstart. Die Biegsamkeit der DNA ermöglicht es ihnen aber, sich trotzdem am Aufbau des Initiationskomplexes zu beteiligen (siehe Abb. 2.4). Die Abgrenzung zwischen Enhancern und weit entfernten Elementen der 'distal promoter' Region ist dabei nicht immer klar definiert [79]. Im Unterschied zu den Promotoren ist die Enhancerfunktion im allgemeinen orientierungsunabhängig.

2.2 Methoden zur Promotoranalyse

Wie in den vorangegangenen Abschnitten dargelegt wurde, handelt es sich bei den Promotoren um wichtige funktionelle Einheiten im Genom. Dementsprechend ist ihre Erforschung und die Analyse der Elemente, aus denen sie aufgebaut sind, für die Molekularbiologie von großem Interesse. Die Verfahren, die im Labor angewendet werden können, entsprechen in ihrer Vielfalt

den Fragestellungen, die sich im Zusammenhang mit regulatorischen Sequenzen ergeben. Das Spektrum reicht von der Bestimmung der Position über die Analyse der einzelnen enthaltenen Bindungsstellen bis zu quantitativen Fragen, wie stark ein Promotor und damit das zugehörige Gen in bestimmten Zellen aktiviert wird. Für viele dieser Probleme existieren nicht nur Labormethoden, sondern auch Software, die die Arbeit im Labor zwar nicht ersetzen kann, aber dazu geeignet ist, die oftmals langwierigen experimentellen Tests einfacher und effizienter zu gestalten. Ein weiteres Ziel dieser Verfahren ist es, begründete Arbeitshypothesen zu erstellen, deren Richtigkeit sich dann im Labor bestätigen oder widerlegen lässt.

In diesem Abschnitt sollen einige der relevanten Methoden sowohl aus dem Labor- wie auch aus dem Softwarebereich vorgestellt werden. Im Mittelpunkt stehen dabei die Programme, deren Ergebnisse im Rahmen dieser Arbeit später weiterverwendet werden. Alle hier vorgestellten Algorithmen können nur Hinweise auf biologisch relevante Muster geben. Die tatsächliche Funktionalität kann nur im Labor mit letzter Sicherheit nachgewiesen werden.

2.2.1 Labormethoden

Die Palette der Methoden in modernen molekularbiologischen Labors ist zu vielfältig, um hier vollständig behandelt zu werden. Nur ein paar der für die Promotorforschung wichtigeren Verfahren sollen daher kurz vorgestellt werden, um einen Überblick zu vermitteln, auf welche Weise die für Promotor-Analysen benötigten Daten ermittelt werden können.

Polymerase-Ketten-Reaktion Die Polymerase-Ketten-Reaktion ('polymerase chain reaction', PCR) dient der Vervielfältigung von DNA. Zunächst werden vom Beginn und Ende der zu vervielfältigenden Sequenz kurze Stücke mittels Oligonukleotidsynthese kopiert. Diese sogenannten *Primer* können maschinell relativ einfach in großer Zahl hergestellt werden. Die zu replizierende DNA wird zunächst durch Hitzedenaturierung in ihre Einzelstränge aufgespalten, so dass eine Anlagerung der Primer möglich wird. Diese werden dann zusammen mit einer DNA-Polymerase (einem Enzym, das partiell einsträngige DNA zur Doppelhelix komplettiert) und einem Nukleotidmix zu der denaturierten DNA gegeben. Ausgehend von den Primern vervollständigt nun die Polymerase die Sequenz. Durch die zyklische Wiederholung dieses Vorgangs erhält man schnell[9] und einfach eine große Anzahl von Kopien

[9] Da jede Kopie wiederum als Vorlage dienen kann, nimmt die Zahl der Kopien in der Regel exponentiell zu.

der ursprünglichen DNA, die dann für weitere Analysen verwendet werden können.

DNA-Sequenzierung Mit der Sequenzierung von DNA kann ihre genaue Nukleotid-Zusammensetzung bestimmt werden. Auch hier benutzt man eine DNA-Polymerase um den Gegenstrang der zu untersuchenden Sequenz zu synthetisieren. Die Synthese wird allerdings durch den Einsatz von Dideoxy Varianten der vier Basen[10] gezielt unterbrochen, so dass verschieden lange Bruchstücke der Sequenz entstehen, deren letztes Nukleotid jeweils bekannt ist. Diese Bruchstücke werden nach ihren Endbasen geordnet auf ein Polyacrylamid-Gel aufgetragen, an das eine elektrische Spannung angelegt wird. Entsprechend ihrer Länge wandern die Bruchstücke innerhalb des elektrischen Feldes im Gel unterschiedlich weit; durch Vergleiche der verschiedenen Positionen kann so die Zusammensetzung der Sequenz ermittelt und als einfach zu lesende Buchstabenfolge verfügbar gemacht werden. Die Automatisierung dieses Verfahrens in großem Maßstab war eine wesentliche Voraussetzung für die Analyse großer genomischer Sequenzen.

Herstellung von cDNA-Stücken Die Herstellung von *cDNA* ('complementary DNA') entspricht der Umkehrung der Transkription. Mit Hilfe eines Enzyms (der *reversen Transkriptase*) wird aus der Nukleotidsequenz der mRNA eines Gens die entsprechende kodierende DNA-Sequenz erstellt. Mit Hilfe dieses Verfahrens lassen sich einzelnen Proteinen ihre Gensequenz innerhalb der DNA zuordnen. Theoretisch ließe sich damit auch die Position des Promotors bestimmen, allerdings wird die cDNA vom Ende der RNA her aufgebaut und liegt in der Praxis oft nur unvollständig vor, so dass eine exakte Lokalisierung häufig nicht möglich ist.

Sepharose-Bindung (SELEX-Verfahren) Mit diesem Verfahren kann festgestellt werden, welche Proteine an ein bestimmtes DNA-Motiv binden. Dazu wird das entsprechende Motiv als Oligonukleotid synthetisiert und in einer Affinitätssäule an Sepharosekörner gekoppelt. Durch diese Säule wird dann ein Proteingemisch geleitet. Proteine mit der entsprechenden Bindedomäne lagern sich in der Säule ab, alle anderen werden unten wieder ausgespült. Da die Sequenz des Oligonukleotids vorgegeben ist, kann so beispielsweise untersucht werden, an welche Bindungsstellen sich ein Transkriptionsfaktor anlagert.

[10]Dabei handelt es sich um speziell modifizierte Basen, sogenannte *Dideoxynukleotide*, die von der DNA-Polymerase nicht weiter verlängert werden können und zum gezielten Strangabbruch führen.

DNA-Schutz Experimente Eine weitere Methode zur Untersuchung der Proteinbindung sind DNA-Schutz Experimente. Dabei wird die zu analysierende DNA mit einem oder mehreren Proteinen versetzt. Binden diese an die DNA, sind die Bereiche, welche die Bindungsstellen enthalten, vor enzymatischen oder chemischen Veränderungen geschützt. Durch Hinzufügen eines Enzyms (beispielsweise der Endonuclease DNAse I) wird die nicht gebundene DNA verdaut, die Bindungsstellen bleiben als ein Bereich (der 'footprint') erhalten. Ist nur ein einzelnes Protein verwendet worden, kann man durch die Analyse von footprints die Nukleotid-Sequenzen seiner spezifischen Bindungsstellen finden.

Transiente Expression Bei der transienten Expression ('transient expression array') wird ein Gen zusammen mit dem zu untersuchenden Promotor über ein Plasmid in Zellkultur-Zellen eingebracht. Wird es dort transkribiert, kann das entstehende Gen-Produkt nachgewiesen und so die Funktionsfähigkeit des Promotors gezeigt werden. Da viele Gene für diese Vorgehensweise zu lang sind und daher nicht einfach in die Zellkultur übertragen werden können, gibt es auch die Möglichkeit, den Promotor mit einem kürzeren, sogenannten *Reporter-Gen* zu verknüpfen. Um Verfälschungen durch von Natur aus in der Zelle vorhandene Proteine zu vermeiden, kann als Reporter-Gen ein zelltyp-unspezifisches Gen verwendet werden. (Beipielsweise kann die bakterielle β-Galactosidase als Reporter-Gen in Eukaryoten-Zellen dienen.) Mit diesem Verfahren kann also die generelle Funktionalität eines Promotors innerhalb einer Zelle nachgewiesen werden.

DNA-Microarrays Diese auch als „DNA-Chips“ bekannte Methodik ist die jüngste unter den hier aufgeführten Verfahren. Ab Mitte der neunziger Jahre wurde sie für Expressionsstudien und die Identifikation von DNA-Sequenzen in großem Maßstab eingeführt [72]. Auf einem Glaskörper oder einem anderen Trägermaterial werden mikroskopisch kleine DNA-Stücke aufgebracht. Abhängig von der verwendeten Technologie können mehrere hundert bis zu mehreren tausend unterschiedliche Oligonukleotide so auf einer Fläche von ein bis zwei Quadratzentimetern angelegt werden. Die zu testende Substanz wird mit fluoreszierenden Chemikalien markiert und auf den Chip aufgetragen. Entsprechen bestimmte Oligonukleotide dem DNA-Motiv eines Transkriptionsfaktors, der in der Testsubstanz enthalten ist, bindet sich dieser an die auf dem Chip aufgebrachte DNA. Anschließend kann mit einem Laser oder unter dem Mikroskop die Bindung der Test-Substanz an die einzelne Oligonukleotide untersucht werden. Unterschiede in der Fluoreszenz zeigen die hohe Anreicherung, bzw. das Fehlen der Testsubstanz an

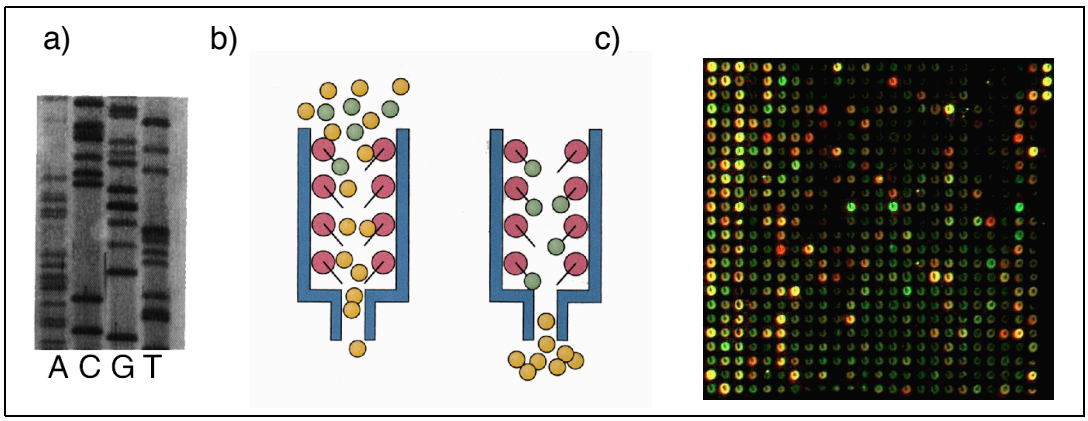

Abbildung 2.7: Labormethoden zur Untersuchung von DNA:
a) Auftrennung radioaktiv markierter Sequenzierungsprodukte durch Gel-Elektrophorese. (aus [37]). Anhand der Banden kann man die Basenreihenfolge des untersuchten DNA-Stücks ablesen.
b) Schematische Darstellung einer Affinitätssäule mit Sepharosekörnern (aus [37]). Gebundene Proteine verbleiben in der Säule, alle anderen werden nach unten ausgeschwemmt.
c) Ein DNA-Microarray (aus [83]). Anhand der Helligkeit der einzelnen Punkte lassen sich unterschiedliche Reaktionen der einzelnen Nukleotidsequenzen ablesen.

einzelnen Stellen auf dem Chip. Microarrays können beispielsweise wie die Sepharose-Bindung zur Untersuchung der Bindung von Transkriptionsfaktoren an bestimmte DNA-Motive dienen; ihre Hauptanwendung besteht in der Messung der Expression von Genen.

Keine der hier vorgestellten Methoden ist für sich allein genommen in der Lage, die exakte Position eines Promotors oder alle seine funktionellen Elemente zu bestimmen. Es sind immer Kombinationen der einzelnen Experimente nötig. Oft müssen diese mehrfach wiederholt werden, um brauchbare Ergebnisse zu liefern. In zunehmendem Maß werden daher algorithmische Verfahren angewendet, die die Experimente entweder ersetzen oder wertvolle Hinweise zur einfacheren, schnelleren Durchführung geben, indem sie zum Beispiel den zu untersuchenden DNA-Bereich einschränken. Im nächsten Abschnitt werden einige dieser Methoden vorgestellt.

2.2.2 Software-Tools

Abgesehen von den Möglichkeiten der Laborforschung gibt es bereits eine große Anzahl von Computerprogrammen, mit deren Hilfe sich DNA-Sequenzen analysieren lassen. Die Sequenzen werden dabei im allgemeinen

als eine lange Zeichenkette aufgefasst, die aus dem Alphabet der vier Basen ($B = \{\texttt{A}, \texttt{C}, \texttt{G}, \texttt{T}\}$) besteht. Versuche, weitere Eigenschaften wie z. B. die dreidimensionale Struktur der DNA mit in die Computeranalysen einzubinden, waren bisher weitgehend erfolglos. Ein Grund dafür ist u. a. die Flexibilität der DNA in der lebenden Zelle. Bei Strukturmessungen mit kristallographischen Methoden geht diese Flexibilität verloren, so dass auf diese Weise gewonnene Informationen über die dreidimensionale Form von Sequenzstücken nur mäßig für weitere Analysen einsetzbar sind.

Für die Computer-unterstützte Suche nach Promotoren gibt es verschiedene Vorgehensweisen. Hier sollen Methoden erklärt werden, die sich an den in Abschnitt 2.1.2 aufgeführten Grundlagen orientieren. Einige weitere Verfahren, denen andere Philosophien zugrundeliegen, finden sich in Abschnitt 3.3. Zunächst werden jetzt Methoden zum Finden von Bindungsstellen vorgestellt, dann soll näher auf Software zur Analyse der Promotoren selbst eingegangen werden.

IUPAC Code Das Auffinden der Motive von Bindungsstellen ist eine der Grundaufgaben bei Sequenzanalysen mit dem Computer. Das naheliegendste elektronische Verfahren beim Suchen von Mustern in der DNA ist die einfache textuelle Suche, bei der das Suchmuster aus einer vorgegebenen Zeichenkette S auf dem Alphabet der vier Basen B besteht. Nachdem die Bindungsstellen variabel sein können, müssen diese Muster jedoch nicht immer eindeutig sein. Für DNA-Abschnitte mit äquivalenter biologischer Funktion können daher unterschiedliche Suchmuster Gültigkeit besitzen.

Eine Möglichkeit, dieser Variabilität Rechnung zu tragen, ist die Einführung eines erweiterten Alphabets für die Suchmuster. Es handelt sich dabei um den IUPAC-Code (siehe Tabelle 2.1), der – zusätzlich zu den vier Basen – Zeichen umfaßt, die für mehrere Elemente aus B stehen können. (Die Auswahl der zusätzlichen Buchstaben ist dabei durch biologische Sachverhalte geprägt, z. B. steht `Y` für die *Py*rimidine `C` und `T`).

Die Benutzung des IUPAC Codes ermöglicht die Erfassung verschiedener individueller Sequenzen durch ein gemeinsames Suchwort. Beispielsweise lassen sich die Sequenzen $S_1 =$ '`ATGTC`' und $S_2 =$ '`AACTC`' beide mit dem Muster $K =$ '`AWSTC`' finden. Allerdings würden auch die Sequenzen $S_3 =$ '`ATCTC`' und $S_4 =$ '`AAGTC`' erkannt werden. Man bezeichnet K als *Konsensus* der Sequenzen S_i. Um solche Konsensussequenzen zu finden, bedient man sich sogenannter

IUPAC-Code	Bedeutung
A	A
B	C, G, T (nicht A)
C	C
D	A, G, T (nicht C)
G	G
H	A, C, T (nicht G)
K	G, T
M	A, C
N	A, C, G, T (alle)
R	A, G
S	C, G
T	T
U	U, T (Uracil $\hat{=}$ Thymin)[11]
V	A, C, G (nicht T)
W	A, T
Y	C, T

Tabelle 2.1: IUPAC Code für die Mustersuche in DNA

'Alignment'-Verfahren. Dabei werden die Sequenzen so gegeneinander verschoben („aliniert"), dass sie an möglichst vielen Positionen übereinstimmen. Die so angeordnete Gruppe von Sequenzen heißt dann *Alignment*. Dabei kann es mehrere optimale Lösungen geben (siehe Abb. 2.8). Je nach verwendeter Methodik ist auch das Einfügen einer (im Regelfall beschränkten) Anzahl von Lücken in die Sequenzen üblich, um so die Übereinstimmung insgesamt zu verbessern.

Während diese Aufgabe für zwei Sequenzen relativ einfach zu lösen ist, ist

```
AAAGGG        AAAGGG
AATGG          AATGG
AAWGG          AAKGG
```

Abbildung 2.8: Bei diesen beiden Sequenzen lassen sich zwei unterschiedliche Alignments finden, die jeweils einen Konsensus liefern, der an einer Stelle ein IUPAC-Zeichen enthält.

sie für mehrere Sequenzen ('multiple alignment') immer noch Gegenstand der

Forschung und es gibt sowohl unterschiedliche Lösungsansätze als auch verschiedene Ansichten darüber, wann ein Alignment optimal ist [48, 49, 13, 78]. Hat man im Labor mehrere Varianten einer Proteinbindungsstelle gefunden, kann man versuchen, diese mit Hilfe eines Alignment-Programms zu einem generellen Muster zusammenzufassen, das sich dann dazu verwenden lässt, in unbekannten Sequenzen nach potentiellen Bindungsstellen für dieses Protein zu suchen.

Matrizen Ein Nachteil solcher mittels Alignment gefundener IUPAC-Muster besteht darin, dass bei der Verwendung von IUPAC-Zeichen keine Gewichtung der einzelnen Nukleotide möglich ist. Finden sich an einer Position im Alignment beipielsweise mehrere '`A`'s aber nur ein '`T`', wird an dieser Stelle im IUPAC-Wort ein '`W`' notiert, was auch der Fall wäre, wenn mehrere '`T`'s und nur ein '`A`' vorkämen. Um solchen Verteilungen besser Rechnung tragen zu können, wurden *Gewichtsmatrizen* ('weight matrices') entwickelt [29, 58]. In diesen Matrizen wird jeder Position im Muster ein vier-dimensionaler Vektor zugeordnet, in dem für jede der vier Basen der jeweilige Anteil eingetragen wird. Bezeichnet l die Länge des zu suchenden Motivs und $p_i(N)$ den Anteil[12] des jeweiligen Nukleotids N an der Position i im Alignment, lassen sich solche Matrizen formal wie folgt definieren[13]:

$$M = \begin{pmatrix} p_1(\texttt{'A'}) & p_2(\texttt{'A'}) & \cdots & p_l(\texttt{'A'}) \\ p_1(\texttt{'C'}) & p_2(\texttt{'C'}) & \cdots & p_l(\texttt{'C'}) \\ p_1(\texttt{'G'}) & p_2(\texttt{'G'}) & \cdots & p_l(\texttt{'G'}) \\ p_1(\texttt{'T'}) & p_2(\texttt{'T'}) & \cdots & p_l(\texttt{'T'}) \end{pmatrix} \tag{2.2}$$

Beim Suchen in einer Sequenz $S = s_1 s_2 s_3 \ldots s_n$ der Länge n bildet man nun an jeder Position $i = 1, \ldots, n - l + 1$ die Summe:

$$score(i) = \sum_{j=1}^{l} p_i(s_{i+j-1}) \tag{2.3}$$

Liegt diese Summe oberhalb eines definierten Schwellwerts, erhält man an der entsprechenden Stelle i einen Treffer. Während man bei den IUPAC-Suchwörtern nur die Information erhält, ob ein Sequenzabschnitt zum Suchwort passt oder nicht, lassen die Matrizen eine qualitative Bewertung zu.

[12] Dieser Anteil kann als relative Häufigkeit oder Wahrscheinlichkeit des Auftretens eines bestimmten Nukleotids aufgefasst werden, daher die in diesem Fall übliche Schreibweise p für die Wahrscheinlichkeit.

[13] manche Algorithmen fügen in das Alignment der Sequenzen Lücken ('gaps') ein, um so ein besseres Resultat zu erzielen. Aus diesem Grund ist die Summe der Anteile der vier Basen in der Praxis nicht immer gleich 1.

Durch die Wahl des Schwellwerts lassen sich die Zahl und die Güte der Treffer beeinflussen. Damit lassen die Matrizen durchaus auch Nukleotide an Positionen zu, die im ihnen zugrundeliegenden Alignment nicht vorkamen, vorausgesetzt der Gesamtwert wird dadurch nicht zu sehr verschlechtert. Diese Möglichkeit der Verallgemeinerung bieten IUPAC-Suchen nicht.

Das Programm *MatInspector* [58] kann dazu verwendet werden, in DNA-Sequenzen sowohl nach IUPAC Wörtern als auch nach Matrizen zu suchen. Dabei wird das oben beschriebene Verfahren für die Matrix-Suche noch erweitert. Überwiegt an einer Position der Matrix ein bestimmtes Nukleotid ($p(b_i) \gg p(b_j), \forall j \neq i$), bzw. wird nur genau ein Nukleotid gefunden, spricht man von einer *gut konservierten Position*. Im Gegensatz dazu kommen an *schlecht konservierten* Positionen alle Nukleotide in etwa gleich häufig vor ($p(b_i) \approx p(b_j) \forall i, j$). Für die MatInspector Software wird für jede Position der Matrix ein Maß für die Konservierung bestimmt, der 'consensus index value'. Dieser Wert liegt zwischen 0 für eine Position mit gleichverteilten Nukleotiden und 100 für eine Stelle an der genau eine der vier Basen auftritt. Er berechnet sich auf der Basis der Shannon'schen Entropie[14]

$$C_i(i) = (100/\ln 5) \times ((\sum_{b \in \{A,C,G,T,gap\}} p_i(b) \times \ln p_i(b)) + \ln 5) \qquad (2.4)$$

wobei

$$p_i(b) \times \ln p_i(b) := 0 \quad \forall \quad p_i(b) = 0;$$
$$p_i(gap) := \textit{Anteil vom Alignment eingefügter Lücken}$$

Der consensus index C_i geht in die Bewertung des scores (2.3) mit ein. Dadurch werden Übereinstimmungen an gut konservierten Stellen noch einmal höher gewichtet. Außerdem wird mit Hilfe dieses Wertes innerhalb der Matrix das vier Positionen breite Fenster mit der insgesamt höchsten Konservierung bestimmt, der sogenannte 'core'. Es wird zunächst nach dem core gesucht und nur dort, wo dieser gefunden wurde, wird mit der vollständigen Matrix auf einen Treffer überprüft, was bei Suchen in langen Sequenzen einen erheblichen Geschwindigkeitsvorteil mit sich bringt. Biologisch gesehen stellt der core den „charakteristischen“ Sequenzabschnitt der zur Matrix gehörenden

[14] Shannons Entropie wird in der Informations- und Codierungstheorie auch als „Maß für die Unsicherheit“ bezeichnet. Je gleichverteilter die Werte einer Zufallsvariable sind, desto größer ist die Unsicherheit über das Ergebnis. Für ein sicheres Ereignis beträgt die Shannon'sche Entropie 0. Mit zunehmender Unsicherheit nimmt ihr Wert zu. Hier wird allerdings die negative Entropie verwendet, um insgesamt Werte zwischen 0 und 100 zu erhalten.

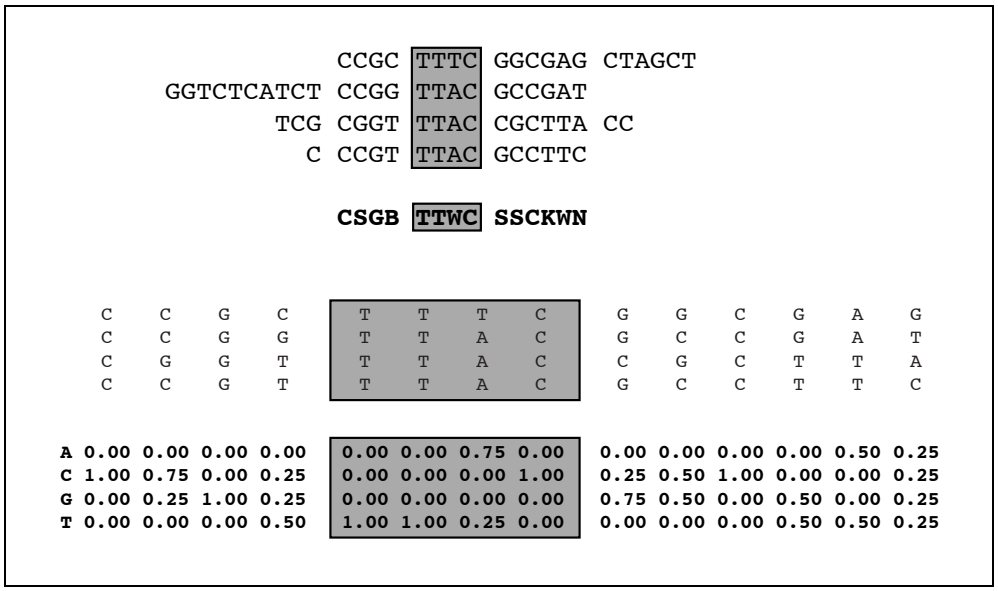

Abbildung 2.9: Ein Beispiel für ein multiples Alignment, aus dem sowohl ein IUPAC-Code, wie auch eine Matrix abgeleitet werden können. Im Alignment werden die Sequenzen zunächst so gegeneinander verschoben, dass ihre positionsweise Ähnlichkeit untereinander möglichst groß ist. Darunter der aus dem Alignment abgeleitete Konsensus als IUPAC-Code. Anstelle des IUPAC-Codes kann auch eine Matrix verwendet werden, die die Anteile der einzelnen Sequenzen genauer berücksichtigt. Bei dem umrahmten Bereich handelt es sich um den 'core', d. h. den Bereich aus vier konsekutiven Basen, der am besten konserviert ist.

Bindungsstellen dar. Bei Matrizen für die TATA-Box lautet er dementsprechend '`TATA`', bei der CCAAT-Box '`CCAA`'.

Modelle Sucht man nur nach Bindungsstellen, werden diese in der DNA auch an Stellen erkannt, die nicht in Promotorbereichen liegen. Für die Suche nach Promotoren werden also zusätzliche Methoden benötigt. Sucht man nach Kombinationen von Transkriptionsfaktoren bzw. den formalen Promotormodellen, werden die Kriterien für die Suche erheblich schärfer. Mit IUPACs oder Matrizen wird man im allgemeinen alle paar tausend Nukleotide, bei kurzen oder sehr allgemein definierten Mustern auch sehr viel öfter fündig. Kombiniert man mehrere Bindungstellen miteinander und führt zusätzlich noch Einschränkungen für die Abstände ein, wie beispielsweise in Definition (2.1), werden die Ergebnisse sehr viel spezifischer. Das Programm *FastM* ermöglicht die Definition von Modellen, die sich dann mit dem *ModelInspector* [36] in DNA Sequenzen suchen lassen. Dabei muss der Aufbau des Modells,

d. h. die beteiligten Transkriptionsfaktorbindungsstellen und die jeweiligen Abstände bereits bekannt sein. Hat man mehrere Promotor-Sequenzen zur Verfügung, in denen man ein einheitliches Modell vermutet, kann das Programm *ModelGenerator* verwendet werden. Auch hier muss eine Definition des Modells vorgegeben werden. Der ModelGenerator überprüft dann die Validität des Modells in dem angegebenen Sequenzset und findet zusätzliche Bindungsstellen, die sich in das Modell inkorporieren lassen. Ohne fundiertes Vorwissen lassen sich jedoch mit diesen Methoden keine Erfolge erzielen.

Statistische Methoden Eine Möglichkeit zu solchem Vorwissen zu gelangen, ist die statistische Untersuchung von Promotoren, deren Verwandtschaft bekannt ist oder generell die Suche nach überrepräsentierten Mustern in regulatorischen Bereichen. Das Programm *GenomeInspector* [59, 60] kann dazu verwendet werden, solche statistischen Untersuchungen durchzuführen. Es sucht beispielsweise nach Überrepräsentationen bestimmter Transkriptionsfaktor-Paare in einem vorgegebenen Sequenz-Set. Auch Transkriptionsfaktoren die sich immer in einer bestimmten Entfernung zu einer definierten Position (etwa dem Transkriptionsstart) häufen, lassen sich mit diesem Programm finden.

Allgemeine Promotorsuche Ein Verfahren um ganz allgemein nach Promotoren zu suchen, ist das Programm *PromoterScan* [57]. Es kombiniert zwei Methoden, die sich bekannter Promotoreigenschaften bedienen. Innerhalb eines Trainingssets aus Promotorsequenzen werden hier Transkriptionsfaktoren gesucht. Im Vergleich zu Nicht-Promotor-Sequenzen wird dann ein Profil für die untersuchten Faktoren erstellt, in dem das unterschiedliche Vorkommen der Bindungsstellen im Positiv- und Negativ-Trainingsset berücksichtigt wird. Unbekannte Sequenzen werden nun anhand dieses Profils auf Bereiche überprüft, die einen Promotor beinhalten könnten. Zusätzlich wird noch nach einer TATA-Box gesucht. Über die Kombination beider Ergebnisse wird dann entschieden, ob es sich um einen Treffer handelt oder nicht. Für Promotoren ohne TATA-Box ist das Verfahren daher nur bedingt geeignet und über die genaue Struktur der gefundenen Promotoren lassen sich mit diesem Programm keine Angaben machen.

Eine ähnliche Methode benutzen die Programme *TSSG* bzw. *TSSW* [73]. Anstelle einer Matrix für die TATA-Box wird hier die Nukleotidzusammensetzung im Bereich des vermuteten Promotors mit Hilfe linearer Diskriminanzanalysen untersucht, um die genaue Position des Transkriptionsstartes zu bestimmen.

2.3 Zusammenfassung

Das in Form der DNA vorliegende Genom enthält Untereinheiten, die Gene, die die Baupläne für die Proteine enthalten. Die Aktivierung der Gene wird durch regulatorische Bereiche, die Promotoren gesteuert. Diese interagieren über bestimmte DNA-Muster mit einer Klasse von Proteinen, den Trankriptionsfaktoren. Die Promotoren sind durch die spezifische Anordnung dieser Bindungsstellen charakterisiert. Sie sind modular aufgebaut; in Promotoren funktionell verwandter Gene finden sich ähnliche Strukturen. Für die biologische Untersuchung der Bindungsstellen und Promotoren existiert eine Reihe von Experimenten, deren Durchführung jedoch oft langwierig und aufwändig ist. Für das Auffinden von Bindungsstellen und die Analyse von Promotoren gibt es eine Reihe von Computer-Programmen. Um Promotorstrukturen zu suchen, ist jedoch in den meisten Fällen biologisches Vorwissen unerläßlich.

Kapitel 3

Künstliche neuronale Netze und Klassifikationsverfahren

Nachdem im letzten Kapitel die biologischen Grundlagen erläutert wurden, wenden wir uns jetzt den für diese Arbeit relevanten informatischen Methoden zu. Bei der Promotorerkennung handelt es sich – wie bei vielen anderen Fragestellungen aus der Molekularbiologie – um ein Problem der Mustererkennung bzw. -zuordnung. Ein bestimmter Bereich der DNA soll gefunden oder, falls das Suchproblem schon gelöst wurde, identifiziert oder zumindest charakterisiert werden.

Ganz allgemein lassen sich solche *Musterklassifikationsprobleme* wie folgt definieren:

> „Eine Grundgesamtheit von Mustern bestehe aus mehreren Teilgesamtheiten (*Klassen*), so dass jedes Muster genau einer Teilgesamtheit angehört. Ziel der Klassifikation ist es, ein Muster aus der Grundgesamtheit, dessen Klassenzugehörigkeit unbekannt ist, aufgrund einer Menge am Muster beobachteter Merkmale derjenigen Klasse zuzuordnen, der es mit größter Wahrscheinlichkeit angehört.“ [69]

Es sind also zwei Dinge für die erfolgreiche *Klassifikation von Mustern*[1] erforderlich:

- Eine Menge von *Merkmalen*, die jedem einzelnen Muster zugeordnet sind. Dabei müssen nicht alle Merkmale für die Klassifikation erforderlich sein.

[1] Im folgenden meist kurz: *Klassifikation*

- Ein Verfahren, das in der Lage ist, aufgrund der Merkmale eines Musters diesem eine bestimmte Klasse zuzuordnen, es zu *klassifizieren*.

Formal betrachtet soll zu einer Menge

$$P = \{p_1, p_2, \ldots, p_n\}$$

von Mustern und einer Menge

$$C = \{c_1, c_2, \ldots, c_l\}$$

von Klassen ein Verfahren

$$cl(P, F_P, C) : P \rightarrow C$$

gefunden werden, das jedem $p_i \in P$ ein $c_j \in C$ zuordnet, wobei es sich bei $F_P = \{F_1, F_2, \ldots, F_n\}$ um die Menge aller den einzelnen p_i zugeordneten Merkmalsmengen $F_i = \{f_{i1}, f_{i2}, \ldots, f_{im}\}$ handelt.[2] Gilt

$$f_{ij} \in \mathbb{R} \forall f_{ij}, \tag{3.1}$$

lassen sich die Merkmale der Muster als Punkte im m-dimensionalen *Merkmalsraum* veranschaulichen. Die Merkmalsmengen werden daher oft auch als *Merkmalsvektoren* $F_i = (f_{i1}, \ldots, f_{im})^T$ aufgefasst. Das Ziel eines Klassifikationsverfahren ist dann, mit Hilfe sogenannter *Trennfunktionen* eine Aufteilung des Merkmalsraums in (möglichst nicht überlappende) Unterräume zu finden, so dass jeder Klasse ein Unterraum zugeordnet werden kann [2, 62]. Bei der Mustererkennung wird dann überprüft, ob sich die Merkmalsvektoren der zu untersuchenden Muster einem dieser Teilräume – und damit einer Klasse – zuordnen lassen. Gelingt dies eindeutig, betrachtet man das Muster als erkannt. Mitunter ist allerdings nur eine nichteindeutige[3] (wenn sich die Klassen überlappen und der Merkmalsvektor in diesem Bereich liegt) oder gar keine (wenn der Merkmalsvektor außerhalb aller einer Klasse zugeordneten Teilräume liegt) Zuordnung möglich [2, 62].

Die Palette der Verfahren zur Mustererkennung bzw. -klassifikation ist vielfältig. Sie beinhaltet u. a. statistische Methoden wie z. B. 'Bayes learning' oder 'maximum-likelihood'-Analysen, Ansätze mit unscharfen Mengen und Anwendungen aus der künstlichen Intelligenz [62]. In den letzten Bereich fallen

[2] Die Bezeichner P,C und F leiten sich dabei von den englischen Wörtern 'pattern', 'class' und 'feature' ab.

[3] In diesem Fall spricht man auch von *zweifelhafter* Erkennung [62].

auch die *künstlichen neuronalen Netze*, die im Rahmen dieser Arbeit näher betrachtet werden sollen.

Eines der ursprünglichen Ziele für die Entwicklung der als künstliche neuronale Netze bekannten Verfahren war die Herausforderung, „'biologische Intelligenz' in künstlichen 'neuronalen Netzwerken' nach[zu]bilden, deren Struktur und Programmierung soweit wie möglich an das Vorbild der Natur angelehnt sein soll." [63] Dabei sollen der Aufbau des Gehirns und die am Denk- bzw. Lernprozess beteiligten Abläufe als Vorbild für die künstliche Nachbildung dienen. Die Strukturierung des Gehirns in durch die *Synapsen* vernetzte Sinneszellen, die *Neuronen*, liefert sowohl die Grundlage für die Konstruktion wie auch den Namen der „künstlichen neuronalen Netze" [63, 64]. Die Einteilung bestimmter Teile des Gehirns in sogenannte *sensorische Bereiche*, bei denen einzelne benachbarte Abschnitte für die Sinneswahrnehmungen an einzelnen benachbarten Körperregionen dienen, liefert die Inspiration für eine weitere Methodik, die *selbstorganisierenden Karten* [39, 63].

Abgesehen von der Möglichkeit der Modellierung biologischer Prozesse haben sich die künstlichen neuronalen Netze als geeignete Verfahren zur Mustererkennung und -klassifikation erwiesen [2]. In dieser Hinsicht sollen sie im Rahmen dieser Arbeit auch schwerpunktmäßig betrachtet werden. Im folgenden werden zunächst einige der unterschiedlichen Verfahren und die ihnen zugrundeliegende Theorie vorgestellt. Anschließend wird dann ihr Einsatz speziell für die Erkennung von Promotoren beschrieben.

3.1 Überwachte Verfahren

Klassifikationsverfahren im allgemeinen und neuronale Netzen im besonderen werden oft in zwei Gruppen unterteilt, die *überwachten* und die *unüberwachten Verfahren*. Bei den überwachten Verfahren wird die Klassifikation dadurch optimiert, dass mit Hilfe von Trainingsdaten, deren Klassenzugehörigkeit bereits bekannt ist, die Parameter des Verfahrens optimiert werden [62].

Ein solches Verfahren besteht im allgemeinen aus den folgenden Schritten:

1. Initialisierung aller Parameter
2. Klassifiziere einen Trainingsdatensatz und ändere die Parameter so, dass die Klassifizierung für diesen Datensatz richtig ist oder zumindest verbessert wird.

3. Prüfe ob eine vorgegebene Abbruchbedingung erfüllt ist, ansonsten fahre mit Schritt 2. fort.

4. Ausgabe aller Parameter und damit der Definition des gefundenen Klassifikators.

Dabei kommen für das Ausmaß der jeweiligen Änderungen und die Abbruchbedingung je nach Verfahren unterschiedliche Vorgehensweisen in Frage. Beispielsweise kann das Training beendet werden, wenn alle Muster einer Testmenge erkannt werden, oder wenn sich die Änderungen der vom Training beeinflussten Parameter nur noch in einem kleinen Bereich bzw. gar nicht mehr bewegen, so dass davon auszugehen ist, dass eine optimale Lösung erreicht ist.

Im Gegensatz dazu versuchen die unüberwachten Verfahren innerhalb der Merkmalsmenge Strukturen zu finden, aufgrund derer sich die einzelnen Muster klassifizieren lassen [2]. In diesem Abschnitt steht die Gruppe der überwachten Verfahren im Mittelpunkt, in Abschnitt (3.2) werden unüberwachte Verfahren vorgestellt.

3.1.1 'Feed-forward'-Netze

Eine der ursprünglichen Motivationen für die Entwicklung der künstlichen neuronalen Netze war, wie bereits in der Einleitung erwähnt, der Wunsch, die Denk-Mechanismen im Gehirn nachzubilden und zu erforschen. Deshalb orientiert sich der Aufbau der Netze an den biologischen Konstruktionselementen des Gehirns. Ähnlich zu den durch Synapsen verbundenen Nervenzellen in der Natur findet man bei den künstlichen Neuronalen Netzen informationsverarbeitende Elemente, die *Neuronen*[4], die untereinander durch *Kanten* verbunden sind, welche die Informationen von einem Element zum nächsten weiterleiten. So wie in der Natur die Neuronen nur dann Signale aussenden, wenn sie dazu angeregt werden, gibt es analog auch bei den Knoten eines künstlichen neuronalen Netzes im allgemeinen eine *Erregung* oder *Aktivierung*, von deren Wert es abhängt, ob sie eine Ausgabe erzeugen und wie „stark“ diese ist [63, 64].

Eines der ersten untersuchten neuronalen Netze war das sogenannte *Perzeptron*, das Anfang der 60er Jahre von Frank Rosenblatt beschrieben wurde

[4] im folgenden oft auch als *Knoten* bezeichnet

[64]. Es besteht aus einem Neuron, das über gewichtete Kanten mit der Eingabe verschaltet ist. Ist n die Anzahl der Eingabekomponenten, dann besteht der Eingabevektor x aus den Komponenten $(\xi_1, \ldots, \xi_n)$, der Gewichtsvektor w aus den Gewichten $(\omega_1, \ldots, \omega_n)$, wobei jeder Kante genau ein Gewicht ω_i zugeordnet ist. Ein vorgegebenes Perzeptron mit dem Schwellwert θ testet die folgende Bedingung:

$$\sum_{i=1}^{n} \omega_i \xi_i > \theta.$$

Der Lernvorgang besteht nun darin, die Gewichte ω_i und den Schwellwert so zu optimieren, dass die Anzahl der korrekt klassifizierten Muster möglichst groß wird. Offensichtlich kann ein Perzeptron auf diese Weise lediglich lineare Klassifikationen verwirklichen (d. h. eine Menge von Mustern nur in zwei Klassen aufteilen, nämlich diejenigen, für die der Schwellwert übertroffen wird und alle übrigen).

Von größerem Interesse sind Netze, in denen mehrere Neuronen dieser Art miteinander verschaltet sind. Unser Interesse beschränkt sich dabei auf Netzarchitekturen, bei denen die Knoten in hintereinanderliegenden Schichten angeordnet sind. Zwischen der Ein- und der Ausgabeschicht können eine oder mehrere sogenannte „verborgene“ Schichten liegen (siehe Abb. 3.1). Die Knoten sind durch gewichtete Kanten miteinander verbunden; innerhalb einer einzelnen Schicht gibt es jedoch keine Verbindungen. Außerdem sind alle Kanten von der Ein- zur Ausgabeschicht hin gerichtet, d. h. die Netze sind zyklenfrei[5]. Die Neuronen berechnen aus den eingehenden Signalen der vorhergehenden Schicht ihre jeweilige Aktivierung. Dazu benötigen sie eine *Integrations-* und eine *Aktivierungsfunktion*. Erstere fasst die eingehenden Signale zusammen, letztere bestimmt aus diesem Wert die Aktivierung des Neurons, die entweder an die nächste Schicht weiter- bzw. ausgegeben wird. Betrachtet man das Perzeptron, handelt es sich bei der Aufsummierung der gewichteten Eingabekomponenten um die Integrationsfunktion; der Vergleich mit dem Schwellwert θ stellt die Aktivierungsfunktion dar [64].

Formal kann man jedes solche Netz aus Neuronen n_i als ein Tupel

$$(E, N_1, \ldots, N_q, A, T)$$

darstellen. E ist die Menge der ρ Eingabestellen, N_i die Mengen der Neuronen in den q verborgenen Schichten, A die Menge der τ Ausgabestellen und T

[5]Daher auch die englische Bezeichnung 'feed-forward'-Netz.

die Menge der gerichteten, gewichteten Kanten. Diese Kanten sind selbst wiederum Tupel $(n_i, n_j, \omega_{ij}^{\eta})$, wobei $n_i \in E \cup \bigcup_{\lambda=1}^{q} N_\lambda$, $n_j \in \bigcup_{\lambda=1}^{q} N_\lambda \cup A$ und die Gewichte $\omega_{ij}^{\eta} \in \mathbb{R}$. Zusätzlich müssen für alle Neuronen $n \in \bigcup_{\lambda=1}^{q} N_\lambda, \cup A$ die Integrationsfunktion $\Psi : \mathbb{R}^n \to \mathbb{R}$ und die Aktivierungsfunktion $\Phi : \mathbb{R} \to \mathbb{R}$ bekannt sein [64]. Die Eingabe wird dann von E ausgehend durch die verborgenen Schichten N_i bis zur Ausgabe A weitergeleitet. Durch die unterschiedlichen Kanten-Gewichte werden bei verschiedenen Eingabe-Mustern jeweils andere Neuronen aktiviert. Das Netz wird mit Mustern, deren Klassenzugehörigkeit bekannt ist, trainiert, um möglichst optimale Werte für die Gewichte zu finden. Mit solchen Netzen lassen sich nicht nur lineare, sondern beliebige stetige Trennfunktionen [63, 2] im Merkmalsraum der zu klassifizierenden Muster realisieren, wodurch dementsprechend nicht-lineare Klasseneinteilungen ermöglicht werden [62, 63].

Ein weit verbreitetes *Trainings-* bzw. *Lernverfahren* für Netze dieser Art ist die *Backpropagation*-Methode.[6]

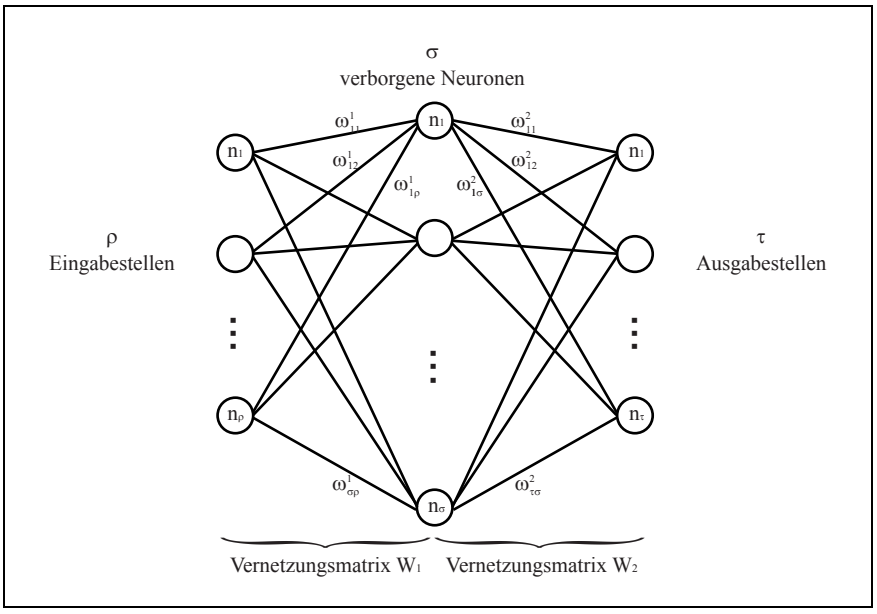

Abbildung 3.1: Struktur eines Feed-forward-Netzes mit einer verborgenen Schicht. Die Vernetzungs- oder Gewichtsmatrix W_1 besteht aus $\rho \times \sigma$ Gewichten ω_{ij}^1 der Kanten zwischen der Eingabe- und der verborgenen Schicht, die Matrix W_2 aus den $\sigma \times \tau$ Gewichten ω_{ij}^2 zwischen der verborgenen und der Ausgabeschicht. Um die Notation zu vereinfachen, sind die Neuronen schichtweise indiziert. [63, 64]

[6] kürzer auch *Backprop-Verfahren* [64]

Bei dieser Methode werden die Gewichte der Kanten zwischen den einzelnen Knoten des Netzes im Verlauf des Trainings angepasst, um so eine optimale Muster-Klassifizierung zu ermöglichen. Anhand der in Abb. 3.1 dargestellten Netzarchitektur soll das Verfahren nun detaillierter vorgestellt werden.

Das betrachtete Netz besteht aus der Ein- und Ausgabeschicht sowie einer Zwischenschicht aus verborgenen Neuronen. Die Ergebnisse lassen sich ohne weiteres auf Netze mit mehreren verborgenen Schichten übertragen. Alle Knoten der verborgenen, bzw. der Ausgabeschicht sind mit allen Knoten der vorangegangenen Schicht durch Kanten verbunden. Diese Verbindungen, bzw. die ihnen zugeordneten Gewichte, lassen sich formal als reellwertige Matrizen W_η betrachten, deren Einträge die jeweiligen Kantengewichte ω_{ij}^η sind.[7] Das Ziel des Lernvorgangs ist, mit Hilfe einer *Trainings-Menge* $\{(s_1, t_1), \ldots, (s_N, t_N)\}, s_i \in \mathbb{R}^\rho, t_i \in \mathbb{R}^\tau$ bestehend aus Paaren einander zugeordneter Ein- und Ausgabevektoren für die Gewichte möglichst optimale Werte zu finden. Im optimalen Fall sollte ein Netz alle Eingabevektoren s_k auf die ihnen zugeordneten Ausgabevektoren t_k abbilden.

Bezeichnet S die Aktivierungsfunktion der Neuronen, läßt sich der Aktivierungsvektor v_i der verborgenen Schicht für das Muster i, zu

$$v_i = S(s_i W_1) \tag{3.2}$$

berechnen.[8] Als Integrationsfunktion wird wie beim Perzeptron einfach die Aufsummierung verwendet. Der Wert für den Ausgabevektor y_i lautet dann

$$y_i = S(v_i W_2) = S(S(s_i W_1) W_2). \tag{3.3}$$

Gibt das Netz anstelle der erwünschten Ergebnisse $t_1, \ldots, t_N$ also die Vektoren $y_1, \ldots, y_N$ aus, definiert man den *quadratischen Fehler* (oder auch ganz allgemein die *Fehlerfunktion*) des Netzes zu:

$$E = \sum_{i=1}^{N} \|t_i - y_i\|^2. \tag{3.4}$$

[7] Sollen zwei Neuronen n_i, n_j nicht durch eine Kante verbunden sein, setzt man $\omega_{ij}^\eta = 0$.

[8] Zur Vereinfachung wird der Vergleich mit einem Schwellwert innerhalb eines Neurons weggelassen. Er läßt sich leicht durch Hinzufügen eines „Schwellwert-Neurons“ in der Eingabe- und den verborgenen Schichten realisieren, das als Aktivierung konstant die 1 liefert und mit allen anderen Neuronen in der darauffolgenden Schicht verbunden ist. Die Kantengewichte entsprechen dann dem negativen Schwellwert des jeweiligen Neurons. Die Optimierung des Schwellwerts entspricht dann der Optimierung dieser Gewichte.

Diese Größe bemisst, wie gut das Netz die Abbildung der Eingangsmuster s_n auf die Ausgaben t_n gelernt hat.

Das Ziel des Backprop-Verfahrens ist es, den Fehler (3.4) zu minimieren. Aus (3.3) und (3.4) sieht man unmittelbar, dass es sich bei $E = E(s_i, t_i, S, W\eta)$ bei festen s_i, t_i, S um eine von den Gewichten ω_{ij}^{η} abhängige Funktion handelt. Die Aufgabe der Methode besteht also darin, Werte für die Kantengewichte zu finden, die (3.4) minimieren. Dazu wird jeweils in Richtung des Gradienten von E nach einem Minimum des Fehlers gesucht. Voraussetzung für die Anwendung dieses *Gradientenabstiegsverfahrens* ist die Existenz der partiellen Ableitungen der Fehlerfunktion für alle Gewichte. Es ist daher nötig, für die Aktivierungsfunktion eine differenzierbare Funktion zu wählen; eine häufig verwendete Funktion ist die Sigmoide[9],

$$S_c : \mathbb{R} \to (0,1), S_c(x) = \frac{1}{1+e^{-cx}}.$$

Da sich die Aktivierungen der einzelnen Neuronen als Verknüpfung einzelner Funktionen berechnen lassen (siehe (3.2) und e (3.3), lassen sich die Komponenten des Gradienten

$$\vec{\nabla} E = \left(\frac{\partial E}{\partial \omega_{11}^1}, \ldots, \frac{\partial E}{\partial \omega_{\sigma\rho}^1}, \frac{\partial E}{\partial \omega_{11}^2}, \ldots, \frac{\partial E}{\partial \omega_{\tau\sigma}^1} \right) \tag{3.5}$$

mit Hilfe der herkömmlichen Kettenregel bestimmen. Diese Werte werden dann verwendet, um die jeweiligen Gewichte entsprechend zu korrigieren.

Insgesamt besteht der Lernvorgang aus der wiederholten Ausführung der folgenden Schritte:

1. Initialisierung aller Gewichte.

2. *Feedforward*-Berechnung:
   Eingabe eines Mustervektors s_i in das Netz und Berechnung der daraus resultierenden Aktivierungen aller Neuronen.

3. *Backpropagation*:
   Aus der Differenz des vorgegebenen Vektors t_i und des ausgegebenen Vektors y_i wird der Wert der Fehlerfunktion E bestimmt. Danach wird

[9] Bei c handelt es sich um die sogenannte Temperaturkonstante, die die Form der Sigmoide beeinflusst. Je größer man c wählt, desto ähnlicher wird die Sigmoide einer Treppenfunktion. Im folgenden soll $c = 1$ vorausgesetzt werden und statt S_c wird einfach S verwendet.

der Fehler komponentenweise rückwärts verteilt, um für jedes Gewicht ω_{ij}^{η} den Wert $\frac{\partial E}{\partial \omega_{ij}^{\eta}}$ zu erhalten.

4. Korrektur der Gewichte:
   Im Folgenden werden dann die einzelnen Gewichte entsprechend des im vorhergehenden Schrittes berechneten Gradienten angepasst. Im allgemeinen wird hier zusätzlich noch eine Komponente α verwendet, welche die Konvergenz des Verfahrens steuert. D. h., die Gewichte werden zu Beginn des Trainings stärker verändert, später ändern sie sich immer weniger. Der neue Wert eines Gewichts berechnet sich also zu $\omega_{ij}^{\eta}(t+1) = \omega_{ij}^{\eta}(t) - \alpha(t), \frac{\partial E}{\partial \omega_{ij}^{\eta}}(t)$, wobei t hier als Index für den jeweiligen Lernzyklus steht.

5. Überprüfen der Abbruchbedingung. Wenn sie noch nicht erfüllt ist, wird mit Schritt 2 fortgefahren.

Das hier beschriebene Verfahren wird als 'on-line'-Methode bezeichnet, weil bei jeder Präsentation eines Eingabevektors die Gewichte verändert werden. Eine andere Möglichkeit ist das 'batch-learning'. Dabei werden zunächst für alle Trainings-Muster die partiellen Ableitungen berechnet, die Änderung der Gewichte erfolgt dann durch Summation über die ermittelten Werte. Während für das 'on-line'-Lernen die Abnahme von α zwingend erforderlich ist, damit das Verfahren ein Minimum für die Fehlerfunktion findet, kann beim 'batch-learning' ein fester Wert für α gewählt werden [62]. Weitere Variationen bestehen darin, die Lernkonstante in Abhängigkeit von der Fehlerfunktion jedesmal neu zu bestimmen (*Backpropagation mit variabler Schrittlänge*), vorangegangene Änderungen der Gewichte mit zu berücksichtigen (*Backpropagation mit Impuls*) oder die zweiten partiellen Ableitungen des Fehlers zu bestimmen und die so erhaltenen Informationen über die Krümmung der Funktion zur Beschleunigung des Lernens zu nutzen (*Quickprop*) [64].

Abstrakt gesehen handelt es sich bei den Backpropagation-Netzen um Funktionen, die eine Abbildung vom Eingabe- in den Ausgaberaum darstellen. Durch das Training soll diejenige Funktion gefunden werden, welche die Eingangsmuster möglichst genau auf die ihnen zugeordneten Ausgabevektoren abbildet. Allerdings sollte das Netz später in der Lage sein, nicht nur die beim Lernen verwendeten Muster richtig zuzuordnen, sondern auch bisher unbekannte Muster korrekt zu klassifizieren.[10] Bei der Auswahl der Netzarchitektur und der Gestaltung des Lernprozesses sollte beiden Ansprüchen

[10] Man bezeichnet dies als die *Verallgemeinerungsfähigkeit* des Netzes [64].

genügt werden. Jedoch gibt es „kein Patentrezept für die Bestimmung der optimalen Parameteranzahl eines Netzes. Sie hängt vom jeweiligen Problem ab“ [64]. Um die Verallgemeinerungsfähigkeit eines Netzes zu überprüfen, benutzt man in der Praxis eine *Test-Menge* von Mustern, für die die Ausgabe des Netzes bekannt ist, die aber nicht in der Trainings-Menge enthalten sind.

Ein Nachteil des Verfahrens in der hier vorgestellten Form stellt die Einschränkung seiner Anwendbarkeit auf einzelne Muster dar. Dennoch wurde es, wie später noch dargestellt wird, schon für die Promotorerkennung eingesetzt, ohne dass dabei den strukturelle Aufbau der Promotoren aus einzelnen Elementen zu berücksichtigen. Für das Lernen von solchen Musterfolgen muss man die Netzwerk-Architektur um zusätzliche Komponenten erweitern. Dieser Ansatz soll im nächsten Abschnitt näher erläutert werden.

3.1.2 Time-Delay-Netze

Eine Variante der Feedforward-Netze, die es ermöglicht, wiederkehrende Folgen bestimmter Muster in einer Sequenz zu lernen, sind die 'Time Delay Neural Networks' (TDNNs)[11]. Für die Erkennung bestimmter Phoneme wurde dieser Typ eines künstlichen neuronalen Netzes von Waibel et al. 1989 eingeführt [77]. Für die Spracherkennung geeignete Feedforward-Netze sollten danach den folgenden fünf Eigenschaften genügen:

> 1. Sie sollten aus mehreren Schichten bestehen und eine ausreichende Anzahl an Verbindungen zwischen den einzelnen Knoten in jeder dieser Schichten besitzen, um so zu gewährleisten, dass das Netz in der Lage ist, komplexe nichtlineare Entscheidungsfunktionen zu lernen.
> 2. Das Netz sollte die Fähigkeit besitzen, Beziehungen zwischen einzelnen Ereignissen im Zeitablauf wiederzugeben.
> 3. Die Merkmale oder Abstraktionen, die vom Netz gelernt werden, sollten gegenüber Verschiebungen in der Zeit invariant sein.
> 4. Der Lernalgorithmus sollte kein exaktes zeitliches Alignment der zu lernenden Abstraktionen erfordern.
> 5. Die Zahl der Gewichte im Netzwerk sollte, verglichen mit der Anzahl der Trainingsdaten, klein sein, damit das Netzwerk gezwungen ist, für Gemeinsamkeiten in den Daten die gleiche Codierung zu verwenden.

Ein Time-Delay Netzwerk fügt im Vergleich zu einem herkömmlichen Feedforward-Netz zusätzliche gewichtete Kanten und Zwischenspeicher-Elemente für

[11] Im folgenden auch einfach: Time-Delay Netzwerke

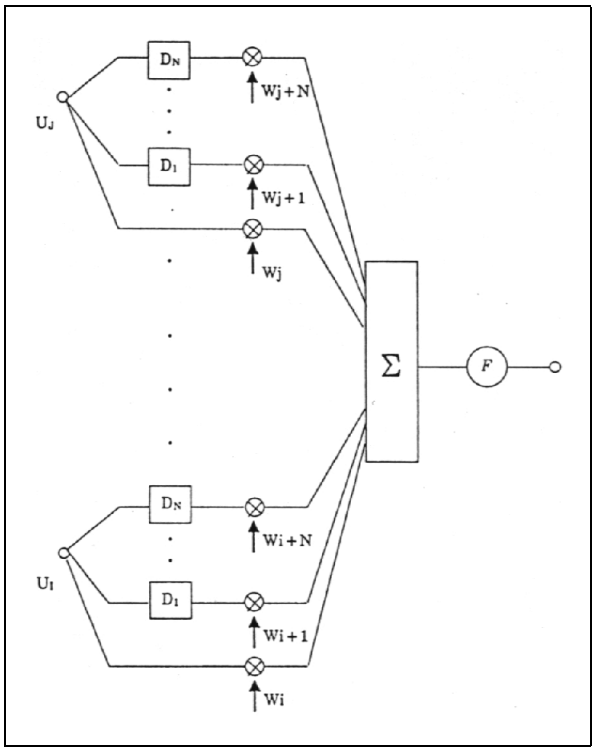

Abbildung 3.2: Ein einzelner TDNN Knoten (aus [77]). U_1 bis U_J sind die Knoten ('units'), deren Zustände übergeben werden. Zusätzlich zur aktuellen Aktivierung werden auch zurückliegende Signale (mit D_1 bis D_N bezeichnet) berücksichtigt. Anstelle von j Gewichten wie bei herkömmlichen Feedforward-Verfahren werden $j \times (N + 1)$ Gewichte benötigt.

vorangegangene Zustände der Eingangsknoten ein. Damit ist es in der Lage, mehrere Eingangsmuster gleichzeitig zu erfassen, und so prägnante Folgen von Mustern in den Trainingsdaten zu lernen.

Das hier vorgestellte TDNN ist ein Mehrschichten-Netzwerk mit Verbindungen zwischen jeweils zwei aufeinanderfolgenden Einzelschichten. Es soll dazu dienen, die gesprochenen Konsonanten 'B', 'D' und 'G' zu erkennen. Anstelle eines einzelnen Vektors wird, im Gegensatz zum üblichen Verfahren, in der Eingangsschicht eine Sequenz von Mustern präsentiert. In den einzelnen Knoten werden für die Berechnung der Aktivierung nicht nur das jeweils aktuelle Signal der Knoten aus der vorhergehenden Schicht berücksichtigt, sondern auch die Signale der unmittelbar zurückliegenden Präsentationen (siehe Abb. 3.2).

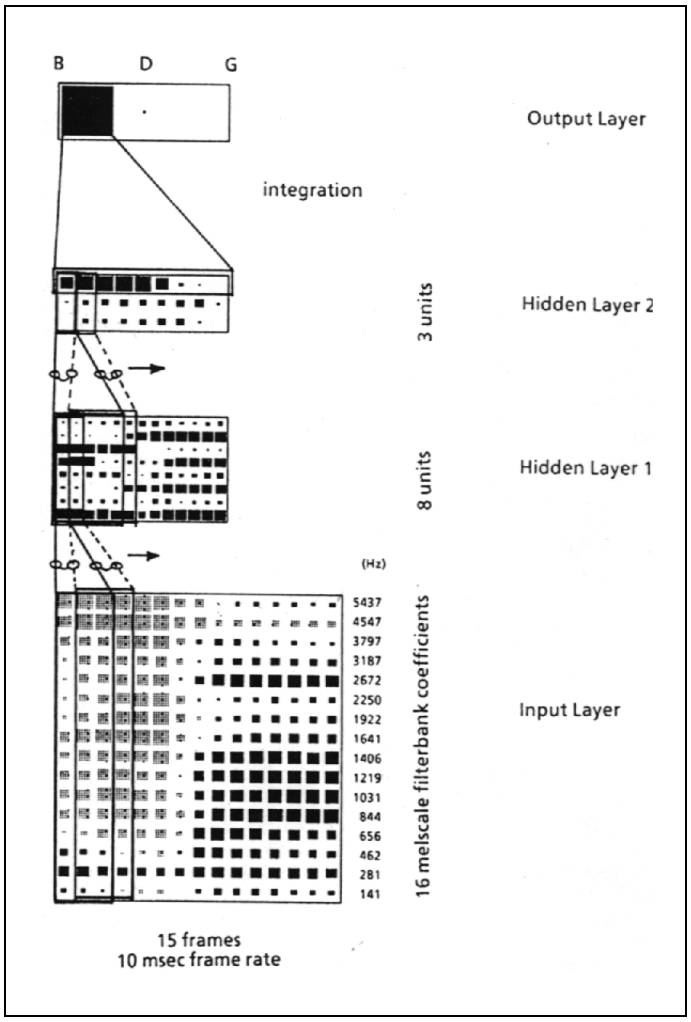

Abbildung 3.3: Die Architektur des TDNN aus [77]. Die drei Phoneme 'B', 'D' und 'G' können unterschieden werden. Das Netz besitzt neben der Ein- und der Ausgabeschicht zwei Zwischenschichten. Aus der Eingabeschicht werd en der aktuelle und die zwei zurückliegenden Zustände berücksichtigt, aus der ersten Zwischenschicht werden insgesamt fünf Zustände nach oben weitergereicht.

Anschaulich betrachtet wird immer ein ganzer Ausschnitt (ein „Fenster“) aus der Musterfolge präsentiert. Die Anzahl der mitbewerteten zurückliegenden Signale ist für alle Knoten einer Schicht gleich, innerhalb des Netzes kann sie jedoch variieren. Wie in Abb. 3.3 zu sehen ist, wird hier zunächst ein Fenster von drei, dann von fünf Signalen berücksichtigt. In die Entscheidung für einen der drei Konsonanten gehen die letzten neun der in der darunterliegenden Schicht aufgetretenen Signale mit ein. Aufgrund dieser Fenstereinteilung

ist das TDNN in der Lage, Musterfolgen unabhängig von ihrer Position innerhalb der Gesamtsequenz zu lernen.

Im Vergleich zu üblichen Feedforward-Netzen benötigt ein Time Delay Netzwerk deutlich mehr Speicher für die Berücksichtigung der dem aktuellen Zustand jeweils vorangegangenen Knoten-Aktivierungen. Für jeden Knoten U_i, der die Signale von j anderen Knoten berücksichtigt, erhöht sich die Anzahl der benötigten Gewichte auf $j \times (n + 1)$, wobei n die Anzahl der zurückliegenden Signale bezeichnet. Im Beipiel würden bei „normaler“ Vernetzung zwischen der ersten und zweiten Schicht insgesamt $16 \times 8 = 128$ Gewichte benötigt; mit Berücksichtigung der zwei zurückliegenden Zustände erhöht sich die Zahl auf 384. Dementsprechend nimmt das Training eines TDNNs je nach Konstruktion deutlich mehr Zeit in Anspruch als das eines herkömmlichen Netzes, weil auch die Berechnung des Gradienten der Fehlerfunktion bei der Backpropagation aufwendiger wird.

Die Möglichkeit, Sequenzen von Mustern erkennen zu können, macht das Verfahrens allerdings für eine Reihe von Anwendungen interessant, so dass der höhere Aufwand durchaus gerechtfertigt ist. Außer zur hier vorgestellten Aufgabe der Spracherkennung kann es beispielsweise auch zur Aktienkursanalyse oder, wie wir noch zeigen werden, für die Klassifizierung von Muster-Folgen in DNA verwendet werden.

Im folgenden wird jedoch zunächst noch eine andere Klasse von Methoden vorgestellt, die nicht zur Gruppe der überwachten Lernverfahren gehört, sondern auf einem völlig anderen Paradigma basiert. Sie stellen damit, wie wir später noch sehen werden, im Bereich der Promotorerkennung eine interessante Alternative zu den überwachten Verfahren dar.

3.2 Verfahren zur Selbstorganisation

Während bei den im vorangegangenen Abschnitt beschriebenen Methoden die Anzahl der Klassen und zumindest ein Teil der in ihnen enthaltenen Muster im allgemeinen schon im Voraus bekannt ist, gibt es zahlreiche Klassifikationsprobleme, bei denen dies nicht der Fall ist [62]. Ohne eine bekannte Klasseneinteilung lässt sich jedoch keine Vorgabe für die Trainingsmuster angeben und damit auch keine darauf basierende Fehlerfunktion finden, wie sie für das Training von Backprop-Netzen verwendet wird. Dementsprechend ist man gezwungen, auf andere Methoden auszuweichen. Im Gegensatz zu

den überwachten Verfahren spricht man von den *unüberwachten Verfahren*. Häufig besteht das Ziel dieser Methoden darin, den meist hochdimensionalen Merkmalsraum so zu transformieren, dass darin enthaltene Strukturen (etwa eine charakteristische Anordnung der Muster) leicht erkannt und bewertet werden können; man spricht in diesen Fällen von *Visualisierungsverfahren*. Eine andere grundlegende Vorgehensweise besteht darin, solche Strukturen dazu zu verwenden, den Merkmalsraum automatisch in Klassen von Mustern mit ähnlichen Merkmalen aufzuteilen[12].

Beispiele für Methoden dieser Art sind *Dendrogramme*, die *Hauptkomponentenanalyse* [62] oder *Sammon's mapping* [68]. Bei Sammon's mapping wird ein hochdimensionaler Merkmalsraum so in einen niedriger dimensionalen Ausgaberaum abgebildet werden, dass die lokale Anordnung der Muster im Merkmalsraum in der Projektion – soweit möglich – erhalten bleibt. Hat man eine endliche Menge n-dimensionaler Muster p_i lässt sich mit einer beliebigen Metrik der Abstand $d_{ij} = d(p_i, p_j)$ zwischen den Mustern p_i und p_j definieren. Auf der Menge der k-dimensionalen ($k < n$) zugehörigen Projektionen q_i seien die Abstände $d^*_{ij} = d^*(q_i, q_j)$ zwischen zwei Punkten q_i und q_j ebenfalls durch eine Metrik gegeben. Dabei werden die Projektionen q_i so gewählt, dass die Funktion

$$E(d, d^*) = \sum_{i \neq j} \frac{(d_{ij} - d^*_{ij})^2}{d_{ij}}$$

lokal minimiert wird. Im allgemeinen wählt man $k = 2$. Diese zweidimensionale Ausgabe kann dann visuell auf Strukturen oder Cluster untersucht werden.

Von den Clustering-Methoden werden einige — hauptsächlich aufgrund ihrer Motivation [62] — ebenfalls den Neuronalen Netzen zugerechnet. Es gibt Neuronen oder Knoten und man findet auch hier wieder das Prinzip des Lernens einer Klassifikation durch das Training dieser Elemente. Das Ziel ist dabei, den einzelnen Clustern eines dieser Elemente als *Repräsentanten* zuzuordnen. Da sich die Anordnung bzw. Organisation dieser Einheiten im Zuge des Verfahrens oft automatisch („wie von selbst“) ergibt, bezeichnet man diese Methoden auch als *selbstorganisierende Verfahren* oder *Verfahren zur Selbstorganisation* [62, 2]. Im Gegensatz zu den überwachten Methoden, bei denen nach der Überprüfung der Klassifizierung eines Musters der Lern-

[12] Im Englischen bezeichnet man solche lokalen Häufungen von Mustern als '*cluster*'; auch in der deutschen Literatur findet man in der Literatur die Bezeichnung *Clustering-Verfahren* [64].

vorgang im allgemeinen darin besteht, dass eine ganze Reihe von Gewichten im gesamten Netz angepasst werden, liegt bei den selbstorganisierenden Verfahren oft das Prinzip des *kompetitiven Lernens* zugrunde. Bei Rumelhart und Zipser [66] werden die grundlegenden Komponenten eines kompetitiven Lernverfahrens wie folgt beschrieben:

1. Man beginnt mit einer Menge von Knoten, die alle gleich sind, mit Ausnahme eines zufällig verteilten Parameters, der dafür sorgt, dass jeder Knoten auf verschiedene Muster einer Eingangsmenge unterschiedlich reagiert.

2. Die „Stärke“ jedes Knotens ist begrenzt.

3. Die Knoten stehen im Wettbewerb („Kompetition“) um das Recht, auf eine Teilmenge von Mustern der Eingangsmenge zu reagieren.

In der Praxis wird in jedem Trainingsschritt ein einzelnes Muster aus der Eingangsmenge präsentiert. Dann wird der *Gewinner-Knoten*, d. h. derjenige Knoten ermittelt, der am stärksten auf dieses Muster reagiert. Dieser kann nun seine „Chancen“ im Wettbewerb verbessern, indem er seinen Parameter so verändert, dass er auf die jeweilige Teilmenge stärker reagiert als die anderen Knoten.

Auf diese Weise spezialisieren sich einzelne Knoten auf Teilmengen ähnlicher Muster und können so für Klassifikationsaufgaben verwendet werden.

Von den verschiedenen Methoden, die sich das Prinzip der Selbstorganisation bzw. des kompetitiven Lernens zu eigen machen [21, 23, 76] sind die selbstorganisierenden Karten[13] von Teuvo Kohonen [38, 39] wohl das am meisten genutzte Verfahren, auf das im folgenden Abschnitt genauer eingegangen wird.

3.2.1 Reguläre SOMs

Bei Kohonens *selbstorganisierenden Karten* handelt es sich streng genommen um eine Methode zur Visualisierung von Strukturen innerhalb hochdimensionaler Merkmalsräume [39]. Im allgemeinen wird das Verfahren jedoch den künstlichen neuronalen Netzen zugerechnet, denn obwohl das verwendete Prinzip des Lernens sich von den in 3.1.1 beschriebenen Verfahren grundlegend unterscheidet, findet man eine ähnliche biologische Grundlage und

[13] im folgenden kurz als *Kohonenkarten* oder *SOMs* (’**S**elf **O**rganizing **M**ap’s) bezeichnet

dementsprechende Konstruktionselemente.

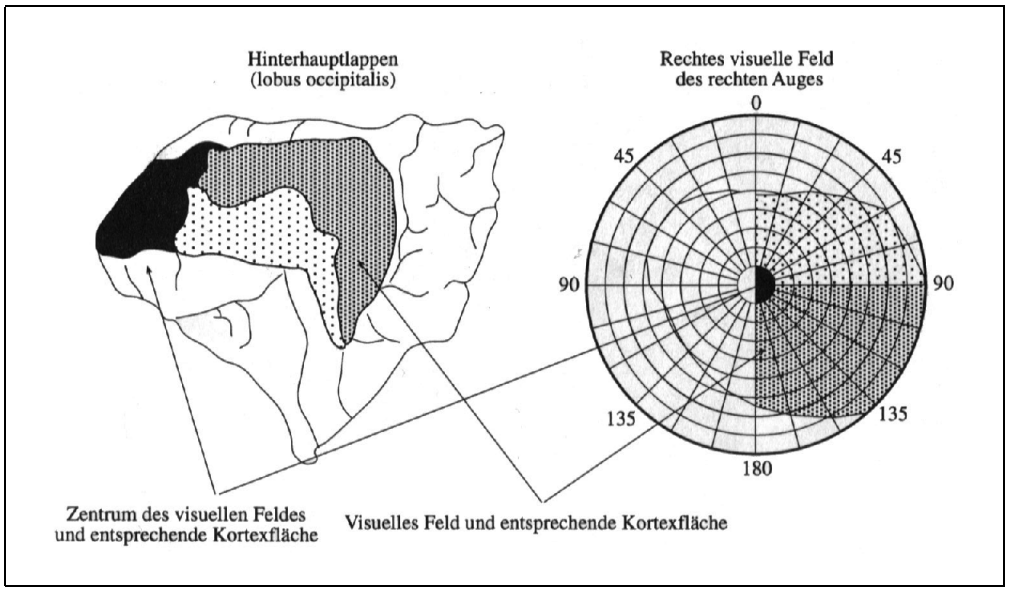

Abbildung 3.4: Die Kartierung des visuellen Feldes im menschlichen Kortex. Benachbarte Bereiche im Sichtfeld werden durch benachbarte Bereiche im Kortex repräsentiert. Die für das Zentrum des Sichtfeldes (kleiner Kreis in der Mitte) zuständige Region des Kortex ist vergleichsweise groß, um eine feinere Verarbeitung der dort wahrgenommenen Information zu gewährleisten. (Aus [64, 22])

Wie bei den Netzen für überwachtes Lernen dient auch hier das Gehirn als biologische Motivation. Doch obwohl es bei den SOMs ebenfalls „Knoten“ gibt, die „trainiert“ werden, um zu „lernen“, entspricht der Ansatz nicht dem konstruktivistischen Modell der Backprop-Netze, mit denen der Aufbau des Gehirns aus Neuronen und Synapsen nachgebildet wird. Stattdessen soll die strukturelle Organisation des Gehirns nachgeahmt werden. Beispielsweise ist für die Verarbeitung der Eindrücke, die beim Sehen vermittelt werden (Farbe, Form, Struktur, Lage einzelner Objekte), innerhalb des Gehirns der *visuelle Kortex* zuständig. Im Gegensatz zur Hochdimensionalität der zu verarbeitenden Daten handelt es sich dabei jedoch um einen im wesentlichen zwei-dimensionalen Bereich in der Gehirnrinde [64]. Innerhalb dieses Bereiches werden benachbarte Gebiete des Sichtfeldes durch benachbarte Regionen verarbeitet; im Sichtfeld nebeneinanderliegende Wahrnehmungen aus der hochdimensionalen Umgebung werden also möglichst „nachbarschaftserhaltend“ im Gehirn verarbeitet. Der Verarbeitung der besonders wichtigen Informationen im Zentrum des Sichtfelds ist dabei ein überproportional

großer Teil des visuellen Kortex gewidmet.

Weitere Beispiele für strukturelle Organisation im Gehirn sind der für den Tastsinn zuständige *somatosensorische* Bereich der Hirnrinde und der *motorische* Kortex, der für die Bewegungssteuerung benötigt wird. Wieder findet man das Prinzip der Topologieerhaltung: Die Verarbeitung von Signalen benachbarter Körperteile geschieht in benachbarten Regionen des Gehirns. Außerdem gibt es auch hier größere Regionen, etwa für die offensichtlich aufwändigere Informationsverarbeitung beim Bewegen der Finger. Sowohl

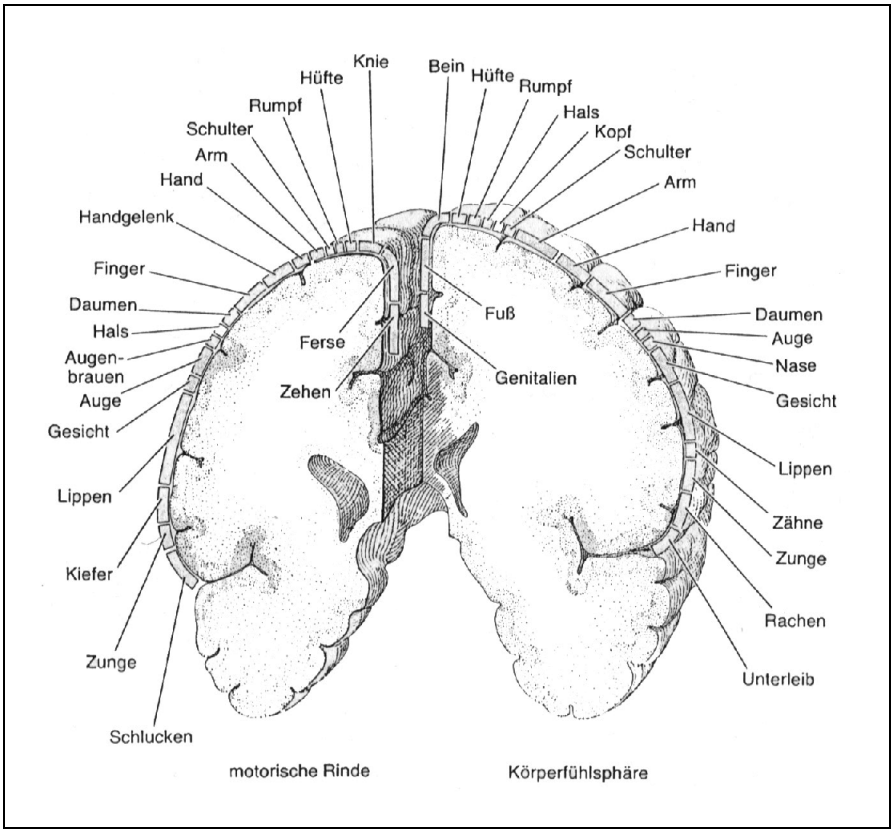

Abbildung 3.5: Ein Schnitt durch den somatosensorischen (für den Tastsinn zuständige) und den motorischen (für die Steuerung der Bewegungen zuständigen) Bereich der Hirnrinde. Entsprechend der räumlichen Aufteilung der Körperteile sind auch hier benachbarte Regionen des Kortex für die Wahrnehmungssignale bzw. Steuerung benachbarter Körperteile zuständig (aus [42, 64]).

das Prinzip der *Topologieerhaltung* als auch das der genaueren Differenzierung von Bereichen mit höherer Informationsdichte findet sich also bei den Kohonenkarten wieder.

Über die Verwendung dieser Ideen hinaus ähnelt auch die Konstruktion der selbstorganisierenden Karten biologischen neuralen Strukturen. Als Elemente dienen wieder miteinander verschaltete Knoten (bzw. Neuronen). Im Gegensatz zu den Feedforward-Netzen, bei denen diese in mehreren Schichten hintereinander angeordnet sind, die durch gewichtete Kanten miteinander verbunden werden, liegen bei den Kohonenkarten alle Neuronen innerhalb einer Schicht. Außerdem dient die Verbindung durch Kanten lediglich der eindeutigen Anordnung der einzelnen Neuronen innerhalb der Karte; die Gewichte sind direkt mit den Neuronen verknüpft.

So wie im somatosensorischen Kortex bestimmte Neuronen Bereiche des Körpers repräsentieren, spricht man auch von den Knoten der Kohonenkarte als *Repräsentanten* für einen ihnen jeweils zugeordneten Teilbereich des Merkmalsraums. Im Unterschied zum allgemeinen Prinzip des kompetitiven Lernens soll bei den Kohonenkarten der Merkmalsraum möglichst topologieerhaltend auf der zweidimensionalen Schicht der Karte nachgebildet werden. Gebiete mit hoher Musterdichte werden dabei im allgemeinen durch mehrere Knoten repräsentiert.

Für die Eigenschaft der Topologieerhaltung ist es notwendig, das Konzept des kompetitiven Lernens um den Begriff der „Nachbarschaft“ zwischen einzelnen Knoten zu erweitern. Während des Lernvorgangs werden nicht nur die Parameter (der Gewichtsvektor) des Gewinner-Knotens sondern auch die seiner Nachbarn adaptiert. Auf diese Weise wird eine ähnliche Reaktion benachbarter Neuronen auf benachbarte Eingabemuster erreicht.

Formal betrachtet handelt es sich bei den Kohonenkarten um eine nichtlineare Abbildung aus dem m-dimensionalen Merkmalsraum auf einen im Regelfall ein- oder zweidimensionalen Verbund von Knoten n_i.[14] Jedem dieser Neuronen ist ein *Referenz-* oder *Gewichtsvektor* $w_i = (\omega_1, \ldots, \omega_m)^T \in \mathbb{R}^m$ zugeordnet. Für jeden Eingabevektor x_i wird dann beim Training der Knoten n_c mit minimalem Abstand gesucht, der *Gewinner-Knoten*.[15]

$$\|x - n_c\| = min_i \|x - n_i\| \tag{3.6}$$

Während des Trainings werden auch die Knoten in den Lernvorgang mit-

[14] Die Form der Anordnung ist dabei zweitrangig, lediglich die Definition der Nachbarschaften muss möglich sein. Im zweidimensionalen Fall sind rechteckige oder hexagonale Anordnungen der Neuronen üblich.

[15] Analog zur Biologie wird hier auch oft vom Knoten oder Neuron mit der maximalen *Erregung* gesprochen [64].

einbezogen, die innerhalb des Verbundes topologisch nahe beim Gewinner-Knoten n_c liegen. Alle diese Knoten lernen dann vom gleichen Eingabevektor x. Dies führt zu einer ausgeglichenen Anpassung der Gewichtsvektoren innerhalb dieses Bereichs und letztendlich zur globalen Ordnung innerhalb der Karte [39]. Die Anpassung[16] der Gewichtsvektoren erfolgt dabei nach der Formel:

$$w_i(t+1) = w_i(t) + h_{ci}(t)[x(t) - w_i(t)], 0 \leq t < T. \tag{3.7}$$

Dabei bezeichnet t den jeweiligen Lernschritt[17], h_{ci} die *Nachbarschaftsfunktion*. Für die Konvergenz des Verfahrens ist $h_{ci} \to 0$ wenn $t \to \infty$ erforderlich. Im allgemeinen gilt außerdem mit $h_{ci}(t) = h(\|r_c - r_i\|, t)$, dass $h_{ci} \to 0$ für $\|r_c - r_i\| \to \infty$, wobei $r_i, r_c \in \mathbb{R}^2$ die Ortsvektoren der Knoten n_c, n_i auf der Karte darstellen. Eine oft getroffene Wahl für die Nachbarschaftsfunktion ist:

$$h_{ci}(t) = \begin{cases} \alpha(t) & : \quad i \in N_c(t) \\ 0 & : \quad i \notin N_c(t) \end{cases} \tag{3.8}$$

Dabei ist $N_c(t)$ die Menge der Indizes i aller Nachbarknoten n_i des Gewinners n_c mit $\|r_i - r_c\| \leq r(t), r(t) \geq 0$, die im allgemeinen mit zunehmendem t abnimmt ($r(t) \to 0$ für $t \to \infty$, $r_0 = r(0)$ bezeichnet dann den *Anfangsradius*). $\alpha(t)$ wird als *Lernrate* bezeichnet; auch hier gilt $\alpha(t) \to 0$ für $t \to \infty$. Eine weitere Möglichkeit ist die Form einer Gaußschen Glocke:

$$h_{ci}(t) = \frac{\alpha(t) \cdot \exp(-\|r_c - r_i\|^2}{2\sigma^2(t)}.) \tag{3.9}$$

wobei α wieder die Lernrate bezeichnet und $\sigma(t)$ die Größe der Nachbarschaft bestimmt (analog zum Anfangsradius r von N_c in Gleichung 3.8).

Der Algorithmus besteht dann aus den folgenden Schritten:

1. Bestimme die Startwerte $w_i(0)$ der Gewichtsvektoren, eine Nachbarschaftsfunktion $h_{ci}(0)$, und entsprechend r und $N_c(0)$ oder $\sigma(0)$.

2. Wähle aus der Menge der Merkmalsvektoren der Eingabemuster zufällig einen Vektor $x(t)$ aus.

3. Ermittele den Gewinner-Knoten n_c

4. Aktualisiere die Gewichtsvektoren aller Neuronen nach der Vorschrift 3.7.

[16] auch *Adaption*
[17] auch *Zyklus*

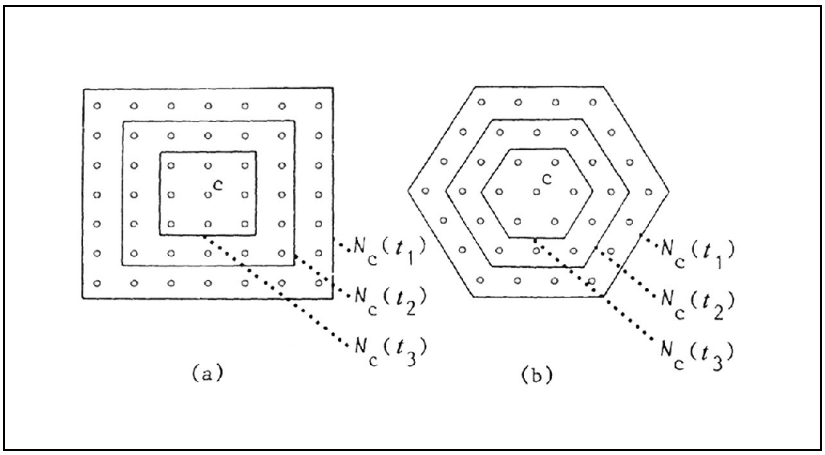

Abbildung 3.6: Schematische Darstellung der Nachbarschaften des Gewinner-Knotens c während des Trainings bei rechteckiger (a) und hexagonaler (b) Anordnung der Neuronen auf der Karte. Dabei gilt $t_1 < t_2 < t_3$. [39]

5. Falls bereits $t = T$ Lernschritte erfolgt sind, brich den Algorithmus ab. Anderenfalls passe $h_{ci}(t)$ an und fahre mit Schritt 2 fort.

Für die Bestimmung der Werte für Lernrate, Anzahl der Zyklen oder den Anfangsradius in der Praxis gibt es nur Richtlinien, keine fest vorgeschriebenen Werte [38]. Im allgemeinen benutzt man für die ersten 1.000 bis 5.000 Schritte eine Lernrate im Bereich von $0,1$ bis $0,8$ und große Werte für den Nachbarschaftsradius (bis zum halben Durchmesser der Karte). In dieser Phase werden die Gewichtsvektoren angeordnet. Danach folgt die Phase der Feineinstellung, die deutlich aufwändiger ist (etwa 50.000 bis 100.000 Zyklen). Hier wird nur noch eine kleine Lernrate von etwa $0,01$ bis $0,1$ verwendet und der Radius der Nachbarschaft relativ klein (zwischen 2 und 5) gehalten. Insgesamt empfiehlt Kohonen, als Zyklenzahl mindestens das 500-fache der Anzahl der Knoten der Karte zu verwenden. Da meist deutlich weniger Trainingsvektoren zur Verfügung stehen, werden diese mehrfach benutzt. Dabei können sie für die Ermittlung des Gewinner-Knotens entweder jedesmal zufällig ausgewählt werden, oder die Trainings-Menge wird zyklisch mehrmals durchlaufen.

Für die Anzahl der Knoten gilt, dass sie je nach Verwendung zwischen der Hälfte und der Gesamt-Anzahl der zu lernenden Muster liegt. Im letzteren Fall kann theoretisch jeder Knoten ein Muster repräsentieren, so dass Karten mit einer höheren Anzahl von Neuronen wenig sinnvoll sind.

Auch für die Startwerte der Gewichtsvektoren gibt es mehrere Möglichkeiten [39]. Zum einen können sie zufällig aus dem $\mathbb{R}^m$ gewählt werden. Eine andere Möglichkeit ist, aus den Merkmalsvektoren der Trainings-Menge einige per Zufall auszuwählen. In diesem Fall entspricht die Verteilung der Gewichtsvektoren in etwa der Verteilung der Muster im Merkmalsraum, jedoch sind sie topologisch gesehen noch ungeordnet. Eine dritte Variante ist die *lineare Initialisierung*. Hier werden zunächst die beiden größten Eigenwerte der Autokorrelationsmatrix aller Trainingsmuster x berechnet. Die dazugehörigen Eigenvektoren spannen eine Ebene im $\mathbb{R}^m$ auf. Auf dieser Ebene wird ein rechteckiges Gitter angelegt, dessen horizontale und vertikale Ausdehnung proportional zu den beiden berechneten Eigenwerten ist, und dessen Mittelpunkt im Zentrum der Verteilung der Merkmalsvektoren x liegt. Die Gewichtsvektoren der Neuronen der Karte werden dann mit den Gitterpunkten belegt. Da hierdurch schon eine gewisse topologische Ordnung vorgegeben ist, und auch die Verteilung der Gewichtsvektoren in etwa derjenigen der Merkmalsvektoren entspricht, kann hier beim Training die Anordnungsphase entfallen und gleich mit der Feineinstellung begonnen werden.

3.2.2 Modifizierte SOMs

Ausgehend von dem im letzten Abschnitt beschriebenen Verfahren von Kohonen, gibt es eine Reihe von Varianten. Teilweise sollen diese dazu dienen, Schwächen des ursprünglichen Verfahrens zu beheben, teilweise handelt es sich um Erweiterungen des Algorithmus für spezielle Anwendungen.

Toroidale Kohonen-Karten Ein Problem bei der Verwendung der kanonischen Form von Kohonens Algorithmus ist die Begrenzung der Karte. Da bei der klassischen Vorgehensweise die Knoten am Rand und in den Ecken weniger Nachbarknoten besitzen als die im Inneren der Karte, tendieren die Gewichtsvektoren am Rand dazu, sich in das Innere des Merkmalsraums zu orientieren [39]. Um diese Randeffekte zu umgehen, können zusätzliche Verbindungen zwischen den Randknoten einfügt werden, so dass die Knoten am oberen Rand der Karte mit denen am unteren Rand verknüpft sind. Analog lassen sich die Knoten an den seitlichen Rändern miteinander verbinden. Topologisch gesehen entsteht so aus der planaren Karte ein Torus. Jeder Knoten auf diesem Torus besitzt gleich viele Nachbarn; das Problem der unterschiedlichen Anpassung entfällt also. Allerdings hat auch die toroidale Lösung ihre Nachteile. Zum einen halbiert sich im Vergleich zu einer äquidimensionalen planaren Karte der maximal mögliche Abstand zwischen zwei

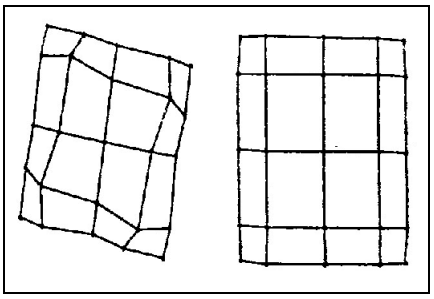

Abbildung 3.7: Zwei Beispiele für die Randeffekte bei herkömmlichen Kohonen-Karten (aus [1]). Die am Rand liegenden Knoten werden zu den in der Mitte liegenden Knoten hingezogen.

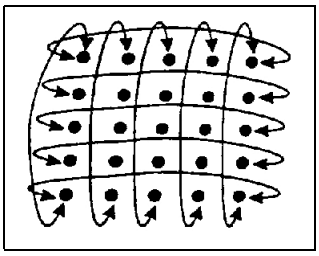

Abbildung 3.8: Schematische Darstellung eines Toroidalen Netzwerks (aus [1]). Durch Einfügen zusätzlicher Verbindungen zwischen den Knoten an den Rändern entsteht aus einer normalen selbstorganisierenden Karte eine mit toroidaler Topologie.

an entgegengesetzten Enden liegenden Merkmalsvektoren, da durch das Training ja eine stetige Änderung in der Ausrichtung der Gewichte erzeugt wird. Dadurch sinkt die „Auflösung“ mit der die Gewichtsvektoren der einzelnen Knoten den Merkmalsraum repräsentieren können. Möchte man nur eine grobe Strukturierung, kann dies ein Vorteil sein. Bei vielen kleinen Gruppen kann es allerdings sein, dass mehrere ununterscheidbar zu einem Cluster zusammengefasst werden, so dass man die Anzahl der Neuronen erhöhen muss, und damit auch die Trainingsphase entsprechend aufwändiger wird. Ein weiterer Nachteil ist leider gerade die Tatsache, dass der Torus keine Ränder mehr besitzt. Dies kann dazu führen, dass die Erhaltung der Nachbarschaft bei der Abbildung der Daten aus dem Merkmalsraum auf die Karte verloren geht. Anhand eines einfachen Experiments soll dieser Sachverhalt veranschaulicht werden:

Ein Quadrat Q der Seitenlänge 3 wird schachbrettförmig in neun gleiche

Teilquadrate der Seitenlänge 1 wie folgt unterteilt:

$$a = \{(x,y)|0 < x \leq 1, 2 < y \leq 3\} \quad b = \{(x,y)|1 < x \leq 2, 2 < y \leq 3\}$$
$$c = \{(x,y)|3 < x \leq 3, 2 < y \leq 3\} \quad d = \{(x,y\|0 < x \leq 1, 1 < y \leq 2\}$$
$$e = \{(x,y)|1 < x \leq 2, 1 < y \leq 2\} \quad f = \{(x,y)|3 < x \leq 3, 1 < y \leq 2\}$$
$$g = \{(x,y)|0 < x \leq 1, 0 < y \leq 1\} \quad h = \{(x,y)|1 < x \leq 2, 0 < y \leq 1\}$$
$$i = \{(x,y)|3 < x \leq 3, 0 < y \leq 1\}$$

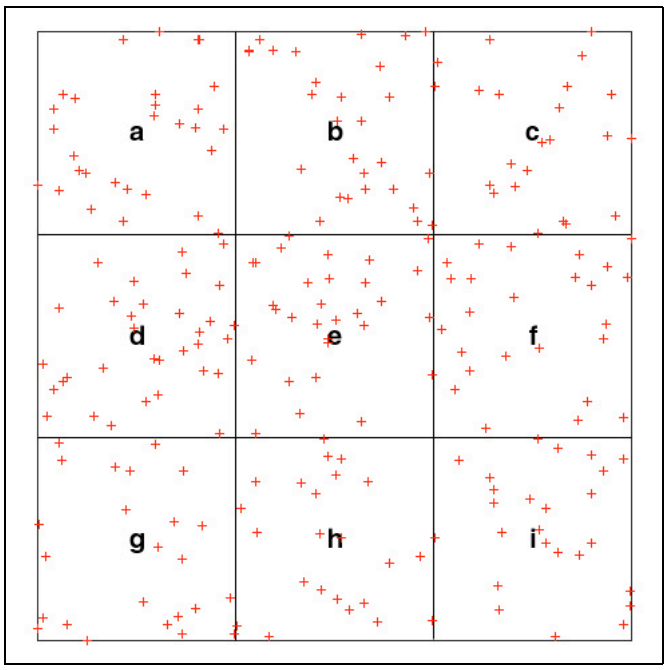

Abbildung 3.9: Die verwendeten Zufallsdaten liegen in einem Quadrat der Kantenlänge 3. Für das Training der Karten wurden die einzelnen Punkte jeweils mit dem Buchstaben des Teilquadrats, in dem sie liegen, bezeichnet.

Mit einem gleichverteilten Zufallszahlengenerator werden die Koordinaten x_P, y_P von 240 Punkten $P = (x_P, y_P), 0 \leq x_P, y_P \leq 3$ auf Q erzeugt. Sie dienen als Trainingsdatensatz für das Experiment und werden je nach ihrer Lage in einem der Teilquadrate $A, \ldots, I$ mit dem entsprechenden Buchstaben ausgezeichnet. Als Merkmale dienen die horizontale bzw. vertikale Koordinate x_P, y_P (siehe Abb. 3.9).

Anschließend wurden eine normale Kohonenkarte und eine toroidale Kohonenkarte mit diesen Daten trainiert. Beide Karten haben quadratische Form

und enthalten jeweils 81 Knoten. Für die Initialisierung wurden ebenfalls zufällig Punkte aus dem Trainingsset ausgewählt. Anschließend wurden die Karten mit 5.000 Zyklen und einem Trainingskoeffizienten $\alpha_0 = 0,2$ trainiert.

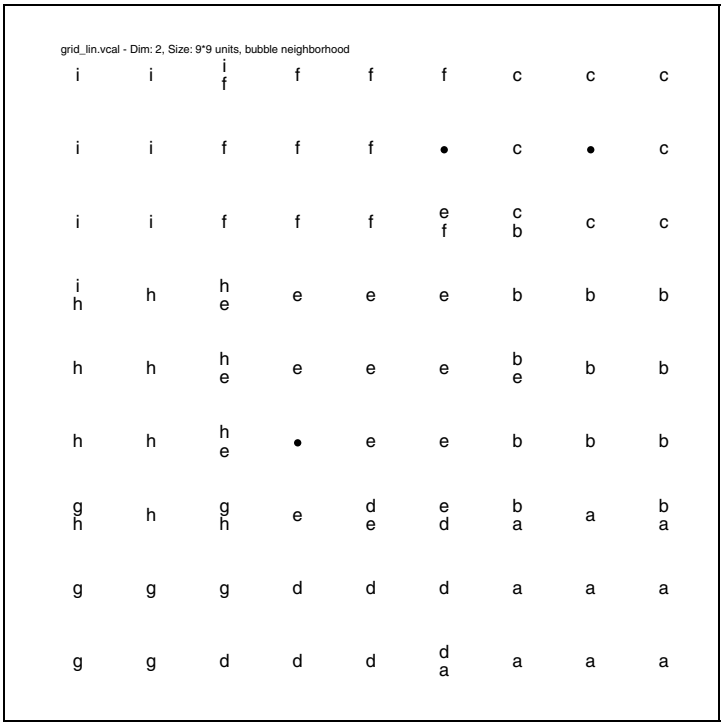

Abbildung 3.10: Auf der normale Kohonenkarte werden die neun Klassen klar getrennt. Nur an den Grenzen zwischen den Gruppen gibt es Knoten, die Repräsentationen für mehrere Klassen gelernt haben.

Während sich auf der normalen Kohonenkarte die Klassen sehr gut voneinander separieren (siehe Abb. 3.10), ist auf der toroidalen Karte die Klasse, die im Merkmalsraum an alle anderen angrenzt ('e') nicht mehr als zusammengehöriger Cluster erkennbar (siehe Abb. 3.11). Auswirkungen auf die Klassifizierung der anderen Muster sind ebenfalls zu beobachten. Während sich die Gruppen 'a' und 'b' beispielsweise auch auf der toroidalen Karte wiederfinden, wird 'f' ähnlich wie 'e' durch verschiedene nicht benachbarte Knoten repräsentiert. Der von 'd' eingenommene Bereich ist auf der toroidalen Karte deutlich kleiner als im herkömmlichen Fall; hier macht sich die bereits erwähnte Reduktion der Auflösung bemerkbar.

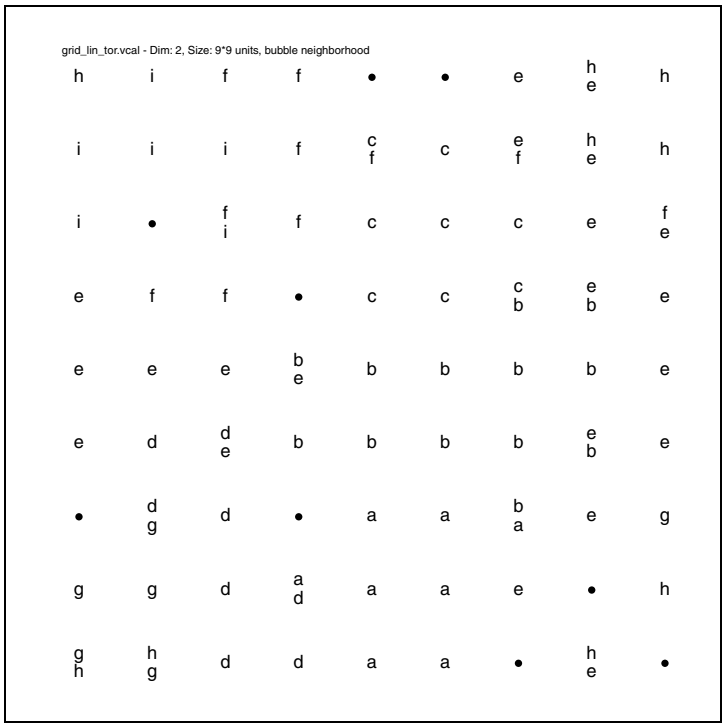

Abbildung 3.11: Auf der toroidalen Karte lässt sich die mittlere Klasse (E) nicht mehr sauber von den anderen acht Klassen trennen.

Da die Verteilung der Daten in hochdimensionalen Merkmalsräumen im allgemeinen nicht so gut absehbar ist wie in diesem Beispiel, wäre ein Bewertungsmaßstab wünschenswert, anhand dessen die Entscheidung für eine der beiden Architekturen getroffen werden kann. Nachdem das wichtigste Kriterium für die Qualität einer Kohonenkarte die korrekte Darstellung der nachbarschaftlichen Beziehungen zwischen Daten bzw. Clustern im Merkmalsraum ist, sollte dies von einem solchen Maßstab berücksichtigt werden. Ein naheliegender Ansatz liegt also im Vergleich der Lage von Muster-Daten im hochdimensionalen Merkmalsraum mit ihrer Position auf der Karte.

Sei n_i ein Knoten der Kohonenkarte M, $r \in \mathbb{N}$ und $N_r(n_i) = \{n_j | d(n_i, n_j) \leq r\}$ die Menge aller Knoten die von n_i höchstens den Abstand r haben. Außerdem bezeichne x einen Merkmalsvektor aus $\mathbb{R}^m$, n_c den Knoten, der dieses Muster auf der Karte repräsentiert und $N_s(x) = \{x_1, \ldots, x_s\}$ die Menge der s Muster mit den geringsten Abständen $\|x_j, x\| \forall 1 < j < s$ und $n_1, \ldots, n_s$ ihre – nicht notwendigerweise unterschiedlichen – Repräsentanten auf der Karte. c_i sei die Anzahl der Muster $x_l \in N_s(x)$, für die $n_l \in N_r(n_c)$ liegt. Dann kann

für jeden Knoten $n_i \in M$ ein Bewertungskoeffizient

$$C_{r,s}(n_i) = \frac{c_i}{s} \tag{3.10}$$

berechnet werden, der beschreibt, wie gut an dieser Stelle die Nachbarschaftsbeziehungen aus dem Merkmalsraum auf der Karte erhalten bleiben. Um eine Gesamtbewertung für die Karte zu erhalten, können die Bewertungskoeffizienten addiert und durch die Anzahl N der Knoten geteilt werden.

$$C_{\text{gesamt},r,s} = \frac{1}{N} \sum_i C_{r,s}(n_i) \tag{3.11}$$

Berechnet man diesen Wert für die beiden aufgeführten Karten mit dem Radius $r = 2$ und der Anzahl der Nachbarn $s = 8$, erhält man für die normale Kohonenkarte den Wert $C_{\text{gesamt},2,8} = 0,88$, für die toroidale Karte nur $C_{\text{gesamt},2,8} = 0,76$. Dies entspricht größenordnungsmäßig in etwa der schlechteren Anordnung von $\frac{1}{9}$ der Merkmalsvektoren.

Abgesehen von der toroidalen Form der SOM gibt es andere Erweiterungen für das Grundprinzip des Algorithmus. Bei vielen Anwendungen (Sprach- und Schrifterkennung, Zeitreihenanalyse) ist nicht nur die Erkennung eines einzelnen statischen Musters gefragt, sondern die Analyse einer Abfolge von Mustern bzw. des Kontexts, in dem ein bestimmtes Muster auftritt. Da das herkömmliche Verfahren lediglich einzelne statische Muster interpretieren kann, gibt es für diese Art der Anwendung eine Reihe von Ansätzen [32], von denen einige im folgenden vorgestellt werden sollen.

Die 'Temporal Kohonen Map' Eine Möglichkeit, Sequenzen von Mustern mit einer Kohonen-Karte zu lernen, wird von Chappell und Taylor [6] und in ähnlicher Form von Fancourt und Principe [11, 12] beschrieben. Der erste der beiden Ansätze, die 'Temporal Kohonen Map', soll hier näher erläutert werden. Die Erregung eines Knotens wird dabei als elektrisches Potenzial

$$V_i(t) = -\frac{1}{2} \sum_{j=1}^{m} (\xi_j(t) - \omega_{ij}(t))^2 \tag{3.12}$$

aufgefasst, wobei die Bezeichnungen für Gewichts- und Merkmalsvektor analog zu Abschnitt 3.2.1 gewählt sind. Dies entspricht der Bestimmung des euklidischen Abstands bei Kohonen.

Im Unterschied zur normalen Vorgehensweise, bei der nur die aktuelle Erregung der Knoten bestimmt wird, haben die Neuronen die Möglichkeit, sich

an vorangegangene Erregungszustände zu erinnern und diese in die aktuelle Bestimmung des Gewinner-Knotens miteinzubringen. Physikalisch entspricht diese Vorgehensweise der Erhaltung eines bestehenden elektrischen Potenzials in den Neuronen; ein vorhandenes Restpotenzial aus vorhergehenden Lernzyklen wird bei der Ermittlung des aktuellen Abstands zwischen Merkmals- und Gewichtsvektor berücksichtigt. Daher lautet die hier propagierte Formel für die Bestimmung der Gesamterregung bei der Präsentation eines neuen Musters:

$$V_i(t) = dV_i(t-1) - \frac{1}{2}\sum_j (\xi_j(t) - \omega_{ij}(t))^2. \tag{3.13}$$

Der physikalischen Tatsache, dass sich das Restpotenzial $V_i(t-1)$ im Verlauf der Zeit verflüchtigt, wird durch die Multiplikation mit dem 'decay'-Faktor $d, 0 < d < 1$ Rechnung getragen. Allgemein lautet die Formel für die Bestimmung des Potenzials eines Neurons im Lernzyklus t dann:

$$V_i(t) = -\frac{1}{2}\sum_{r=0}^{n-1} d^r \sum_j (\xi_j(t-r) - \omega_{ij}(t-r))^2 + d^n V_i(t-n). \tag{3.14}$$

Als Adaptionsfunktion wurde die Standard-Funktion von Kohonen eingesetzt. Bei Tests mit einfachen englischen Sätzen, bei denen dieselben Wörter zwar an der gleichen Position innerhalb eines Satzes, aber in unterschiedlichen Kontexten verwendet wurden, konnte von den erzeugten SOMs auch eine dementsprechend unterschiedliche Repräsentation gefunden werden. Die Methode ist also dazu geeignet, Sequenzen von Mustern zu lernen und Kontext einzelner Muster innerhalb einer Sequenz zu unterscheiden. Allerdings sind, durch die Minderung der vorangegangenen Potenziale durch den decay-Faktor, für die Repräsentation einer Sequenz auf der Karte hauptsächlich die zuletzt vorkommenden Muster ausschlaggebend. Beispielsweise würden zwei aus den Mustern x_1 und x_2 bestehende Sequenzen $x_2x_1x_1x_1x_1x_1$ und $x_1x_1x_1x_1x_1x_1$ im Normalfall durch den gleichen Knoten repräsentiert werden, so dass die Methode nur für sehr kurze Sequenzen verwendbar ist [31]. Außerdem sind die Informationen über vorangegangene Muster nur noch implizit enthalten.

Der SARDNET-Algorithmus Ein weiteres Verfahren zum Lernen von Sequenzen stellen James und Miikkulainen [31] vor. Diese als *SARDNET*[18] bezeichnete Methode unterscheidet sich vom üblichen Algorithmus durch die Einführung einer *Aktivierungskomponente* $a, 0 \leq a \leq 1$ für jeden Knoten

[18] **S**equential **A**ctivation **R**etention and **D**ecay **NET**work

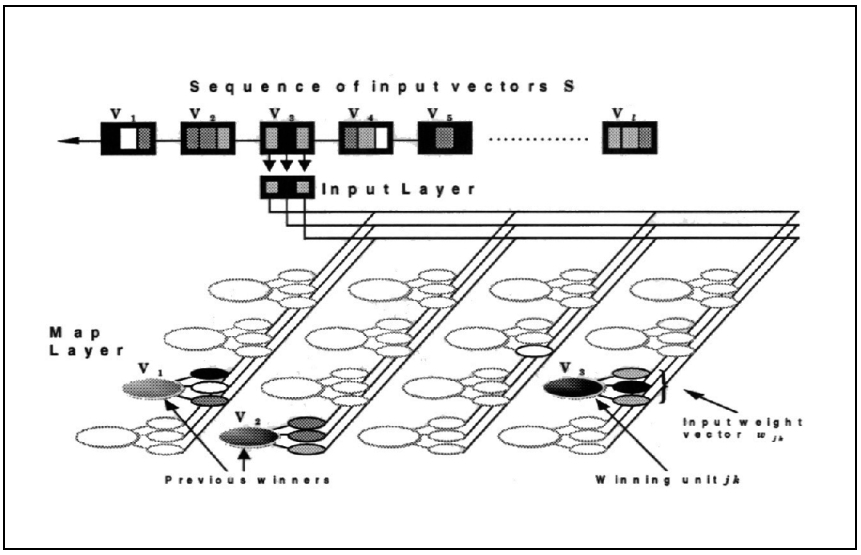

Abbildung 3.12: Schematische Darstellung der Architektur einer SARDNET Karte. Jeder Merkmalsvektor aus der Sequenz aktiviert einen Knoten der Karte. Bereits aktivierte Neuronen werden bei der Ermittlung des Gewinner-Knotens ausgelassen. Die Aktivierung wird dabei schrittweise abgebaut, um so die Position des Knotens in der Sequenz festzuhalten. [31]

der Karte. Nach der Ermittlung des Gewinner-Knotens für ein Muster wird dessen Aktivierung auf $1,0$ gesetzt und nimmt dann im weiteren Verlauf des Trainings, ähnlich wie das Potenzial bei der Temporal Kohonen Map, kontinuierlich ab. Ein weiterer Unterschied zu Kohonens Original besteht darin, dass ein einmal als Gewinner ermittelter Knoten nicht ein zweites Mal gewählt werden kann. Stattdessen wird das Neuron aktiviert, dessen Gewichtsvektor den nächstkleinsten Abstand vom Merkmalsvektor aufweist.

Das Training der Karte erfolgt analog zum Algorithmus von Kohonen, allerdings wird statt eines einzelnen Musters immer die ganze Sequenz präsentiert. Der Teil für das Lernen einer Muster-Sequenz besteht aus den folgenden Schritten

1. Die Aktivierung a jedes Knotens wird auf 0 gesetzt.

2. Finde einen Knoten n_c mit $a_i = 0$ und minimalem Abstand $\sum_{j=1}^{m}(\xi_j(t) - \omega_{ij}(t))^2$.

3. Setze $a_c = 1$.

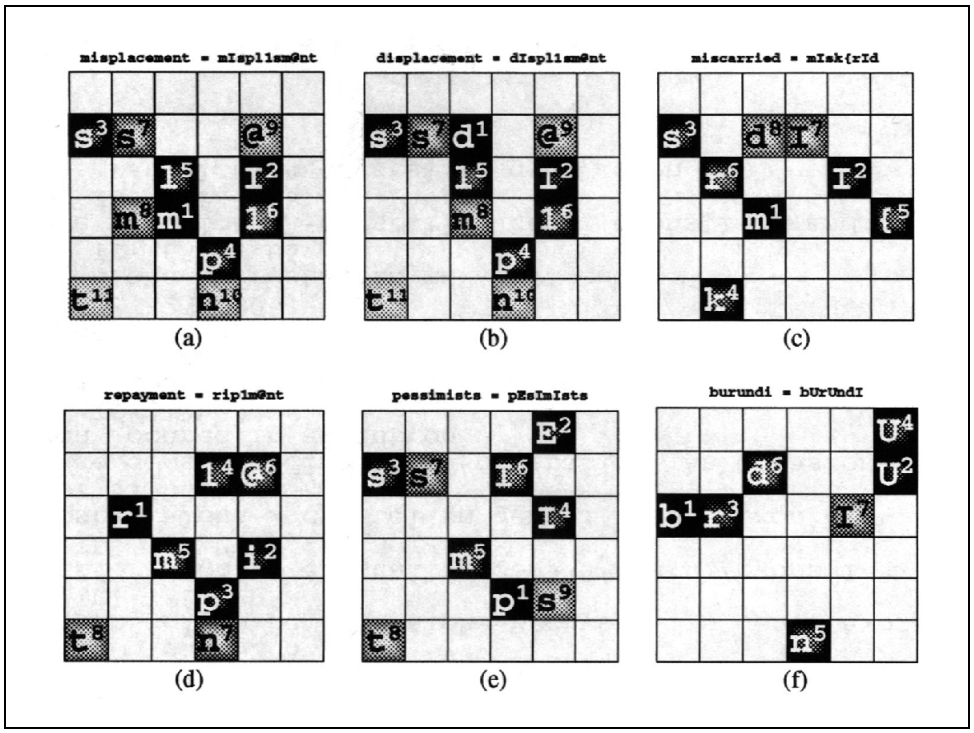

Abbildung 3.13: Beispiele für die Visualisierung einzelner Wörter auf einer SARDNET Karte. Je dunkler ein Knoten gefärbt ist, desto früher wurde er aktiviert. Die Gewichte der Knoten entsprechen dabei den Phonemen, nicht den Buchstaben; für die Repräsentation auf der Karte ist also die Aussprache wichtig, nicht die Schreibweise. Die beiden ersten Wörter, 'misplacement' und 'displacement', unterscheiden sich nur in ihrem Anfangslaut und sind daher sehr ähnlich. Entsprechend unterscheiden sich auch ihre Visualisierungen auf der Karte nur minimal. Das Neuron, welches durch das erste 'm' von 'misplacement' aktiviert wurde, wird auch bei 'displacement' aktiviert, allerdings ist die verbleibende Aktivierung am Ende der Sequenz deutlich höher. In der unteren Reihe sind Wörter dargestellt, die mit denen aus der oberen Reihe nur Teile gemeinsam haben. Wie man sieht, werden dabei dieselben (oder unmittelbar benachbarte) Neuronen aktiviert. Das Wort 'burundi' hat am wenigsten mit allen anderen Wörtern gemeinsam. Dementsprechend unterschiedlich ist seine Repräsentation auf der Karte. [31]

4. Adaptiere die Gewichtsvektoren w_k aus der Nachbarschaft $N_c(t)$ analog zu Kohonens Algorithmus.

5. Reduziere die Aktivierung aller anderen Knoten mit $a > 0$ um dieselbe Größe d.

6. Prüfe, ob das Ende der Sequenz erreicht wurde, ansonsten fahre mit Schritt 2 fort.

Die aktivierten Knoten stellen dann, nach ihren Aktivierungsknoten geordnet

beginnend mit dem kleinsten a, eine Repräsentation der Sequenz auf der Karte dar. Durch den Ausschluss von bereits aktivierten Knoten sind hier auch Wiederholungen desselben Musters wiedererkennbar. Von einer herkömmlichen SOM unterscheidet sich die entstandene Karte dadurch, dass Muster, die relativ häufig in einer Folge vorkommen, durch mehr als einen Knoten auf der Karte repräsentiert werden.

Um das Verfahren zu testen, wurden Wörter in ihre Phoneme zerlegt und anschließend auf der Karte dargestellt. Wie sich zeigt, werden ähnlich klingende Wörter durch vergleichbare Folgen aktivierter Neuronen visualisiert. Auch einzelne Wortteile lassen sich vergleichen. Dabei stimmen zwar die Werte der Aktivierungen nicht überein, jedoch werden dieselben oder benachbarte Neuronen aktiviert (siehe Abb. 3.13).

Mit dem SARDNET-Algorithmus gibt es also ein unüberwachtes Verfahren, welches die Möglichkeit bietet, Folgen von Mustern zu lernen und ähnliche Folgen durch ähnliche Repräsentationen darzustellen. Dabei sind Wiederholungen möglich und die Informationen über die gesamte Mustersequenz bleiben in der Visualisierung erhalten.

3.3 Vorhandene Verfahren zur Promotoranalyse

In diesem Abschnitt sollen einige bereits vorhandene Verfahren zur Promotoranalyse vorgestellt werden, die auf künstlichen neuronalen Netzen basieren. Bei allen dreien handelt es sich um Verfahren, denen ein Backpropagation-Algorithmus zu Grunde liegt.

Promotorerkennung im Genom von Escherichia coli Das Bakterium Escherichia coli[19] ist ein Symbiont aus dem Darm von Säugetieren, der bereits in den 50er Jahren häufig Untersuchungsobjekt der Molekularbiologie war. Sein Genom besteht aus einer ringförmigen DNA, die ca. 4,72 Millionen Basenpaare enthält [37]. Michael O'Neill hat Anfang der 90er für drei Gruppen von Promotoren dieses Bakteriums Verfahren entwickelt, um vorher unbekannte Promotoren innerhalb des Genoms zu finden [50, 51]. Dabei wurden Backpropagation-Netzwerke mit drei Schichten verwendet. Für die Eingabe wurden die vier unterschiedlichen Basen in binärer Form codiert

[19] im folgenden meist kürzer: E. coli

(A = 0001, C = 0010, G = 0100, T = 1000). Als Eingabe-Sequenz dienten für die beiden ersten Promotor-Klassen jeweils Sequenz-Stücke von 58 Basen (bzw. 232 Binärwerten). Für die dritte Klasse „konnte in mehr als 30 Versuchen kein Netzwerk mit einer aus 58 Basen bestehenden Eingabe trainiert werden“ [51], daher wurde „die Eingabe auf die Stellen mit dem höchsten Informationsgehalt reduziert“ [51], dabei handelte es sich um 20 Basen (bzw. 80 Binärwerte). Analog bestand die Eingabe-Schicht der Netze aus 232 bzw. 80 Neuronen. Für jede Klasse wurden mehrere Netze mit einer unterschiedlichen Anzahl von Neuronen in der verborgenen Schicht trainiert. Für das Training wurde Promotoren ein Ausgabe-Wert von $1,0$, Nicht-Promotoren ein Wert von $0,0$ zugeordnet. Beim Testen von unbekannten Sequenzen wurde ein Muster dann als Promotor eingestuft, wenn der ausgegebene Wert des trainierten Netzes über $0,90$ lag. Als Trainings- und Test-Menge dienten bekannte Promotor-Sequenzen. Da es sich dabei jeweils nur um wenige Exemplare handelte (zwischen 18 und 41), wurden diese künstlich erweitert, indem an allen „für die Promotor-Funktion unkritischen Stellen“ [51] jeweils eine Mutation eingeführt wurde.

Für die erste Klasse von Promotoren, für die neun Trainingssequenzen vorlagen, wurden beispielsweise 28 Stellen innerhalb der 58-Basen Sequenz als „unkritisch“eingestuft. An diesen wurden jeweils die drei Basen eingesetzt, die nicht in der ursprünglichen Sequenz vorkamen. So erhält man aus den 9 ursprünglichen Trainings-Sequenzen insgesamt 1.008 ($9\times28\times4$). Als Negativ-Trainingsmenge wurden 4.000 per Zufallsgenerator erzeugte Sequenzen (mit jeweils 50% Anteil an A und T) verwendet, die zunächst auf bekannte Promotoren untersucht wurden, um die etwaige Verwendung zufällig erzeugter positiver Muster zu vermeiden. Die 1.008 Positiv-Exemplare wurden verdoppelt, so dass in der Menge der Trainingsdaten das Verhältnis von Promotoren zu Nicht-Promotoren bei etwa $1/2$ lag. Mit diesen Daten wurden dann Netze mit unterschiedlich vielen Neuronen in der verborgenen Schicht bzw. unterschiedlich vielen Lernschritten trainiert. Zur Verifizierung der Ergebnisse diente eine Menge von neun weiteren, nicht in der Trainingsmenge enthaltenen Promotoren. Aus diesen Testfällen wurden von allen vier Netzen jeweils die gleichen sieben Sequenzen ($\sim 78\%$) korrekt als Promotoren klassifiziert. Von den Nicht-Promotoren wurden in keinem Fall mehr als $0,5\%$ fälschlicherweise als Promotoren erkannt.

In analoger Weise wurden mit vergleichbaren Ergebnissen für die beiden anderen Promotor-Klassen jeweils mehrere Netze trainiert. Der Versuch, ein einzelnes Netz für alle drei Klassen zu finden, schlug fehl. Die Erkennungs-

rate lag bei allen entsprechend trainierten Netzen unter 60%.

Time-Delay Verfahren zur Promotorerkennung Auch das in Abschnitt 3.1.2 beschriebene 'Time Delay Neural Network' von Waibel ist für die Erkennung von Promotoren eingesetzt worden. In zwei unterschiedlichen Ansätzen [44, 61] wurde das Verfahren verwendet, um distinkte Merkmale von Promotoren zu erkennen. Dabei handelte es sich in beiden Fällen um das Auffinden der TATA-Box und der Initiator-Sequenz.

Analog zu ONeills Verfahren wurden auch hier die vier Basen binär kodiert.

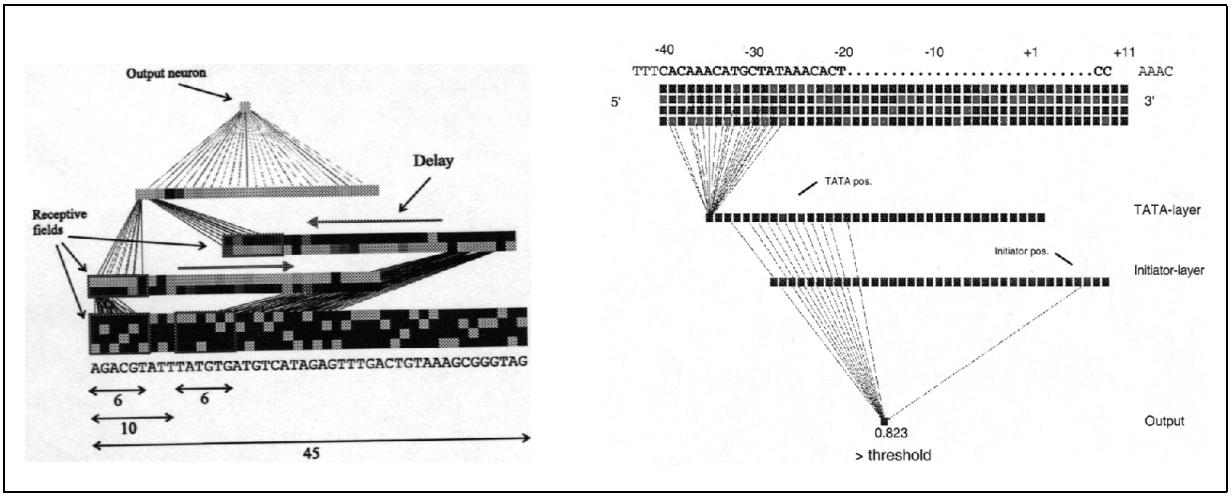

Abbildung 3.14: TDNNs zum Lernen von Promotormerkmalen (links aus [44] und rechts aus [61]).

Beide Ansätze benutzen zusätzlich zur Eingabeschicht jeweils eine Schicht für die Erkennung von TATA-Box und Initiator und eine weitere Schicht, welche die Ergebnisse dieser beiden kombiniert und schließlich mit einem Ausgabe-Neuron verknüpft ist. Unterschiede gibt es in der Anzahl der Neuronen in den einzelnen Schichten sowie in der Vorgehensweise beim Training.

Bei Mache et al. [44] wird ein Fenster von 45 Basen betrachtet, dementsprechend bestand die Eingabeschicht aus 180 Neuronen (4×45). Die beiden Schichten für die Erkennung der Motive besteht aus 29 Neuronen. Aus der Eingabeschicht werden je 6 aufeinanderfolgende Signale an die Zwischenschichten weitergegeben, das zweite Fenster mit einem Offset von 10 Basen zum Beginn des ersten, so dass beide Fenster für unterschiedliche Motive kodieren. Aus den Zwischenschichten wurden wieder 6 aufeinanderfolgen-

de Signale an die nächste Schicht von 25 Neuronen weitergegeben. Um die Erkennung weiter auseinanderliegender Signale zu vereinfachen, wurde die Eingabeschicht jeweils in umgekehrter Reihenfolge mit den beiden Zwischenschichten verknüpft. Dieses Netz wurde mit Sequenzen aus 640 zufällig aus der EPD ('Eukaryotic Promoter Database') [52, 53, 54] ausgewählten Promotoren trainiert. 35 andere Promotorsequenzen (ebenfalls aus der EPD) dienten als unmittelbare Test-Menge für die Qualität des Netzes. Anschließend wurde die oberste Time Delay Schicht durch ein 'Dynamic Time Warping' Verfahren [67] ersetzt. Dieses aus der Sprachverarbeitung stammende Verfahren dient der Erkennung von ähnlichen Mustern unterschiedlicher Ausdehnung und wird hier für den Vergleich der Aktivierungsmuster der Neuronen in der Zwischenschicht verwendet. Die beiden Zwischenschichten lernten dabei Repräsentationen für die beiden häufigsten Motive: Die TATA-Box und den Initiator. Nach Abschluss des Trainings wurden mit dem Netz 74 Gen-Sequenzen mit bekannten Promotoren durchsucht. Von diesen wurden 58% gefunden. Die Rate der fälschlicherweise als Promotoren erkannten Abschnitte lag bei etwa $0,066\%$, d. h. pro 1.512 Basen wurde ein falscher Promotor gefunden.

Einen etwas anderen Weg geht Reese [61]. Er betrachtet ein Fenster von 51 Basen (204 Neuronen in der Eingangsschicht). Seine beiden Zwischenschichten aus jeweils 37 Knoten sind direkt mit dem Ausgabe-Neuron verbunden. Die beiden Zwischenschichten für TATA-Box und Initiator wurden zunächst getrennt trainiert. Dazu wurden die Teilnetze mit entsprechenden Motiven trainiert, bis ein lokales Minimum erreicht war. Dann wurden die Gewichtskanten gelöscht, die am wenigsten zu einer richtigen Vorhersage beitrugen. Dieser Schritt wurde mehrmals wiederholt, bis eine vorgegebene Fehlertoleranz erreicht war. Für das Training wurden 429 Promotoren mit einer Länge von 400 Basen aus der EPD ausgewählt, deren paarweise Ähnlichkeit 25% nicht überschreiten durfte. Als Negativ-Muster für das Training dienten 305 kodierende Sequenzen aus der Genbank [4]. Als Trainings-Menge dienten 300 Promotoren und 3.000 aus den Gen-Sequenzen ausgeschnittene Stücke kodierender DNA. Als Testset wurden die übrigen 129 Promotoren und 1.000 Abschnitte aus den Gen-Sequenzen verwendet. Nach Abschluß des Trainings wurden mit dem Netz 284 der 305 vollständigen Gen-Sequenzen auf Promotoren durchsucht. Dabei ergab sich bei einer Trefferquote von ebenfalls 58% für die bekannten Promotoren eine Rate von $0,14\%$ für falsch positive Vorhersagen ($\sim$ ein falscher Treffer pro 717 Basen).

3.4 Zusammenfassung

In diesem Kapitel wurden einige Verfahren zur Mustererkennung aus dem Bereich der künstlichen neuronalen Netze vorgestellt und ihre theoretischen Grundlagen erörtert. Bei der Darstellung der überwachten Verfahren wurde insbesondere auf die Methoden eingegangen, die schon für Promotorerkennung eingesetzt worden sind. Das in den nächsten Kapiteln beschriebene neu entwickelte Verfahren basiert allerdings auf unüberwachten Methoden; dementsprechend wurden auch die hierfür relevanten Algorithmen erläutert.

Zusätzlich wurde die Vorgehensweise von drei bereits existierenden Verfahren zur Promotorerkennung dargestellt. Die beschriebenen Methoden zur Erkennung von Promotoren sind jedoch nur eingeschränkt für die Suche in großen genomischen DNA-Sequenzen einsetzbar. Auf die Gründe hierfür wird im nächsten Kapitel näher eingegangen werden. Außerdem wird ein neues Verfahren präsentiert, das einen grundlegend anderen Ansatz für die Klassifizierung und Erkennung von Promotoren bieten soll, der stärker an der Biologie orientiert ist.

Kapitel 4

Promotorklassifizierung mit SOMs

Die wichtigste Anforderung, die Software zur Vorhersage biologischer Strukturen zu erfüllen hat, ist die Bereitstellung von Intformationen für die Vereinfachung bzw. Reduzierung experimentaler Arbeitsschritte im Labor. Bezüglich dieses Gesichtspunktes sind die Ergebnisse der im letzten Abschnitt vorgestellten bisherigen Methoden nur teilweise zufriedenstellend. Obwohl bei eukaryotischen Promotoren Erkennungsraten um die 60% [44, 61], bei den prokaryotischen sogar fast 80% [51] erreicht werden, relativieren sich diese Ergebnisse angesichts der hohen Anzahl falsch positiver Vorhersagen. Beschränkt sich eine Analyse darauf, in einem kurzen Sequenzstück, von dem bereits bekannt ist, dass es einen Promotor enthält, nach dessen genauer Lokalisation zu suchen, kann man von beiden Verfahren korrekte Ergebnisse erwarten; problematisch ist jedoch die Suche in längeren Sequenzen etwa bei Genom-Analysen. Legt man die veröffentlichen Raten von einem falschen Treffer auf 1.512 Basen [44] bzw. 717 Basen [61] zugrunde, überträfe bei einer Analyse des menschlichen Genoms rein statistisch betrachtet die Anzahl der falsch positiven die der tatsächlichen Treffer selbst dann um einen Faktor 50-100, wenn man die korrekte Erkennung aller echten Promotoren annimmt. Die wichtigste Frage für die Entwicklung eines neuen Verfahrens bzw. die Verbesserung der bestehenden Methoden muss daher sein, worin die Ursache für die hohe Anzahl der falsch positiven Treffer zu suchen ist, und wie man diese reduzieren kann.

Nach der Untersuchung dieser Fragestellung im ersten Teil dieses Kapitels wird anschließend ein neuer Ansatz zur in *silico* Promotoranalyse vorgestellt, der sich durch seine verstärkte Berücksichtigung der biologischen Eigenschaf-

ten von Promotoren von den bisher präsentierten Methoden grundlegend unterscheidet. Dabei steht nicht mehr so sehr das Finden von Promotoren im Vordergrund sondern die tatsächliche, biologisch begründete Klassifikation von Promotoren in funktionell relevante Gruppen.

4.1 Vorüberlegungen zur *in silico* Promotorklassifizierung

Betrachtet man die in Abschnitt 2.1.2 erläuterten komplexen Promotorstrukturen im Zusammenhang mit den in Abschnitt 3.3 vorgestellten Methoden, wird deutlich, dass bei der Entwicklung der Verfahren nur ein Bruchteil des biologischen Wissens über Promotoren genutzt wurde. O'Neill [50, 51] verwendet als Trainings-Muster jeweils gleichlange Fenster aus dem Promotorbereich. Variabilitäten innerhalb des Promotors müssen hier vom Netz gelernt werden. Diese Einschränkung ist insofern nachvollziehbar, als die Promotoren der Prokaryoten im Vergleich zu denen der Eukaryoten deutlich einfacher aufgebaut sind [37, 43]. Obwohl das Verfahren direkt einzelne lange Sequenzstücke verwendet und eventuell enthaltene Strukturen wie etwa die Pribnow-Box nur implizit für die Mustererkennung verwendet werden, hat es die höchste Erkennungsraten der vorgestellten Verfahren; mit $0,5\%$ falsch positiver Vorhersagen pro Nukleotid jedoch auch eine hohe Rate an falschen Treffern.

Ein zusätzlicher Kritikpunkt ist die Verwendung zufällig erzeugter Sequenzen für das Training. Obwohl der Anteil der vier Nukleotide `A`, `C`, `G` und `T` in DNA-Sequenzen insgesamt jeweils etwa 25% beträgt, gibt es Bereiche, in denen eine oder mehrere Basen deutlich häufiger vorkommen als andere. Diese Überrepräsentationen stellen im allgemeinen keine zufällige Laune der Natur dar, sondern sind durch biologische Sachverhalte begründet. Die sogenannten '*CpG islands*'[1] findet man beispielsweise im menschlichen Genom oft vor den kodierenden Bereichen der 'housekeeping genes' [41], was als Hinweis auf Promotorbereiche dienen kann. Innerhalb von 'UTRs'[2] am Ende des kodierenden Bereichs einzelner Gene findet man hingegen oft sogenannte *poly-A* Regionen, die einen deutlich erhöhten Anteil von Adenin enthalten [55]. Im allgemeinen ist es daher nicht ratsam, Analysen biologischer Sequenzen auf

[1] Während das Dinukleotid `CG` im Genom insgesamt nur unterdurchschnittlich oft vorkommt, ist der Anteil innerhalb dieser Bereiche deutlich höher [5].

[2] 'Untranslated regions'. Der diesem DNA-Bereich zugeordnete Abschnitt der mRNA wird nicht translatiert, daher die Bezeichnung.

den rein statistischen Eigenschaften von Sequenzen aufzubauen.

Die Verfahren von Mache [44] und Reese [61] tragen dieser Tatsache bereits Rechnung und verwenden für das Negativtraining keine zufällig erzeugten Sequenzen, sondern Ausschnitte aus kodierenden Bereichen des Genoms. Auf diese Weise wird sichergestellt, dass es sich nicht um Promotoren handelt, aber dennoch in der Natur vorkommende Sequenzen mit den darin enthaltenen Sequenzstrukturen benutzt werden. Dementsprechend versuchen beide Methoden auch nicht, ein einzelnes kontinuierliches Sequenzstück zu klassifizieren, sondern berücksichtigen den modularen Promotoraufbau immerhin insofern, als sie die zwei häufigsten Elemente (TATA-Box und Initiator) sowie deren variablen Abstand für die Klassifikation verwerten; d. h. sie suchen eigentlich nach dem 'core promoter' (siehe Abschnitt 2.1.2). Dabei ist die hohe Anzahl an falsch positiven Treffern im wesentlichen dem schlecht konservierten Transkriptionsstart zuzuschreiben. Bei Reese, der für jedes der beiden Elemente zunächst ein eigenes Netz trainiert, übertrifft die Anzahl der fälschlich erkannten INRs die der TATA-Boxen je nach der Gesamterkennungsrate des Netzes um das 15-20-fache. Dennoch ist die Fehlerrate bei der Kombination beider Elemente deutlich geringer als für die TATA-Box alleine [61].

Muster	korrekte Treffer	Trefferquote in %	falsch positive Treffer	Wahrscheinlichkeit in % pro Basenpaar
TATA	412	68,1	2.140	0,35
INI	501	82,8	21.816	3,63
$\mathcal{M}_{CP}$	301	49,8	487	0,08

Tabelle 4.1: Werte für die Suche nach TATA-Box, Initiator und dem 'core promoter' Modell. Die Trefferrate des Modells ist mit 49,8% etwa 8% schlechter als die der TDNN-Verfahren, dafür ist die Rate der falsch Positiven mit 0,08% ebenfalls deutlich niedriger als der Wert von 0,14% beim Verfahren von Reese.

Um festzustellen, ob die hohen Fehlerraten tatsächlich auf die niedrige Konservierung des Transskriptionsstarts-Motivs zurückzuführen sind, ist ein Vergleich mit den Ergebnissen anderer Methoden angebracht. Die in Abschnitt 2.2.2 beschriebenen Programme MatInspector, FastM und ModelInspector lassen sich ebenfalls dazu verwenden, um nach TATA-Box und Initiator, bzw. einem 'core promoter'-Modell, das beide Elemente beinhaltet, zu suchen. Dazu wurden 605 Vertebraten-Promotoren aus der EPD [54] untersucht.[3] Als

[3]EPD release 65, dabei wurde die Standard Filter-Funktion der EPD benutzt, um von

Einzelmuster dienten eine TATA-Box Matrix sowie eine Initiator-Matrix aus [37]. Als Modell wurde $\mathcal{M}_{CP} = \{\mathsf{TATA,INI}\}, (22), (38)\}$[4] mit FastM realisiert und anschließend mit dem ModelInspector gesucht. Dabei wurden die Variablen für die Programme so gewählt, dass die Erkennungsrate mit 301 richtigen Treffern in den Testsätzen und damit rund 50% Erkennungsquote in etwa denen der anderen Methoden entspricht. Auch hier zeigt sich, dass bei den Einzelmustern die Qualität der TATA-Matrizen die der Initiator-Matrix dominiert, wohingegen sich durch Kombination beider Muster nochmals eine Verbesserung der Ergebnisse erzielen lässt.

Insgesamt ist daher auch bei dieser Methode die Auftretenswahrscheinlichkeit für eine falsche Vorhersage mit etwa einem Treffer pro 1.233 Basenpaaren zu hoch, um für den Laboreinsatz verwertbare Ergebnisse zu erzielen.

Offensichtlich sind die hohen Fehlerraten bei den Verfahren von Mache und Reese also nicht in den verwendeten Verfahren an sich, sondern in der Verwendung der schlecht konservierten Initiator-Region begründet. Aus der Sicht der Biologie entspricht dies der Tatsache, dass Promotoren in der Natur ebenfalls nicht nur über die Identifizierung des 'core promoters' aktiviert werden, sondern erst durch die Bildung des Transkriptionskomplexes mit Hilfe zusätzlicher Bindeproteine die Transkription gestartet werden kann. Obwohl auch die Bindungsstellen-Motive variabel sind, müssen sie gewisse Einschränkungen erfüllen, um die spezifische Anlagerung eines Proteins zu gewährleisten. Der Initiator ist hingegen eher als zusätzliches Positionierungssignal zu verstehen, das deutlich weniger ausgeprägt ist. Für eine ganze Reihe von Transkriptionsfaktorbindungsstellen sind Matrizen vorhanden [27], deren Qualität die des Initiator-Motivs weit übertrifft. Dementsprechend werden auch Module aus zwei solchen Faktoren im Normalfall deutlich seltener gefunden als das hier untersuchte $\mathcal{M}_{CP}$. Von Klingenhoff et al. [36] wurden 50 solcher Module untersucht. Die Auftrittshäufigkeit lag dabei bei unter einem Treffer pro 10,000 Nukleotide. Noch besser schneiden Modelle aus mehr als zwei Transkriptionsfaktorbindungsstellen ab. Für die bei Frech et al. [20] beschriebenen Aktin-Promotoren liegt die Erkennungsrate bei 69,7% mit nur einem falschen Treffer in 1.290 Negativbeispielen.

Theoretisch ließen sich aus den etwa bekannten Transkriptionsfaktoren ei-

Promotoren mit hoher Homologie lediglich jeweils ein Exemplar zu erhalten; die Position des Transkriptionsstarts ist bei diesen Promotoren bekannt.

[4] ebenfalls mit den beiden Matrizen aus [37]

ne eineindeutige Kombination und damit ein eineindeutiger Promotor für jedes Gen konstruieren, so dass jedes einzelne separat reguliert werden könnte. Wie in Abschnitt 2.1.2 bereits dargelegt wurde, trifft dies in der Natur jedoch nicht zu [43, 79]. Stattdessen ähneln sich die Strukturen von Promotoren funktionell zusammenhängender Gene und ermöglichen so beispielweise die gleichzeitige Regulation mehrer Gene durch eine bestimmte Gruppe von Transkriptionsfaktoren [79]. Im Prinzip ist der Ansatz, nach einer Kombination mehrerer Bindungsstellen zu suchen, also erfolgversprechend. Allerdings tritt keines dieser Motive in allen Promotoren auf, dementsprechend benötigt man verschiedene Modelle, um möglichst viele Promotoren zu erfassen. Erschwert wird die Suche nach solchen Modellen durch die Tatsache, dass sich mit aktueller Software [19] innerhalb der Promotor-Sequenz deutlich mehr Bindemotive finden lassen, als die tatsächlich für die Bildung des Transkriptionskomplexes benötigten. Funktionell ähnliche Promotoren enthalten also eine Vielzahl möglicher Bindungsstellen, aus denen das spezifische Modell herausgefiltert werden muss. Hier stoßen die Verfahren von Reese und Mache schnell an ihre Grenzen. Die Netz-Architektur müsste so erweitert werden, dass sie alle möglichen Bindungsstellenmotive ebenso berücksichtigt wie die unterschiedlichen Reihenfolgen und Abstände der einzelnen Muster. Für jedes Promotor-Modell würde also eine eigene Netzarchitektur benötigt, d. h. der Aufbau der Modelle oder zumindest die Anzahl der beteiligten Faktoren müsste von vornherein bekannt sein, um in die Konstruktion der Netze miteinzufließen.

Für die Konstruktion und Suche solcher Modelle wurden die in Abschnitt 2.2.2 beschriebenen Programme ModelGenerator und ModelInspector [36, 20] entwickelt. Sie suchen mit Hilfe von vordefinierten Matrizen oder IUPACs direkt nach Bindungsstellen. Dies ersetzt die unterste Schicht der TDNNs von Mache und Reese. Anstelle direkt innerhalb der Sequenz nach ähnlichen Bereichen zu suchen, wird so nur noch nach Mustern gesucht, deren Informationsgehalt bereits gesichert ist. Die weitere Beschränkung durch die feste Reihenfolge und die vorgegebenen Abstände zwischen den Motiven führt dann zur hohen Spezifität dieser Modelle und damit zu Ergebnissen, die für die Laboranalyse geeignet sind [18].

Um brauchbare Ergebnisse liefern zu können, benötigt der ModelGenerator jedoch einen Satz funktionell verwandter Promotoren, sowie das Grundgerüst eines Modells, dass dann automatisch überprüft und erweitert werden kann. Diese Vorarbeit kann bisher nicht automatisiert werden. Funktionelle Gemeinsamkeiten einzelner Gene müssen aus der Literatur erarbeitet werden,

anschließend können die dazugehörigen Promotoren auf gemeinsame Bindungsstellen untersucht werden. Diese Vorgehensweise ist sehr zeitaufwändig und aufgrund mangelnder Kenntnisse über die Funktionalität vieler Gene nicht in großem Maßstab durchführbar.

Wünschenswert wäre also eine Methode, die ähnliche Promotoren automatisch in funktionell verwandte Gruppen einteilt und bereits Grundinformationen über die Struktur dieser Promotoren liefern kann. Diese könnten dann mit Hilfe existierender Methoden verfeinert werden, um so hochspezifische Modelle für einzelne Promotorklassen zu erhalten. Eine entsprechende Architektur wurde im Rahmen dieser Arbeit entworfen; sie wird im folgenden Abschnitt vorgestellt.

4.2 Architektur des Verfahrens

Das Ziel der neuzuentwickelnden Methode war es, dem in Abschnitt 2.1.2 beschriebenen modularen Aufbau von Promotoren Rechnung zu tragen. Daher wurden sowohl die unterschiedlichen Typen der regulatorischen Bindungsstellen als auch deren Anordnung sowie die Abstände innerhalb der Promotoren berücksichtigt.

Für die Erkennung der beiden unterschiedlichen Elemente des 'core promoters' dienten bei den TDNNs von Reese und Mache [61, 44] die Zwischenschichten, die jeweils eine Repräsentation für TATA-Box bzw. Initiator lernten. Um weitere Bindungsstellen erkennen zu können, müssten bei beiden Verfahren also entsprechende Teilnetze trainiert werden. Allerdings sind für die Identifizierung Bindungsstellen in DNA-Sequenzen bereits verschiedene Programme verfügbar [29, 58]. Anstatt eine eigene Methodik zu entwickeln, wurde hier der MatInspector und die dazugehörigen Matrizen (siehe Abschnitt 2.2.2) für die Erkennung von Bindungsstellen eingesetzt werden. Das Programm liefert als Ergebnis eine Liste aller potentiellen Transkriptionsfaktorbindungsstellen sowie deren Position in einer vorgegebenen Nukleotid-Sequenz. Promotor-Analysen können dann direkt auf diesen Resultaten, und damit bereits auf einer, im Vergleich zur Sequenz, biologisch funktionalen Ebene aufgebaut werden.

Nachdem die Art der einzelnen Bindungsstellen erkannt ist, sind ihre Anordnung und die Abstände zu berücksichtigen, um funktionelle Promotor-Klassen zu finden. Wie im vorhergehenden Abschnitt bereits dargelegt wur-

de, stellen Ansätze, bei denen die Anzahl der Promotor-Elemente von vornherein als feste Größe in die Konstruktion des Verfahrens mit eingeht, nur eine eingeschränkte Lösung dar. Darüberhinaus läßt sich über die zu erwartende Anzahl der unterschiedlichen Promotorklassen keine exakte Aussage treffen. Aus diesem Grund erscheint die Verwendung eines überwachten Verfahrens wenig sinnvoll, da ohne vorgegebene Klasseneinteilung kein Training möglich ist. Besser geeignet erscheinen Methoden der Selbstorganisation, um die Vielzahl der unterschiedlichen Promotoren in Gruppen ähnlicher Funktion aufzuteilen.

Für das Lernen von Muster-Folgen wurden in Abschnitt 3.2.2 mehrere unüberwachte Verfahren, wie z. B. die 'Temporal Kohonen-Map' von Chappell [6] vorgestellt. Das Ziel dieses Verfahrens war die kontext-abhängige Erkennung einzelner Wörter in gesprochenen Sätzen. Auch die für die 'core promoter'-Suche [61, 44] verwendete Netzarchitektur stammt aus der Sprachverarbeitung [77]. Offensichtlich gibt es bei den Fragestellungen aus diesem Bereich der Mustererkennung einige Ähnlichkeiten zur Problematik der Promotor-Klassifizierung: Während bei der Sprache Wörter oder Sätze erkannt werden sollen, die aus unterschiedlich aneinandergereihten Lauten oder Wörtern bestehen, können in der Promotorerkennung die einzelnen DNA-Motive der Bindungsstellen beispielsweise als Phoneme angesehen werden, die Promotoren als die zu erkennenden Wörter oder Sätze. Von besonderem Interesse sind für uns daher unüberwachte Verfahren aus dem Bereich der Sprachverarbeitung [80, 81, 31].

Das in Abschnitt 3.2.2 beschriebene SARDNET-Verfahren [31] kann durch die unterschiedliche Aktivierung seiner Knoten einzelne Wörter als Aneinanderreihung von Phonemen visualisieren. Dabei entsprechen ähnliche Wortteile ähnlichen Aktivierungssequenzen auf der Karte. Durch den Ausschluss von bereits aktivierten Knoten während der Klassifizierung werden auch Wiederholungen einzelner Laute korrekt dargestellt. Bei anderen Verfahren zum Lernen von Muster-Sequenzen ist diese Möglichkeit nicht vorhanden [6]. Wiederholungen sind aber in Promotoren durchaus üblich und von funktioneller Bedeutung (siehe Abschnitt 2.1.2: das Einfügen einer zusätzlichen SRF Bindungsstelle in den Aktin-Promotor führt zu einer Spezifizierung auf das Muskelgewebe). Das SARDNET-Verfahren scheint daher am besten für die Darstellung von Promotor-Strukturen geeignet.

Ein erheblicher Unterschied zur Sprachverarbeitung besteht bei der Promo-

torerkennung darin, dass zwischen den unterschiedlichen Matrizen für die Beschreibung der Transkriptionsfaktorbindungsstellen a priori keine Ähnlichkeitsrelation gegeben ist. Während die unterschiedlichen Phoneme relativ gut durch physikalische Eigenschaften beschrieben werden können (in [31] werden z. B. der Klang, der Erzeugungsort, die Art, die Klangfarbe und die Klangfülle verwendet), die sich in reelle oder ganzzahlige Werte und somit einfach vergleichbare Merkmalsvektoren fassen lassen, gibt es diese Möglichkeit für die Matrizen nicht. So wie in der Spracherkennung beispielsweise die Laute 'ä' und 'e' zwar unterscheidbar, aber doch ähnlich sind, gibt es auch Matrizen, die dieselbe Bindungsstelle finden, obwohl sie sich in ihrem Aufbau unterscheiden. Mitunter werden auch Bindungsstellen vorhergesagt, die nicht dem tatsächlich bindenden Protein entsprechen, deren Matrixbeschreibung diesem aber sehr ähnlich ist. Dies kann etwa dann der Fall sein, wenn einzelne Nukleotide ausgetauscht, eingesetzt oder weggelassen wurden. Da die tatsächliche Transkriptionsfaktorbindung in den meisten Fällen im Verbund mit anderen Proteinen geschieht, kann auch an solchen veränderten Motiven noch eine Bindung stattfinden. Um einen echten Vergleich zwischen den gefundenen Bindungsstellen zu gewährleisten, muss daher zunächst ein Weg gefunden werden, für die Matrizen eine adäquate Ähnlichkeitsrelation zu definieren.

Im zweiten Schritt wird dann die SARDNET-Karte verwendet, auf der die unterschiedlichen Typen der Transkriptionsfaktorbindungsstellen durch die ihre jeweilige Position dargestellt werden, während die unterschiedliche Aktivierung der Knoten die Positionen der Bindungsstellen innerhalb der Sequenz charakterisiert. Auf diese Weise kann jeder Promotor durch seine abstrakte Darstellung auf der Karte repräsentiert werden. So wie bei James und Miikkulainen [31] gleichlautende Wörter durch ähnliche Repräsentationen dargestellt werden (siehe Abb. 3.13), sollte die Anwendung auf Promotoren mit gemeinsamen Strukturmerkmalen ebenfalls zu vergleichbaren Visualisierungen führen.

Um dann letztendlich eine Gruppierung der Promotoren zu erhalten, müssen diese Repräsentationen ebenfalls aufgeteilt werden. Dafür ließe sich eine herkömmliche Kohonenkarte verwenden. Allerdings müssen zunächst die mit der SARDNET-Karte erstellten Visualisierungen in geeignete Merkmalsvektoren umgewandelt werden, mit denen dann die Karte trainiert werden kann.

4.2.1 Merkmalsgenerierung für die Bindungsstellen

Zunächst soll nun das grundlegende Problem der fehlenden Ähnlichkeitsrelation zwischen den Matrizen im Mittelpunkt stehen. Für viele der bekannten Proteinbindungsstellen gibt es unterschiedliche Beschreibungen durch Matrizen. Weil oft verschiedene Forschungsgruppen oder -labors mit Transkriptionsfaktoren arbeiten, die zwar funktionell ähnlich oder verwandt sind, aber unterschiedliche Namen tragen, gibt es auch Matrizen, die dem Namen nach vollkommen unterschiedlich sind, aber letztendlich das gleiche funktionelle Motiv beschreiben. Leider sind außer der Sequenzähnlichkeit der Bindungsstellen, die mehrere ähnliche Transkriptionsfaktor-Proteine binden, bis heute keine weiteren gemeinsamen biochemischen oder -physikalischen Merkmale bekannt, anhand derer eine Klassifikation möglich wäre. Für einen Vergleich der einzelnen Matrizen ist daher nur die in ihnen enthaltene Sequenzinformation verwendbar.

Einfache Ähnlichkeitsmaße, die auf Stringebene basieren wie z. B. das von *Hamming* [39], lassen sich aufgrund der unterschiedlichen Länge nicht anwenden. Daran scheitert auch die Definition einer mathematischen Vergleichsfunktion (beispielsweise die Anwendung einer Norm) auf der Matrixebene. Auch das als *Levenshtein-Measure* oder *Edit Distance* [39] bekannte Maß ist für die Matrizen ungeeignet, weil es jede Insertion oder Änderung innerhalb der Matrix gleich bewertet. Das Einfügen eines gut definierten IUPACs in die Matrix (z. B. `A`) würde in diesem Fall der Insertion einer Wildcard (`N`) entsprechen, was der höheren Bewertung gut konservierter Positionen widerspricht.

Ein mögliches Verfahren für den Vergleich der einzelnen Matrizen untereinander stammt von der Arbeitsgruppe, die die TRANSFAC Datenbank betreut [27, 28] aus der viele der Bindungsstellen-Beschreibungen stammen. Dabei werden die einzelnen Matrizen so paarweise aliniert, dass sich bei einem spaltenweisen Vergleich der Werte ein minimaler Abstand ergibt. Bei unterschiedlich langen Matrizen werden die leeren Stellen der kürzeren Matrix mit `N`s, aufgefüllt, d. h. das Auftreten jeder der vier Basen wird an diesen Stellen als gleich wahrscheinlich betrachtet. Außerdem kann die längere der beiden Matrizen um bis zu vier `N`s verlängert werden, so dass die kürzere Matrix nicht vollständig mit der längeren überlappen muß. Damit ergibt sich für jedes Matrix-Paar ein Abstand Δ, der sich wie folgt berechnet:

$$\Delta = \min\left(\frac{\sum_{i=1}^{w} \sum_{B} |f_{B,Matrix1}(i) - f_{B,Matrix2}(i)|}{w}\right) \tag{4.1}$$

Dabei bezeichnet w die Breite der längeren der beiden Matrizen (inklusive etwaiger Verlängerungen), B steht für die vier Basen `A`, `C`, `G` und `T`, i für den Spaltenindex und f_B für die normalisierte Auftretenswahrscheinlichkeit eines Nucleotids B in der $Matrix1$ oder der $Matrix2$. Δ kann Werte zwischen 0 und 2 annehmen, wobei sich für identische Matrizen ein Wert von 0, für vollständig verschiedene Matrizen die 2 ergibt. Um die unter Umständen vorhandenen Unterschiede in der Leserichtung der DNA zu kompensieren, wird eine der beiden Matrizen invertiert[5], und auch für diesen Fall das minimale Δ berechnet. Von beiden Resultaten wird schließlich das Minimum gewählt und auf diese Weise jeweils zwei Matrizen ein Maß für die gegenseitige Ähnlichkeit zugewiesen.

Um eine globale Anordnung aller Matrizen zu finden, wurde in einem ersten Ansatz für jedes mögliche Paar (M_i, M_j) der Wert $\Delta_{i,j}$ berechnet. Anschließend wurden aus diesen Werten für jede Matrix M_i ein Vektor $\vec{v} = (\Delta_{i,1}, \cdots, \Delta_{i,n})$ bestimmt, der die jeweiligen Ähnlichkeiten zu jeder vorhandenen Matrix enthält. Die Matrizen wurden dann mit Hilfe dieser Vektoren sowohl mit *Sammon's mapping* als auch mit herkömmlichen Kohonenkarten (siehe jeweils Abschnitt 3.2) angeordnet und die Ergebnisse anhand bekannter biologischer Gegebenheiten verifiziert und bewertet. Für qualitativ hochwertige Matrizen liefert das Verfahren eine recht gute Gruppierung. Bei den schlechter definierten Matrizen[6] gibt es allerdings zahlreiche Abweichungen und zum Teil nur sehr inhomogene Gruppen. Vor allem kürzere Matrizen sind problematisch, da diese oft die erlaubte Erweiterung der längeren Matrix „ausnutzen“ um so das optimale Alignment und damit das Minimum beim der Ähnlichkeitsberechnung zu erreichen. Auf diese Weise werden die für die Proteinbindung wichtigeren *core*-Nukleotide der einen Matrix mit den für die Charakteristik der Matrix nicht so essentiellen Nukleotiden am Rand der anderen Matrix aliniert. Das Ergebnis ist dann eine verfälschte Bewertung der tatsächlichen Ähnlichkeit der beiden Matrizen.

Diese Unzulänglichkeiten waren die Motivation für die Entwicklung eines weiteren Ansatzes, der nicht so sehr auf den direkten Vergleich der Matrizen untereinander abzielt, sondern versucht, den charakteristischen Untereinheiten der den Matrizen zugrundeliegenden Sequenzen Rechnung zu tragen. Als Ausgangsidee wird ein Verfahren verwendet, das für die An-

[5] D. h. die Leserichtung wird geändert, und die Werte der Einträge werden entsprechend der Basenpaarung in der DNA vertauscht ($f_{\mathtt{A}} \leftrightarrow f_{\mathtt{T}}$, $f_{\mathtt{C}} \leftrightarrow f_{\mathtt{G}}$). Anschaulich entspricht dies einer Spiegelung der Matrix an ihrer horizontalen und ihrer vertikalen Mittelachse.

[6] D. h. Matrizen, mit einem hohen Grad an Variabilität.

ordnung von Proteinen anhand ihrer Aminosäure-Sequenz gute Ergebnisse erzielt hat [14, 15, 16]. Bei diesem Verfahren wurden für alle möglichen *Bipeptide* (d. h. direkt aufeinanderfolgende Aminosäurebausteine) innerhalb einer Proteinsequenz die Auftretensfrequenzen berechnet. Theoretisch kann jedes Bipeptid gleich häufig vorkommen. Tatsächlich gibt es jedoch charakteristische Unterschiede, welche Bausteine in welchen Proteinen verwendet werden und auch in welcher Abfolge dies geschieht. Mit den 20 existierenden Aminosäuren erhält man so für jedes Protein einen 400-dimensionalen Merkmalsvektor, der beschreibt, wie oft ein bestimmtes Bipeptid in der jeweiligen Sequenz tatsächlich vorkommt. Diese Merkmalsvektoren wurden dann auf einer Kohonen-Karte angeordnet, die anschließend zur Klassifizierung neuer Proteine verwendet werden konnte.

Analog zu den Bipeptiden bei den Proteinen lassen sich in DNA-Sequenzen Nukleotid-Abfolgen betrachten. Da es statt der 20 Aminosäuren nur vier unterschiedliche Arten von Nukleotiden gibt, lassen sich hier sogar Kombinationen aus mehr als zwei Basen betrachten, außer Di- können auch Tri- und Tetra-Nukleotide oder Kombinationen daraus untersucht werden, ohne dass die Dimension des Merkmals-Vektors allzu groß wird. Auch für die Transkriptionsfaktorbindungsstellen gilt, dass sie eine bestimmtes Muster aufweisen müssen um funktionell wirksam zu werden.

Die zu untersuchenden Matrizen enthalten in jeder ihrer Spalten die Auftrittshäufigkeit für die einzelnen Basen an dieser Position. Aus diesen Einzelwahrscheinlichkeiten lässt sich die Wahrscheinlichkeit für das Auftreten eines bestimmten Polynukleotids an jeder Stelle innerhalb der Matrix ausrechnen. Bezeichnet man die Einzelwahrscheinlichkeiten für das Auftreten einer bestimmten Base B an der Position i der Matrix mit $p_i(B)$, gilt folgende Formel für die Berechnung der Wahrscheinlichkeit eines Tetranukleotids $B_1B_2B_3B_4$ an der Position i:

$$P_i(B_1, B_2, B_3, B_4) = p_i(B_1) \cdot p_{i+1}(B_2) \cdot p_{i+2}(B_3) \cdot p_{i+3}(B_4). \tag{4.2}$$

Für Di- oder Trinukleotide gelten entsprechend verkürzte Formeln. Auch für dieses Verfahren gilt, dass man aufgrund der nicht eindeutigen Leserichtung die $P(i)$ sowohl für die Standardform der Matrix wie auch für die invertierte Variante berechnen muss. Beide Werte werden dann anschließend addiert.

Diese P_i kann man für alle möglichen Polynukleotide für jede Matrix ausrechnen. Dabei werden die für die Funktion der Bindungsstelle wichtigen

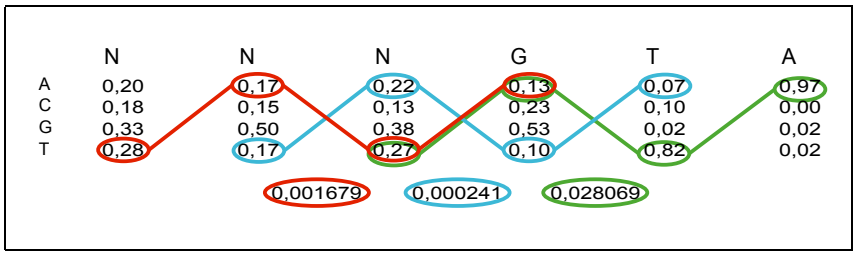

Abbildung 4.1: Hier werden die $P_i(\texttt{TATA}), i = 1, 2, 3$ berechnet. Die verwendeten Einzelwahrscheinlichkeiten sind miteinander verbunden, das Ergebnis steht darunter. In der obersten Zeile stehen die den Auftrittswahrscheinlichkeiten entsprechenden IUPAC Codes. Wie man sieht, sind die beiden letzten Basen (`T`und `A`) sehr gut konserviert, daher der hohe Wert für $P_3(\texttt{TATA})$.

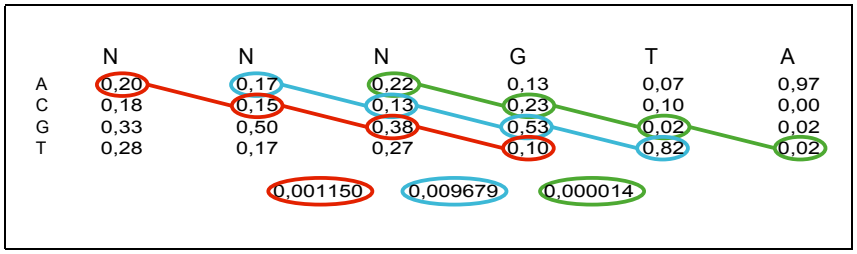

Abbildung 4.2: Ein weiteres Beispiel, wie die P_i berechnet werden. In diesem Fall handelt es sich um $P_i(\texttt{ACGT})$. Das `G` an vierter Stelle ist nur zu 53% konserviert. Deshalb ist hier $P_2(\texttt{ACGT})$ zwar deutlich höher als $P_1(\texttt{ACGT})$ und $P_3(\texttt{ACGT})$, aber nicht so hoch wie $P_3(\texttt{TATA})$ im Beispiel in der Abb. 4.1.

Nukleotidstücke eine höhere Wahrscheinlichkeit aufweisen, weil sie in den für die Erstellung der Matrix verwendeten DNA-Sequenzen besser konserviert sind als die für die Proteinbindung nicht unbedingt benötigten.

Für jede Matrix werden dann diese Einzelwahrscheinlichkeiten aufsummiert um für jedes Polynukleotid einen Gesamtwert zu erhalten. Ist N das Polynukleotid, l_N seine Länge und l_M die Länge der Matrix, erhält man für alle möglichen Nukleotidstücke N:

$$S_N(M) = \sum_{i=1}^{l_M - l_N + 1} P_i(N). \tag{4.3}$$

So entsteht für jede Matrix ein charakteristisches Profil aus der Menge

$$\mathbf{S}_{l_N}(M) = \{S_{N_1}(M), \ldots, S_{N_{(l_N)^4}}(M)\} \tag{4.4}$$

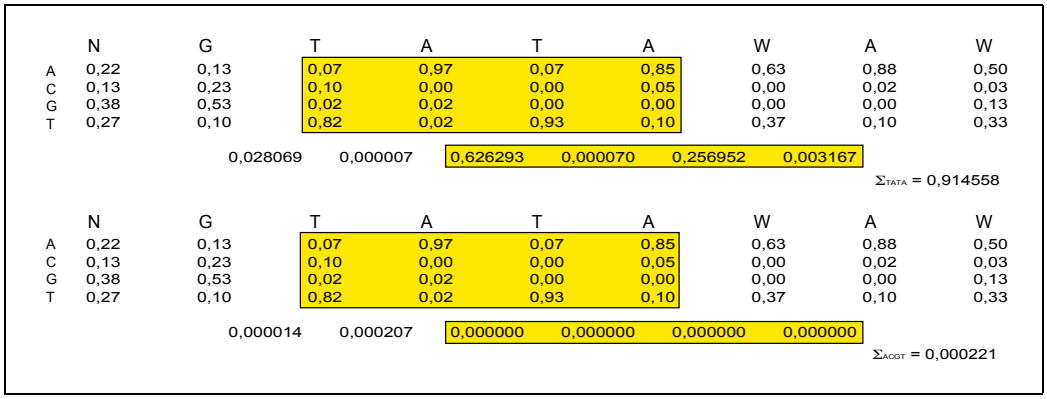

Abbildung 4.3: Ein Beispiel für die unterschiedliche Bewertung eines prominenten Nukleotidstücks bei der Berechnung der Wahrscheinlichkeiten S_N. Bei der betrachteten Matrix handelt es sich um einen Auschnitt aus der von Knippers [37] beschriebenen TATA-Matrix. Die Sequenz `TATA` als Namensgeber dieser Bindungsstelle ist für die Anlagerung der entsprechenden Proteine essentiell. (Der umrahmte Bereich markiert die entsprechend konservierte Teilsequenz.) Man sieht, dass die Auftretenswahrscheinlichkeit $S_{\texttt{TATA}}$ deutlich höher ist, als z. B. $S_{\texttt{ACGT}}$.

aller Auftretenswahrscheinlichkeiten für die jeweiligen Nukleotidstücke. Dieses Profil lässt sich als Merkmalsvektor für die Anordnung der Matrizen verwenden. Bindungsstellen, die die gleichen prominenten DNA-Motive enthalten, binden auch dieselben oder ähnliche Transkriptionsfaktoren. Es kommt bei der Ähnlichkeitsbewertung also mehr auf die enthaltenen charakteristischen Teilsequenzen an, als auf einen direkten Vergleich durch Alignments.

Die Dimension der Merkmalsvektoren ergibt sich aus allen möglichen Basenkombinationen innerhalb eines Polynukleotids N zunächst zu $(l_N)^4$. Durch die Berücksichtigung beider Leserichtungen bei der Berechnung der Merkmale ergibt sich jedoch eine platzsparende Einschränkung. Da die Basen in der DNA immer gepaart auftreten (`A`↔ `T`, `G`↔ `C`) gehört zu jedem Nukleotidstück N auf dem Strang ein inverses Stück N^* auf dem Gegenstrang (z. B. `AAA`↔ `TTT`, `ATG`↔ `CAT`, `GATA`↔ `TATC`, usw.) Bei der Berechnung der S_N genügt es, von jedem dieser Paare (S_N, S_{N^*}) nur jeweils einen Vertreter zu behalten. Dadurch reduziert sich bei ungeradzahligen Nukleotidstücken die Dimension des Merkmalsvektors auf die Hälfte. Bei geradzahligen Sequenzen ist zu beachten, dass einige der Sequenzen zu sich selbst invers sind (sogenannte *Palindrome*); deshalb bleiben hier mehr als die Hälfte der Merkmale erhalten. Für Tri-Nukleotide erhält man also 32 statt 64 Merkmale, für Tetra-Nukleotide sind es 136 statt 256. Auf diese Weise werden redundante Komponenten im

Merkmalsvektor entfernt und durch die Reduktion der Dimension des Merkmalsraumes auch das Training entsprechend beschleunigt.

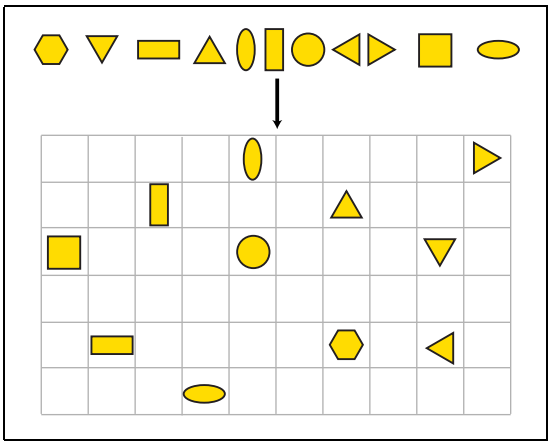

Abbildung 4.4: Schematische Darstellung der Anordnung von Bindungsstellen-Matrizen auf einer Kohonenkarte. Dabei stehen ähnlich geformte Muster für miteinander verwandte Transkriptionsfaktoren.

Die Untersuchung ihrer Struktur bietet also die Möglichkeit, für jede Matrix einen von ihrer Länge unabhängigen Merkmalsvektor zu finden, der insbesondere die am besten konservierten Bereiche und damit für die Bindungsstelle charakteristischen Basen berücksichtigt. Lassen sie sich auf diese Weise, wie in Abb. 4.4 dargestellt, auf einer Kohonenkarte anordnen, wäre das der Beweis, dass es sich bei diesen Beschreibungen durch Polynukleotide um ein biologisch relevantes Ähnlichkeitsmaß handelt.

4.2.2 Die Klassifikation von Sequenzen von Bindungsstellen

Als nächstes wollen wir uns mit der für die Promotorklassifizierung essentiellen Anordnung der Bindungsstellen beschäftigen. Die SARDNET-Methode [31] ermöglicht es, sowohl die Reihung einzelner Signale als auch, durch eine kleine Modifikation des Algorithmus, deren Abstände auf einer selbstorganisierenden Karte darzustellen.

So wie sich bei der Spracherkennung einzelne Laute in bestimmten Bereichen der Karte anordnen, sind es jetzt die untereinander ähnlichen Bindungsstellen-Motive, die sich zu Clustern zusammenfassen lassen. Die zeitliche Abfolge der

Phoneme wird durch die unterschiedliche Aktivierung der einzelnen Knoten wiedergegeben. Diese Methodik lässt sich auf die Abfolge der Bindungsstellen innerhalb der Promotersequenz übertragen. Hier sind es statt der zeitlichen räumliche Abstände, die zu berücksichtigen sind.

Das Training der Karte erfolgt analog zur Sprachverarbeitung wie in Abschnitt 3.2.2 beschrieben. Anstelle von Phonemsequenzen werden Folgen von Bindungsstellen verwendet. Im Unterschied zu den Wörtern, bei denen der Wortanfang als Start für die zeitliche Anordnung der Phoneme verwendet wird, kann bei den Promotoren der Transkriptionsstart als maßgebliches Element für die räumliche Anordnung benutzt werden. Die Ausschnitte aus den DNA-Sequenzen der Promotoren müssen dann so gewählt sein, dass bei allen zu vergleichenden Stücken der Initiator innerhalb der Sequenz immer an der gleichen Position zu finden ist. Unterschiedlich lange Sequenzen stellen insofern kein Problem dar, als diese am vorderen Ende durch `N`s (die IUPAC Wildcard) aufgefüllt werden können. In diesen Bereichen werden keine Bindungsstellen gefunden, insgesamt ist die Länge der betrachteten Sequenzen jedoch gleich und die Positionen der einzelnen Bindungsstellen-Motive sind auf diese Weise relativ zum Transkriptionsstart vergleichbar. Diese Erweiterung stellt keine Veränderung der biologischen Verhältnisse dar, da weder Positionen verändert, noch zusätzlich Bindungsstellen eingefügt werden.

Während in der Spracherkennung die Phoneme innerhalb eines Wortes unmittelbar aufeinander folgen, gibt es bei den Promotoren längere Abschnitte ohne Information zwischen den einzelnen Bindungsstellen. Diese unterschiedlich großen Abstände haben durchaus biologische Bedeutung; sie können beispielsweise durch die dreidimensionale Struktur der verschiedenen Proteine bedingt sein. Es ist deshalb nicht ausreichend, lediglich die Abfolge der einzelnen Motive zu betrachten. Aus diesem Grund wurde eine entscheidende Änderung am SARDNET-Algorithmus vorgenommen. Anstatt die Aktivierung der Knoten bei jeder Präsentation eines Einzelmusters um den gleichen Betrag zu ändern, wird sie in Abhängigkeit vom Abstand zwischen dem vorherigen und dem aktuellen Muster vermindert. Dies entspricht von der Idee her dem von Chappell und Taylor [6] beschriebenen Potentialansatz (siehe Abschnitt 3.2.2). Die Aktivierung wird jeweils anteilig zur Gesamtsequenzlänge reduziert. Auf diese Weise werden zusätzlich zur Reihenfolge auch unterschiedliche Abstände bzw. die Positionen der einzelnen Bindungsstellen auf der Karte visualisiert. Der geänderte SARDNET-Algorithmus sieht dann wie folgt aus:

1. Die Aktivierung a jedes Knoten wird auf 0 gesetzt.

2. Finde einen Knoten n_c mit $a_i = 0$ und minimalem Abstand

$$\sum_{j=1}^{m}(\xi_j(t) - \omega_{ij}(t))^2.$$

3. Setze $a_c = 1$.

4. Adaptiere die Gewichtsvektoren w_k aus der Nachbarschaft $N_c(t)$ analog zu Kohonens Algorithmus.

5. Reduziere die Aktivierung aller anderen Knoten mit $a > 0$ um die Größe $d = \frac{Pos(\xi(t)) - Pos(\xi(t-1)}{l_{Seq}}$, wobei $Pos(x)$ für die Position des Motivs innerhalb der Sequenz, l_{Seq} für die Länge der Sequenz steht.

6. Prüfe, ob das Ende der Sequenz erreicht wurde, ansonsten fahre mit Schritt 2 fort.

7. Reduziere die Aktivierung aller Knoten mit $a > 0$ um den Wert

$$d_{end} = \frac{(l_{Seq} - 1) - Pos(\xi(t_{end})}{l_{Seq}},$$

wobei $\xi(t_{end})$ das letzte innerhalb der Folge präsentierte Muster ist.

Nach dem Training können für jeden Promotor die einzelnen Bindungsstellen auf der Karte visualisiert werden.
Die so entstandenen Visualisierungen müssen nun verglichen werden, um eine Gruppierung der untersuchten Promotoren zu ermöglichen. Da die Anzahl der in den Promotorsequenzen gefundenen Bindungsstellen die von Phonemen innerhalb von Wörtern um den Faktor 5-15 übertrifft, und hier im Gegensatz zur Spracherkennung deutlich mehr redundante Muster auftreten, sind Ähnlichkeiten zwischen einzelnen Visualisierungen leider nicht so offensichtlich wie bei den Beispielen aus der SARDNET-Veröffentlichung (siehe Abb. 3.13). Bereits für den Vergleich weniger Promotoren ist es daher hilfreich, diese Gruppierung zu automatisieren. Sollen Promotoren im genomischen Maßstab verglichen werden, ist dies ohnehin notwendig.

Um die erhaltenen SARDNET-Karten und damit die Promotorsequenzen clustern zu können, müssen sie in Form von Merkmalsvektoren erfassbar gemacht werden. Der triviale Ansatz wäre, die Knoten der SARDNET-Karte durchzunummerieren und einen Merkmalsvektor zu verwenden, dessen Dimension mit der Anzahl der Knoten übereinstimmt. Als Werte für die Komponenten würde dann die jeweilige Aktivierung des zugeordneten Neurons

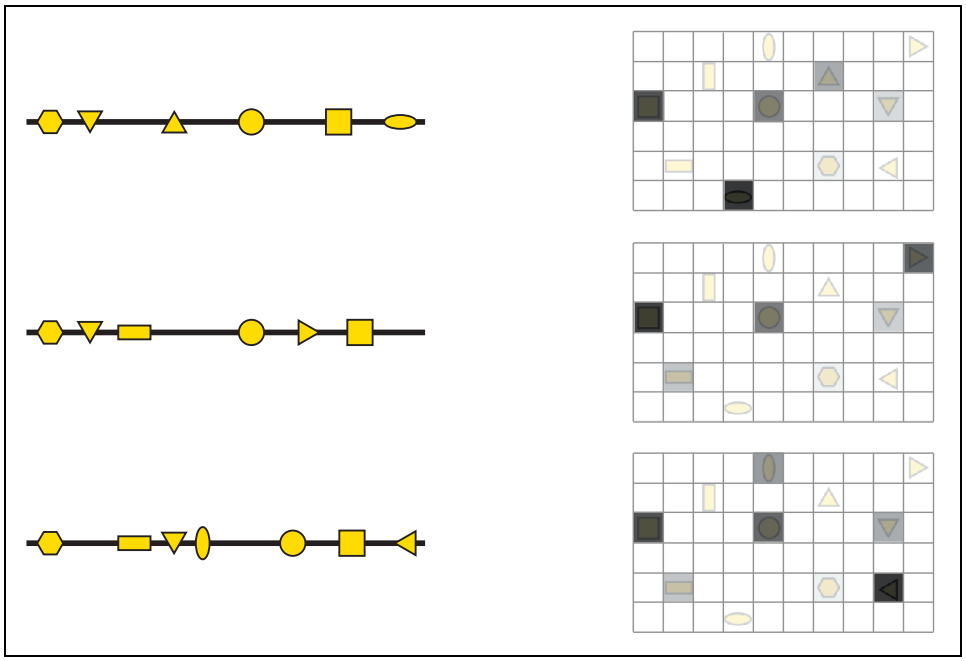

Abbildung 4.5: Diese Abbildung zeigt (schematisch) die Visualisierung der in den Promotoren gefundenen Bindungsstellen auf den SARDNET-Karten. Die Position ähnlicher Bindungsstellen innerhalb der Karte bleibt dabei gleich, durch die jeweilige Aktivierung (hier durch die unterschiedlichen Graustufen dargestellt) werden die Reihenfolge und der Abstand konserviert.

verwendet. Bei dieser Vorgehensweise wird allerdings die Nachbarschaftserhaltung der Kohonenkarte zerstört. Betrachtet man beispielsweise zwei Karten der Breite 10, wobei auf der einen der Knoten n_1 (mit den Koordinaten $(1, 1)$), auf der anderen der Knoten n_{11} (mit den Koordinaten $(2, 1)$) aktiviert ist. Die beiden Knoten liegen nebeneinander und repräsentieren also ähnliche Muster. Im Merkmalsvektor geht dies jedoch verloren, die Aktivierungen werden in jeweils unterschiedlichen Komponenten und damit verschiedenen Dimensionen des Merkmalsraums gespeichert. Dieser Weg ist daher für die Weiterverarbeitung nicht geeignet.

Eine andere Möglichkeit wäre die Erfassung der „Aktivierungsspur“ auf der jeweiligen Karte. Dabei werden die Gewinner-Knoten in der Reihenfolge ihrer Aktivierung miteinander verbunden. Die so entstehende Spur könnte dann zum Vergleich der Karten herangezogen werden [80, 81]. Dabei würde aber lediglich die Reihung der einzelnen Knoten und deren Position berücksichtigt, nicht der durch die unterschiedliche Aktivierung vermittelte Abstand zwi-

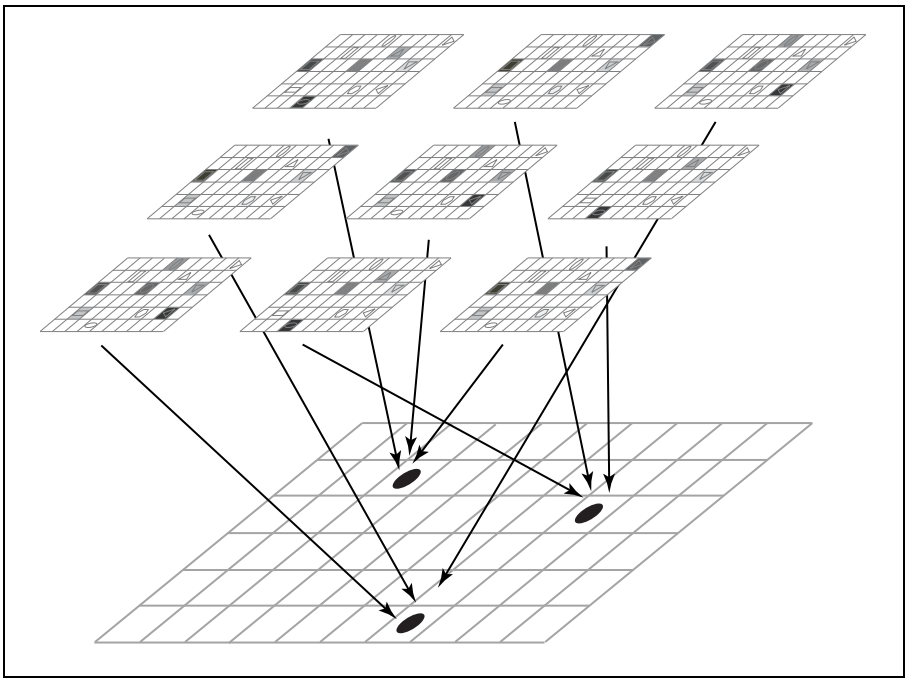

Abbildung 4.6: Die mittels SARDNET erhaltenen Visualisierungen der unterschiedlichen Promotoren werden auf einer normalen Kohonenkarte angeordnet. Dazu muss zunächst wieder eine Möglichkeit gefunden werden, die SARDNET Karten in geeignete Merkmalsvektoren zu übersetzen.

schen den Bindungsstellen. Außerdem wären wegen der unterschiedlichen Anzahl aktivierter Neuronen die Längen der entstandenen Spuren verschieden, weshalb diese Vorgehensweise insgesamt ebenfalls nicht geeignet erscheint.

De Ketelaere et al. [34] haben 1997 ein System aus zwei hierarchisch aufgebauten SOMs für die Klassifizierung von Messdaten vorgestellt, das im Vergleich „mit anderen auf das Problem angewendeten statistischen und heuristischen Algorithmen nicht zu übertreffen war“. Das Ziel war eine Einteilung der Muster in drei unterschiedliche Klassen, die, nach Abschluß des Trainings jeweils zu $97,5\%$, 84% und 65% richtig erkannt wurden. Dabei musste ebenfalls eine Methode gefunden werden, um unterschiedliche Aktivierungen auf der einen Karte in einen Merkmalsvektor zu fassen, der für die weitere Klassifikation auf der anderen Karte geeignet erschien.

Für jedes zu klassifizierende Objekt wurde eine Folge von 6 Messungen auf einer herkömmlichen Kohonenkarte visualisiert. Um diese Visualisierungen

weiter zu klassifizieren wurden dann alle Knoten der Karte jeweils als „Aktivitätszentrum“ einer zugehörigen „Aktivitätsumgebung“ betrachtet und mit Hilfe einer leicht modifizierten Gaußschen Dichtefunktion die zugehörige Gesamtaktivität aus allen Knoten der Karte so berechnet, dass weiter entfernt liegende erregte Neuronen weniger stark in den Wert mit einfließen als die unmittelbar benachbarten. Ist n_j der betrachtete Knoten (das Aktivitätszentrum) mit den Koordinaten (x_j, y_j) und sind $n_i, i = 1, \ldots, l$ die aktivierten Gewinner-Knoten mit den Koordinaten (x_i, y_i), ergibt sich die Gesamtaktivität für die Umgebung des Zentrums n_j zu

$$A(n_j) = \sum_{i=1}^{l} e^{-\frac{\sqrt{(x_i-x_j)^2+(x_i-y_j)^2}}{2\sigma^2}}. \tag{4.5}$$

Die Autoren betrachten nur quadratische Karten und geben als „experimentell ermittelten optimalen Wert“ für σ das $0,3$-fache des Kartendurchmessers an. Diese Verallgemeinerung führt bei unterschiedlich großen Karten allerdings zu sehr verschiedenen Bewertungen von Knoten, die nicht in der direkten Nähe des Aktivitätszentrums liegen. Bezeichnet d sowohl den Durchmesser der Karte als auch den maximal möglichen Abstand in jeder Richtung, dann beträgt der Anteil der für zwei voneinander maximal entfernte Knoten in (4.5) eingeht[7]:

$$e^{-\frac{\sqrt{d^2+d^2}}{2\cdot(0.3d)^2}} = e^{-\frac{\sqrt{2}d}{0,18d^2}} \approx e^{-\frac{7,857}{d}}. \tag{4.6}$$

Wie man aus (4.6) sieht, wächst der Anteil, den ein aktivierter Knoten in die Gesamtaktivität einbringt mit der Größe der Karte. Tatsächlich wird auf einer Karte mit 10×10 Neuronen ein vom Aktivitätszentrum maximal entfernter Knoten im Vergleich mit dem Zentrum nur mit 49% bewertet, bei einer mit 20×20 Knoten trotz des größeren Abstandes aber mit annähernd 70%. Diese Wahl von σ widerspricht also dem Gedanken einer lokal beschränkten Aktivitätsumgebung. Eine Alternative wäre, einen von der Kartengröße unabhängigen Wert für σ zu bestimmen.

Die Anzahl der Gewinner-Knoten ist bei de Ketelaere [34] für alle unterschiedlichen Klassifikationen konstant, es werden immer jeweils sechs Muster präsentiert.[8] Auf den SARDNET-Karten für die Promotorerkennung ist da-

[7] Tatsächlich müsste für den maximalen Abstand $d - 1$ gesetzt werden. Die Wahl von d führt zu einer formal einfacheren Abschätzung und einem niedrigeren Wert, was für die Schlussfolgerung demnach unerheblich ist.

[8] Einzelne Knoten können mehrfach aktiviert werden. Dann werden ihre Koordinaten in (4.5) mehrfach berücksichtigt. Die Summe der Knotenaktivierungen ist im Unterschied zu den SARDNET Karten also immer gleich. (4.5) könnte jedoch durch einfaches Multipli-

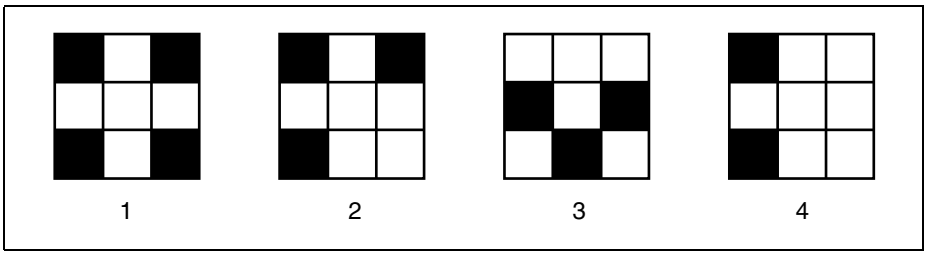

Abbildung 4.7: Die vier im Beispiel verwendeten Aktivierungsmuster. Jedes Teilquadrat entspricht einem Knoten der Karte. Die aktivierten Knoten sind eingefärbt. Die Karten 1 und 2 unterscheiden sich lediglich in der Aktivierung eines zusätzlichen Neurons. Das Aktivitätsmuster auf Karte 4 lässt sich als Teilmuster der Karten 1 und 2 auffassen. Karte 3 hat mit allen anderen am wenigsten gemeinsam.

gegen die Anzahl der aktivierten Knoten von Promotor zu Promotor unterschiedlich. Da in (4.5) alle erregten Neuronen zum Wert von $A(n_j)$ beitragen, ergeben sich bei einer unterschiedlichen Anzahl von Knoten erhebliche Differenzen bei den Komponenten der Merkmalsvektoren.

Dies soll an einem einfachen Beispiel erläutert werden. Betrachtet werden die in Abbildung 4.7 dargestellten Karten. Berechnet man zunächst nach (4.5) die Merkmalsvektoren $v_i, i = 1, \ldots, 4$ und anschließend die euklidischen Abstände untereinander erhält man folgende Ergebnisse:

d	v_2	v_3	v_4
v_1	1,11	1,00	2,05
v_2		1,14	1,11
v_3			1,69

Tabelle 4.2: Die Abstände zwischen den einzelnen Merkmalsvektoren $v_1, \ldots, v_4$. Die tatsächlichen euklidischen Abstände wurden derart normiert, dass der Wert der kleinsten Abstands $d_{v_1,v_3} = 1,00$ beträgt, um die Vergleiche mit den folgenden Ergebnissen zu vereinfachen.

Aus Tabelle 4.2 lässt sich leicht erkennen, dass die Merkmalsvektoren v_1 und v_3 den geringsten Abstand voneinander haben; demnach wären die Karten 1 und 3 aus Abbildung 4.7 die zueinander ähnlichsten. Analog wären Karte 1 und Karte 4 die zueinander unähnlichsten. Dieser Vergleich der Aktivierungsmuster entspricht damit leider nicht den Erwartungen und resultiert im

zieren jedes Summanden mit der Aktivierung des jeweiligen Knotens für die Verwendung von mit der SARDNET Methode erstellten Karten angepasst werden.

wesentlichen aus den ungleichen Werten der Gesamtaktivität aufgrund der unterschiedlichen Anzahl der berücksichtigten Gewinner-Knoten.

Normiert man die Merkmalsvektoren entsprechend ihrer durchschnittlichen Komponentenwerte erhält man für die Abstände die Resultate in Tabelle 4.3.

d	v_2^*	v_3^*	v_4^*
v_1^*	1,00	1,27	1,93
v_2^*		1,84	1,50
v_3^*			2,31

Tabelle 4.3: Die Abstände zwischen den normierten Merkmalsvektoren $v_1^*, \ldots, v*_4$. Wieder wurde der Wert des kleinsten Abstands zu $1,00$ gesetzt, die anderen entsprechend angepasst. Die Komponenten $\beta_{i,1}^*, \ldots, \beta_{i,9}^*$ der normierten v_i^* berechneten sich aus den Komponenten $\beta_{i,1}, \ldots, \beta_{i,9}$ der ursprünglichen Vektoren v_i zu: $\beta_{i,j}^* = \frac{\beta i,j}{\sum_{j=1}^{9} \beta_{i,j}}, j = 1, \ldots, 9$.

Wie man sieht, werden nun zumindest die Aktivierungsmuster der Karten 1 und 2 als die ähnlichsten erkannt, die der Karten 3 und 4 als die unähnlichsten. Auch der Abstand $d_{v_2^*,v_3^*}$ hat im Vergleich zu d_{v_2,v_3} zugenommen. Insofern hat die Normierung also das Ergebnis verbessert. Allerdings wäre es immer noch wünschenswert, den Abstand zwischen den Merkmalsvektoren der Karten 1 und 4 zu verringern, da es sich beim Muster auf Karte 4 offensichtlich um ein Teilmuster von Karte 1 handelt und gerade die Erkennung solcher Übereinstimmungen bei der Untersuchung von Promotoren auf gemeinsame Teilstrukturen eine wesentliche Rolle spielt.

Zu diesem Zweck reduzieren wir die Menge der bei der Berechnung der Merkmalsvektorkomponenten berücksichtigten Knoten. Anstatt wie in (4.5) alle aktivierten Neuronen mit einzubeziehen, verwenden wir als Aktivitätsumgebung einen scharf abgegrenzten Bereich der Karte. Dazu werden überlappende Fenster auf der Karte betrachtet, innerhalb derer die jeweiligen Aktivierungen einfach addiert werden. Auf die Abstandsabhängigkeit, wie sie in (4.5) durch die Verwendung der Gaußschen Dichtefunktion induziert wird, verzichten wir bei diesem Ansatz ganz. Auf diese Weise gehen nur die Werte der unmittelbaren Nachbarn des jeweiligen Aktivitätszentrums in den Merkmalsvektor ein. Gegeben sei eine Karte mit den Dimensionen x und y, bestehend aus den Knoten n_{ij} $(1 < i < x, 1 < j < y)$ in rechteckiger Anordnung. Verwendet man nun ein quadratisches Fenster der Breite b, das jeweils zeilenweise über die Karte geschoben wird, erhält man einen Merkmalsvektor

w mit $(x-b+1)\times(y-b+1)$ Komponenten ω_i. Für die Werte der einzelnen

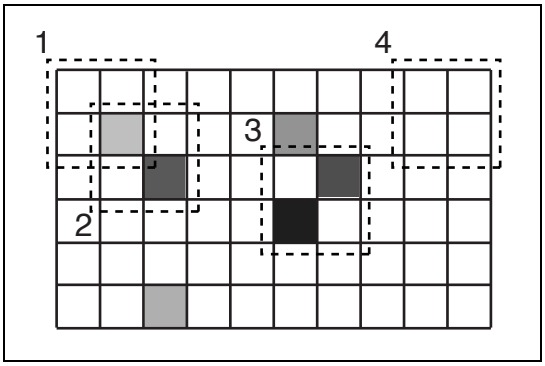

Abbildung 4.8: Verfahren zur Gewinnung von Merkmalsvektoren für die Weiterverarbeitung der in den Promotoren enthaltenen Folgen von Transkriptionsfaktoren. Um die Übersicht zu bewahren, wurden exemplarisch 4 Fenster ausgewählt. Die Nummerierung dient der Identifikation, und bezeichnet keine Abfolge. Die ersten 3 Fenster tragen jeweils unterschiedliche Werte zum Merkmalsvektor bei; Fenster 1 liefert dabei den niedrigsten, Fenster 3 den höchsten Wert. In Fenster 4 befinden sich keine aktivierten Knoten, dementsprechend wäre die zugeordnete Komponente des Merkmalsvektors 0.

Komponenten werden die Aktivierungen der jeweils innerhalb des Fensters liegenden Knoten addiert:

$$\omega_{x\times(j-1)+i} = \sum_{s=j}^{j+b-1}\sum_{t=i}^{i+b-1} a(n_{s,t}) \qquad \forall \quad i=1,\ldots,x-b+1, j=1,\ldots,y-b+1. \tag{4.7}$$

Die Überlappung der einzelnen Fenster dient dazu, Bereiche zu unterscheiden, in denen ein Knoten mit hoher Aktivierung bzw. mehrere Knoten mittlerer Aktivierung liegen. Ohne Überlappung ergäbe sich, der Summierung der Aktivierungswerte wegen, für solche unterschiedlich besetzte Bereiche unter Umständen derselbe Wert im Merkmalsvektor. Durch die Überlappung erhält man zwar insgesamt mehr Merkmale, in der eben beschriebenen Situation gehen solche Bereiche dann aber unterschiedlich in die Bewertung ein. Im ersten Fall erhält man ein Fenster mit hoher Gesamtaktivierung, die der umliegenden ist eher niedrig. Im zweiten Fall erhält man in allen Fenstern höhere Werte. Dementsprechend unterscheiden sich auch die Komponenten der Merkmalsvektoren.

Wendet man das beschriebene Verfahren mit $b = 2$ wieder auf die in Abbildung 4.7 dargestellten Karten, erhält man die folgenden Abstände:
Mit den nach (4.7) berechneten Merkmalsvektoren sind sich die Karten 1

d	w_2	w_3	w_4
w_1	1,00	1,41	1,41
w_2		2,24	1,00
w_3			2,45

Tabelle 4.4: Die Abstände zwischen den nach (4.7) berechneten Merkmalsvektoren $w_1, \ldots, w_4$. Eine Normierung war in diesem Fall nicht notwendig, da der kleinste Abstand bereits den Wert $1,00$ hatte.

und 2 bzw. die Karten 2 und 4 am ähnlichsten. Außerdem gilt jetzt zumindest $d_{w_4,w_i} \geq d_{w_j,w_i} \forall i = 1,2,3; j = 1,2,4$. Damit stellt sich die Karte 3 als die „unähnlichste“ zu allen anderen dar; das Ergebnis entspricht damit am ehesten den Erwartungen. Wie bereits dargelegt wurde, erhöhen sich bei größeren Karten die Beiträge der nicht unmittelbar benachbarten Knoten zu den jeweiligen Komponenten der Merkmalsvektoren. Die Unterschiede fallen dementsprechend noch mehr zu Ungunsten von (4.5) aus.

Damit erweist sich (4.7) in der Theorie als am besten geeignetes Verfahren für die Bestimmung von Merkmalsvektoren aus den Visualisierungen der Transkriptionsfaktorbindungsstellen auf den SARDNET Karten. Zusammen mit der in Abschnitt 4.2.1 dargestellten Vorgehensweise ist damit der theoretische Entwurf des Verfahrens zur Klassifizierung von Promotoren auf der Basis der enthaltenen Proteinbindungs-Motiven abgeschlossen.

4.3 Zusammenfassung

In diesem Kapitel wurden die theoretischen Grundlagen für das neue Verfahren bereitgestellt. Im Unterschied zu den existierenden Algorithmen basiert es nicht auf der Erkennung spezieller Sequenz-Muster, sondern auf dem Vergleich allgemeiner Promotor-Elemente und deren Anordnung. Dafür wurde zunächst ein längenunabhängiges Ähnlichkeitsmaß für Bindungsstellen-Matrizen entwickelt. Anschließend wurden verschiedene Methoden für den Vergleich von Sequenzen von Mustern diskutiert. Im nächsten Kapitel werden die einzelnen Teile des Verfahrens auf ihre praktische Anwendbarkeit hin untersucht.

Kapitel 5

PromoterMap in der Anwendung

Ausgehend von den theoretischen Überlegungen im vorangegangenen Abschnitt wurde nun die Praxistauglichkeit des Verfahrens erprobt. Da das gewünschte Ergebnis die funktionelle Anordnung von Promotoren auf einer selbstorganisierenden Karte ist, erschien *PromoterMap* als geeignete Bezeichnung für das Verfahren. Zuerst wurde die Verwendbarkeit der einzelnen Schichten der Methode (siehe Abb. 5.1), danach die des gesamten Pakets betrachtet.

Dabei sei an dieser Stelle nochmals an die bereits in Abschnitt 1.2 erwähnte Problematik der sehr unterschiedlichen Datenlage erinnert. Für die Erkennung von Transkriptionsfaktorbindungsstellen existieren beispielsweise über 300 Matrizen. Die tatsächliche Anzahl unterschiedlicher Transkriptionsfaktoren lässt sich noch nicht genau festlegen, nach jüngsten Schätzungen wird hier von einer Größenordnung von mehreren hundert Proteinen ausgegangen [5]. Auch wenn einzelne Bindungsstellen durch mehrere unterschiedliche Matrizen beschrieben werden, ist hier bereits ein großer Anteil der Daten bekannt. Bei den Promotoren hingegen sind im Vergleich zur geschätzten Gesamtanzahl im menschlichen Genom bisher nur wenige Sequenzen verfügbar.

In der verwendeten Version 62 der Promotordatenbank EPD [54] finden sich 274 humane Promotorsequenzen, was einem Anteil von etwa 0,9% der vermuteten 30.000 Gene entspricht. Betrachtet man zusätzlich auch noch die Sequenzen für andere Wirbeltiere[1], erhält man insgesamt 943 Promotoren.

[1] Dabei wird davon ausgegangen, dass sich bei den unterschiedlichen Wirbeltierspezies viele der Gene ähneln, so dass die Einbeziehung der Daten durchaus sinnvoll ist.

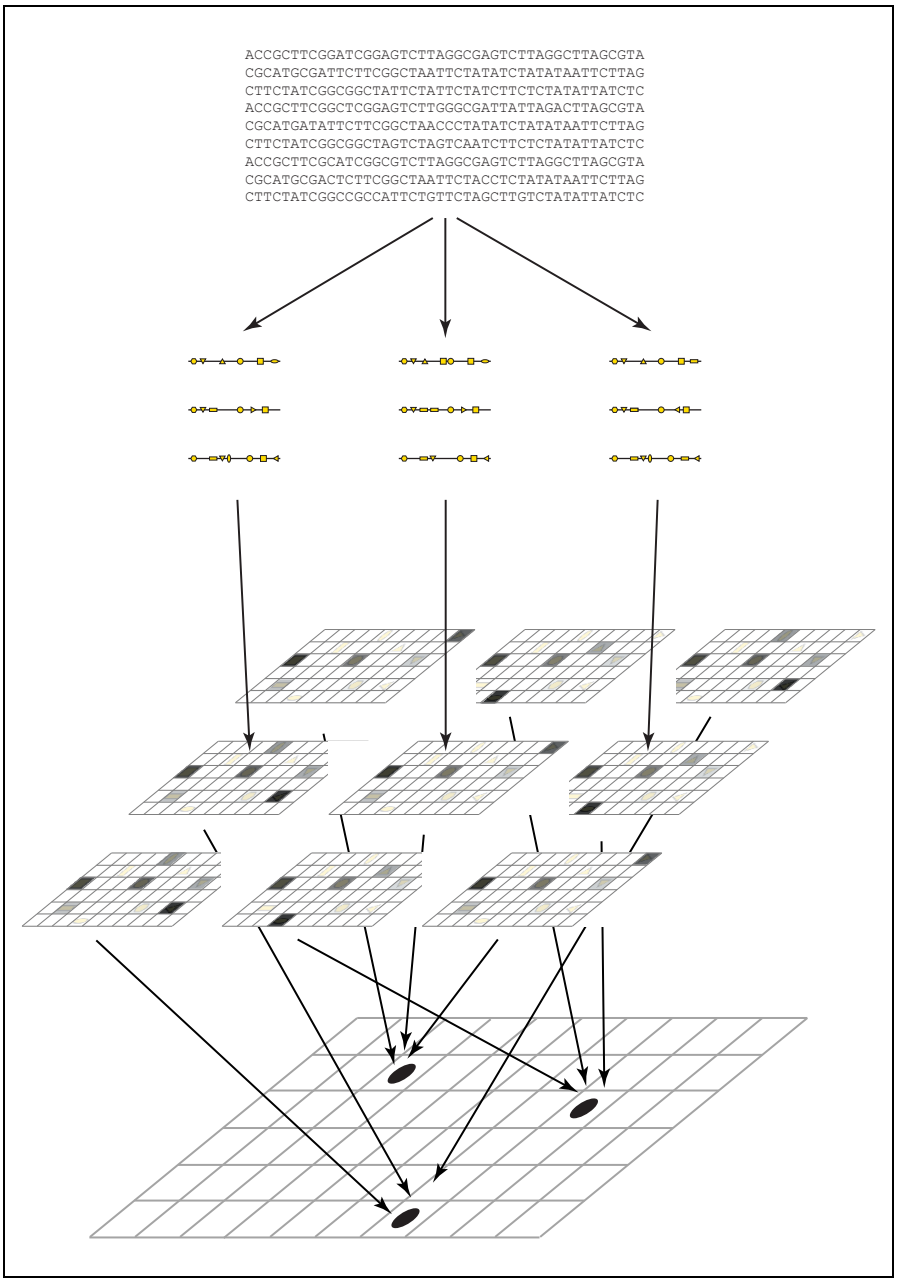

Abbildung 5.1: Schematische Darstellung des Aufbaus von *PromoterMap*. Ausgehend von den DNA-Sequenzen einzelner Promotoren werden Folgen von putativen Transkriptionsfaktorbindungsstellen generiert, die dann auf den SARDNET-Karten angeordnet werden. Diese wiederum werden für die Generierung von Merkmalsvektoren benutzt, die auf einer herkömmlichen Kohonenkarte angeordnet werden.

Allerdings enthält die EPD viele zueinander ähnliche Sequenzen. Durch Anwendung der vorgegebenen Filterfunktion[2] zur Redundanzvermeidung verblieben noch 192 humane Sequenzen. Ein zusätzliches Problem stellen die Längenvariationen der vorhandenen Sequenzen dar. Teilweise sind mehrere hundert bis tausend Nukleotide bekannt, was eine für unsere Analysen ausreichende Datenmenge darstellt, teilweise sind jedoch nur deutlich kürzere, den Transkriptionsstart flankierende Bereiche vorhanden, so dass lediglich der core promoter enthalten ist. Die zuletzt genannten Sequenzen sind damit ebenfalls nur eingeschränkt für unsere Zwecke geeignet.

Trotz dieser Einschränkungen konnten mehrere Gruppen funktionell verwandter Promotoren zusammengestellt werden, um *PromoterMap* zu testen. Die erhaltenen Resultate werden in den folgenden Abschnitten dargelegt. Außerdem werden wichtige zusätzliche Ergebnisse vorgestellt, die sich mit Hilfe dieser Methode ableiten lassen.

5.1 Von der Sequenz zur Promotorgruppierung

Zunächst wurde das in Abschnitt 4.2.1 vorgestellte Verfahren in der Praxis untersucht. Danach folgte die Anwendung des SARDNET-Algorithmus auf die aus den Promotorsequenzen gewonnenen Folgen von Bindungsstellen und deren Gruppierung.

5.1.1 Bindungsstellen

Der Grundstein von *PromoterMap* ist die Definition einer Ähnlichkeitsrelation zwischen den Matrizen der einzelnen Transkriptionsfaktorbindungsstellen. Diese erlaubt es, die per Computer ermittelten Vorhersagen für die Bindung von Proteinen vom statistischen Modell ausgehend in einen biologisch funktionalen Kontext zu bringen. Die unterschiedlichen Beschreibungen der Bindungsstellen sollten so zusammengefasst werden, dass die in der Variabilität der Sequenzen begründeten Unterschiede der gefundenen statistischen Beschreibungen biologisch sinnvoll verglichen werden können. Das in Abschnitt 4.2.1 vorgestellte Konzept musste daher zunächst auf eben diese biologische Relevanz hin untersucht werden.

[2]Damit wird von Sequenzen, die zu mehr als 50% übereinstimmen, jeweils nur ein Exemplar beibehalten [54].

Als Testdatensatz dienten dabei 341 verschiedene Matrizen für etwa 120[3] verschiedene Proteinbindungsstellen. Diese stammten zum Teil aus der TRANSFAC Datenbank [27] oder wurden aus der Literatur zusammengestellt [58]. Sie lassen sich nach ihrer Herkunft in vier große Klassen einteilen. Es wurden Motive aus der DNA von Wirbeltieren, Insekten, Pflanzen und Pilzen untersucht. Dabei gibt es für manche der Proteinbindungsstellen bis zu sechs unterschiedliche statistische Beschreibungen, für andere existiert nur genau eine einzige Matrix.

Für alle 341 Matrizen M wurden analog zu (4.3) und (4.4) die Profile für Di-, Tri- und Tetranukleotide $\mathbf{S}_2(M), \mathbf{S}_3(M)$ und $\mathbf{S}_4(M)$ berechnet und entsprechend der in Abschnitt 4.2.1 beschriebenen Vorgehensweise die enthaltenen Redundanzen entfernt. Die so erhaltenen Profile wurden dann als Merkmalsvektoren für das Training einer Kohonenkarte verwendet. Dabei wurden sowohl jeweils die einzelnen Profile als auch Kombinationen aus diesen verwendet, so dass die Anzahl der Dimensionen des Merkmalsraums zwischen 8 und 176 lag.

Für die Erstellung der Kohonenkarte wurde zunächst die am Lehrstuhl von T. Kohonen an der Universität Helsinki entwickelte SOM_PAK Software [39, 84] benutzt. Dieses Programmpaket stellt alle benötigten Werkzeuge für die Erstellung und Visualisierung von selbstorganisierenden Karten analog zum in Abschnitt 3.2.1 beschriebenen Algorithmus zur Verfügung. Dabei stehen zwei verschiedene Möglichkeiten zur Verfügung, um den Anfangszustand der Karte zu bestimmen. Entweder man geht von einer gitterförmigen Anordnung der Gewichtsvektoren entlang der beiden Hauptkomponenten des Merkmalsraums aus [39], oder man beginnt mit einer zufälligen Anordnung der Gewichte. Im zweiten Fall teilt sich das Training der Karte in die „Vorordnungsphase“ und die „Feineinstellungsphase“ auf. Im ersten Teil des Trainings werden dabei mit einer vergleichsweise hohen Lernrate ($0.1 < \alpha < 0.8$) und einem großen Anfangsradius r_0[4] innerhalb weniger tausend Schritte die Gewichtsvektoren zunächst grob angeordnet, bevor im zweiten Teil mit Lernraten α im Bereich von 0.01 bis 0.05 und Umgebungsradien von 2 bis 5 in mehreren zehntausend Lernschritten das Training abgeschlossen wird. Bei den linearen Ausgangsformation der Gewichtsvektoren wird beim Training nur die „Feineinstellung“ verwendet.

[3] Hier ist eine exakte Angabe sehr stark abhängig davon, inwieweit man einzelne Transkriptionsfaktor-Proteine als unterschiedlich betrachtet.

[4] Laut Kohonen kann dieser so groß gewählt werden, dass von der Anfangsumgebung etwa die Hälfte der Karte überdeckt wird [39].

Die erzeugten Anordnungen wurden anhand bekannter Daten auf ihre biologische Relevanz hin untersucht. Dabei wurde als Kriterium verwendet, ob unterschiedliche Matrizen, für die eine Ähnlichkeit bereits aus der Literatur bekannt war, sich in dementsprechend geringem Abstand voneinander auf der Karte wiederfinden. Abgesehen von trivialen Fällen, wie etwa den verschiedenen EVI oder GATA Matrizen, gibt es auch weniger offensichtliche Beipiele. So berichten etwa Becker et al., dass das STAT3 Protein ein ähnliches Bindungsverhalten aufweist wie Proteine der REL-Familie, obwohl es nur geringe Sequenzübereinstimmungen zwischen beiden gibt [3]. Ein weiteres Beispiel ist die ähnliche Bindung von ETSF und PU1 Proteinen [56]. Die Wiedergabe solcher Sachverhalte konnte auf den erzeugten Karten direkt überprüft werden.

Darüberhinaus ergab sich bei der Betrachtung der verschiedenen Klassen der durch die unterschiedlichen DNA-Motive gebundenen Proteine ein weiteres, äußerst interessantes Ergebnis: Ein großer Teil der Matrizen ordnet sich entsprechend der Bindungsdomänen der ihnen zugeordneten Transkriptionsfaktoren an. Diese Bindedomänen (siehe Kapitel 2) sind von der DNA völlig unabhängige dreidimensionale Strukturen. Dennoch scheinen bestimmte DNA Muster als Gruppe von Proteinen desselben Typs erkannt zu werden. Dieses Resultat ist nicht nur insofern interessant, als es die Wahl der Merkmalsvektoren für eine biologisch sinnvolle Anordnung von Matrizen unterstützt, sondern auch weil ein solcher gemeinsamer 'recognition code' bei den Proteinen zwar vermutet wird, aber bisher nur in Einzelfällen nachgewiesen werden konnte [40, 46, 65]. Die erzeugte Kohonenkarte ist also ein weiteres, überzeugendes Indiz für die Existenz diese Codes, da sie nicht Ähnlichkeiten in der Proteinstruktur, sondern lediglich die der Bindungsstellen auf der DNA betrachtet und trotzdem zur Anordnung der Proteine in ähnliche Gruppen führt.

Beim Vergleich unterschiedlicher Merkmalsvektoren ergaben sich aus der Kombination von Tri- und Tetranukleotiden und den daraus entstehenden 168-dimensionalen Merkmalsvektoren die besten Ergebnisse. Die aus den Dinukleotiden erzeugten achtdimensiononalen Vektoren lieferten für sich betrachtet die schlechtesten Ergebnisse und konnten auch in Kombination mit den anderen die Ergebnisse nicht verbessern.

Bei den Netzwerkparametern führten unterschiedliche Anfangsinitialisierun-

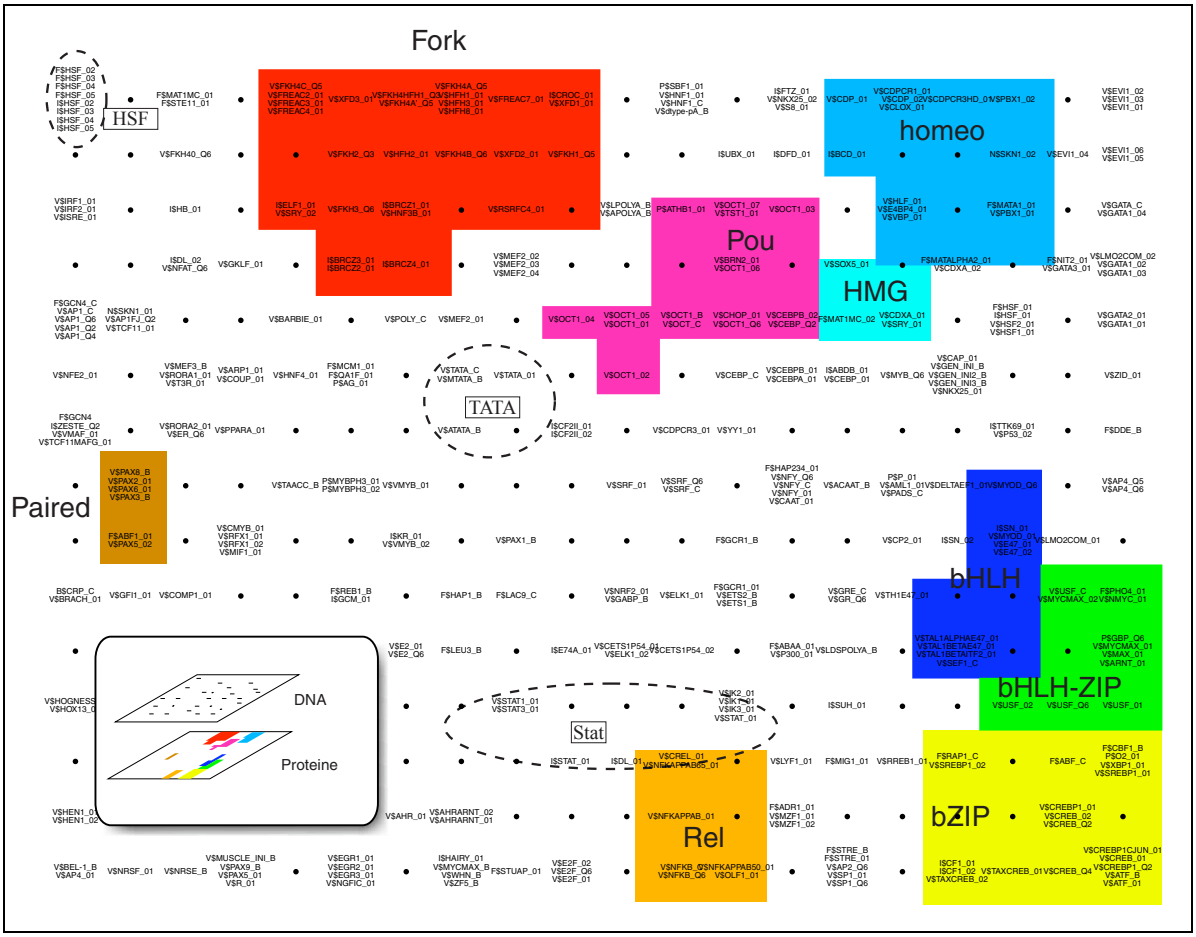

Abbildung 5.2: Kohonenkarte der Matrizen der MatInspector-library. Verschiedene biologische Aspekte der Anordnung werden dargestellt. Links oben und in der Mitte sind (mit gestrichelter Umrandung) zwei Gruppen von zueinandergehörenden Matrizen markiert. Im ersten Fall handelt es sich um die 'heat shock factor' Bindungsstellen in Insekten und Pilzen, im zweiten Fall um die zum 'core promoter' gehörende TATA-Box. Die im unteren Teil in der Mitte der Karte ebenfalls gestrichelt markierte Nachbarschaft der STAT3-Proteinbindungsstellen zu CREL bzw. NFκb ist hingegen weniger offensichtlich, aber dennoch biologisch begründet [3]. Die farbig unterlegten Gebiete stellen Gruppen von Proteinbindungsstellen dar, bei denen die Proteine denselben Typ der Bindedomäne besitzen (siehe Kapitel 2) und daher zur gleichen Protein-Klasse gerechnet werden können. Besonders deutlich wird die tatsächliche biologische Anordnung im unteren rechten Teil der Karte. Dort ist die Mischform der bHLH-ZIP Bindedomänen zwischen den „reinen“ bZIP und bHLH Proteinen zu finden. Das Fehlen einer Zuordnung in den übrigen Bereichen der Karte ist hauptsächlich im Mangel an Informationen bezüglich des Bindungsdomänentyps der gebundenen Proteine begründet. Die kleine eingesetzte Grafik unten links soll verdeutlichen, dass es sich hier um die Kombination zweier unterschiedlicher biologischer Bereiche handelt. Die Matrizen beschreiben die Bindungsstellen, bei denen es sich um DNA-Motive handelt. Die durch die Farben gekennzeichneten Bereiche stehen für unterschiedliche Proteintypen, die an diese DNA-Motive binden.

gen der Gewichtsvektoren im allgemeinen zu vergleichbaren Ergebnissen. Bei der Knotenanzahl stellte sich 300 (angeordnet in 15 Reihen von jeweils 20 Knoten) als guter Wert heraus: damit finden sich die meisten der sehr ähnlichen Matrizen auf einem Knoten, die Muster sind auf der Karte dennoch nicht allzu dicht angeordnet, so dass sie auch für die Verifizierung übersichtlich genug ist. Für die Lernparameter wurden die in [39] empfohlenen Werte verwendet.[5] Kleinere Abweichungen hiervon führten ebenfalls nur zu minimalen Veränderungen der erzeugten Karten.

Da des öfteren einige eigentlich zusammengehörende Gruppen am Rand der Karte auffällig auseinandergezogen waren, wurde ein Programm zur Erzeugung einer toroidalen Kohonenkarte (siehe Abschnitt 3.2) entwickelt. Für dieselben Merkmalsvektoren ergaben sich dabei für die Gruppen am Rand in einigen Fällen bessere Ergebnisse, allerdings wurden, analog zu der in Abschnitt 3.2.2 beschriebenen Problematik, andere vorher in einem Cluster enthaltene Matrizen nun nicht mehr zusammengruppiert. Um einen Vergleich zu ermöglichen, wurden 10 verschiedene toroidale Karten mit 10 herkömmlichen Karten[6] durch die Anwendung des in (3.11) definierten Maßes verglichen.[7] Dabei schnitten in sieben Fällen die herkömmlichen Karten geringfügig besser ab, die drei anderen Fälle konnten die toroidalen Karten ebenfalls mit minimalen Unterschieden für sich entscheiden.

Die Wahl der Kartenarchitektur hat also offensichtlich keinen allzu großen Einfluss auf die Anordnung. Die Tatsache, dass sich einzelne Matrizen immer wieder an anderen Stellen der Karte wiederfinden, lässt sich daher wohl eher auf die Beschaffenheit der Matrix zurückführen, als auf das Verfahren. Insgesamt ist mit der Erzeugung der Profile aus den Polynukleotidwahrscheinlichkeiten die Anforderung, ein Ähnlichkeitsmaß für den Vergleich der Matrizen untereinander zu finden, erfüllt. Dabei ist die gefundene Lösung unabhängig von der Länge der Matrizen und benötigt außer den Matrixeinträgen selbst keine weiteren Daten. Dennoch ist damit eine biologisch sinnvolle Anordnung erreicht, so dass auch ein Vergleich von Folgen von Bindungsstellen, in denen für dieselbe Bindungsstelle unterschiedliche Matrizen stehen, ermöglicht wird.

[5] Für die Vorordnung: $T = 10.000, \alpha_0 = 0,2, r_0 = 8$; für die Feineinstellung: $T = 80.000, \alpha_0 = 0,02, r_0 = 3$.

[6] Dabei wurden 5 Anfangsinitialisierungen zufällig erzeugt und anschließend mit dem jeweiligen Verfahren mit zwei unterschiedlichen Parametersätzen trainiert.

[7] Es wurden jeweils $C_{\text{gesamt},2,8}$ und $C_{\text{gesamt},3,8}$ berechnet. Die erhaltenen Werte lagen zwischen $0,66$ und $0,81$.

5.1.2 Promotorgruppierung

Der nächste Schritt bestand darin, in den Nukleotidsequenzen der Promotoren nach Transkriptionsfaktorbindungsstellen zu suchen, und diese dann in Merkmalsvektoren für die Gruppierung umzuwandeln. Als Testmenge wurden dabei Promotorsequenzen aus der 'Eukaryotic Promoter Database' verwendet. [54] Da die darin enthaltenen Nukleotidsequenzen von sehr unterschiedlicher Länge sind, wurde jeweils ein Bereich von 450 Basen vor dem vermuteten Transkriptionsstart bis 50 Basen danach ausgewählt.[8] Diese Sequenzen wurden dann mit Hilfe der in Abschnitt 2.2.2 vorgestellten MatInspector Software[9] nach Bindungsstellen durchsucht.

Die aus den MatInspector-Analysen stammenden Ergebnisse wurden analog zu der im vorangegangenen Abschnitt beschriebenen Methodik in Folgen von Merkmalsvektoren umgewandelt. Jeder gefundenen Matrix wurde der ihr entsprechende 168-dimensionale Vektor zugeordnet. Zusätzlich wurde die Strangorientierung der Matrix sowie die 'core' und 'matrix similarity scores' des MatInspector-Ergebnisses mit berücksichtigt[10], so dass jede gefundene Matrix durch einen Vektor aus insgesamt 171 Merkmalen repräsentiert wurde. Diese Sequenzen dienten dann als Trainingsdaten für die Erstellung einer SARDNET-Karte. Der in Abschnitt 4.2.2 beschriebene Algorithmus musste dazu mangels verfügbarer Programme [47] neu implementiert werden.

Nach Abschluss des Trainings[11] können die in den Promotorsequenzen gefundenen Bindungsstellen-Folgen auf der Karte visualisiert werden. So wird jeder Promotor durch eine Visualisierung auf der SARDNET-Karte repräsentiert. So wie bei der Anordnung der Bindungsstellen im vorangegangenen Abschnitt (siehe Abb. 5.2) können den verschiedenen Bindungsstellenmatrizen auch bei der Verwendung des SARDNET-Algorithmus je nach Typ einzelne Bereiche auf der Karte zugeordnet werden. Wie in Abbildung 5.3 zu sehen ist,

[8] Unbekannte Nukleotide werden in der EPD einfach durch `N`s ersetzt. Die Auswahl eines größeren Fensters hätte in etwa 60% der Fälle lediglich dazu geführt, dass die Sequenzen vorne und hinten durch diese Füllzeichen ohne jeden Informationsgehalt verlängert worden wären.

[9] Verwendet wurde die Version 4.3.

[10] Bei der Strangorientierung wurde der Wert auf 1 gesetzt, wenn es sich um eine Matrix auf dem Strang handelte; wurde sie auf dem Gegenstrang gefunden, wurde stattdessen -1 verwendet. 'core' und 'matrix similarity score' wurden einfach aus dem Matinspector-Ergebnis übernommen.

[11] Das Training erfolgt dabei analog zu den herkömmlichen Kohonenkarten, mit dem Unterschied, dass während eines Lernzyklus nicht ein einzelner, sondern immer gleich eine ganze Sequenz von Merkmalsvektoren präsentiert wird.

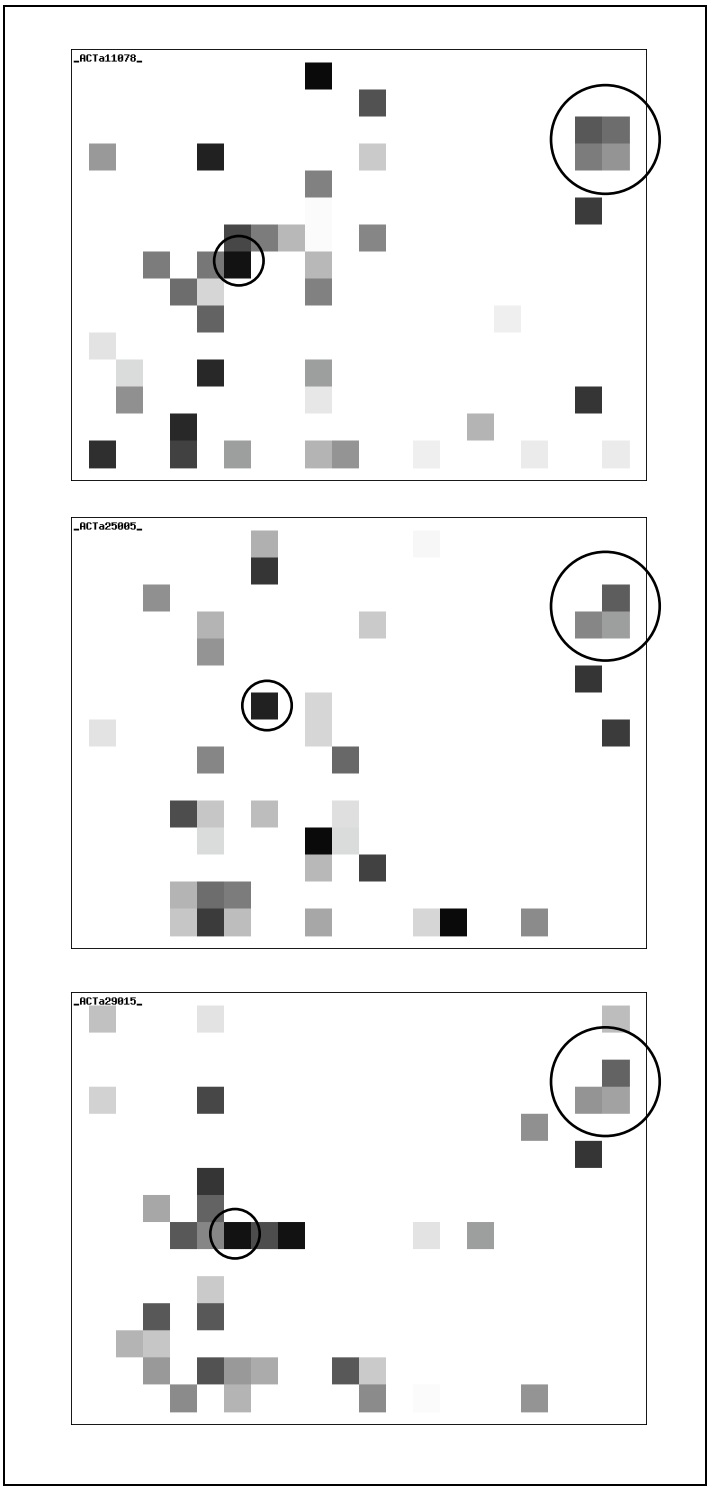

Abbildung 5.3: Visualisierung dreier α-Aktin-Promotoren (EP11078, EP25005 und EP29015 [54]) mit der SARDNET-Methode. Obwohl keine der Karten mit einer der anderen vollständig übereinstimmt, sind einzelne Knoten mit derselben oder vergleichbarer Aktivierung auf allen dreien an den gleichen oder benachbarten Positionen zu finden. Markiert sind hier die den SRF-Matrizen (großer Kreis rechts) und die der TATA-Box zugeordneten Knoten (kleinerer Kreis), die auch im Promotor-Modell auftauchen.

werden dieselben Matrizen immer durch identische oder benachbarte Knoten auf der Karte repräsentiert. Die Übereinstimmungen auf den einzelnen Visualisierungen stellen also Übereinstimmungen innerhalb der zugehörigen Promotorsequenzen dar. Dabei sind sowohl die Komponente der Position auf der Karte, die für die Art der Bindungsstelle steht, wie auch die am Ende der Visualisierung verbleibende Aktivierungsenergie der jeweiligen Gewinner-Knoten, die die Position der Bindungsstelle repräsentiert, zu berücksichtigen.

Bei den in Abbildung 5.3 dargestellten Karten handelt es sich um Visualisierungen dreier α-Aktin-Promotoren, für die, wie in Kapitel 2 bereits beschrieben, schon Promotor-Modelle aus verschiedenen Trankriptionsfaktoren bekannt sind ($\mathcal{M}_{Aktin}$ = {(USF,CAAT,SRF,TATA,INI,SP1), (18,31,31,20,7), (161,99,193,49,90)} für normale, bzw. $\mathcal{M}_{MuskelAktin}$ = {(USF,CAAT,SRF,SRF, TATA,INI,SP1), (24,34,31,18,20,0), (218,100,74,167,49,90)} für in Muskelgewebe vorkommende Aktin-Gene) [20]. Vergleicht man die Anzahl der mit dem MatInspector gefundenen Treffer innerhalb einer Promotorsequenz mit der der im Modell gefundenen Bindungsstellen, so wird diese um ein Vielfaches übertroffen. Das Modell stellt also lediglich eine gemeinsame Teilmenge dieser Bindungsstellen dar, die sowohl eine genügend hohe Sensitivität wie auch Spezifität aufweist, um Aktin-Promotoren möglichst exakt finden zu können. An der Ausbildung des Transkriptionskomplexes können durchaus auch im Modell nicht berücksichtigte Bindungstellen beteiligt sein. Für die Gruppierung der Promotoren in einzelne Klassen ist das Auffinden spezifischer Modelle, die nicht unmittelbar biologisch begründet sind, jedoch zunächst vollkommen ausreichend. Es stellt sich daher die Frage, ob sich für die Gruppierung der Promotoren mit Hilfe der SARDNET-Karte Visualisierungen finden lassen, die für einzelne Promotor-Klassen spezifisch sind. Wenn die zu vergleichenden Promotoren gemeinsame Modelle enthalten, sind diese auch auf den Visualisierungen zu finden. Zusätzlich sind auf den Karten allerdings noch andere Treffer enthalten, die im Modell nicht berücksichtigt werden. Wie aus Abbildung 5.4 hervorgeht, lassen sich in Teilbereichen der Karten dennoch spezifische Gemeinsamkeiten ausmachen.

Es ist deutlich zu sehen, dass bei verschiedenen Promotoren unterschiedliche Bereiche der Karte besetzt sind. Bei funktionell verwandten Promotoren bleiben bestimmte Kombinationen von Knoten mit ähnlicher Aktivierung jedoch erhalten. Modelle für Promotoren sind sowohl für die Aktine, wie auch die Histone und die LTRs[12] bereits bekannt [18, 20], so dass hier über die funk-

[12]'long terminal repeats': retrovirale Promotor-ähnliche Strukturen, die im Genom von

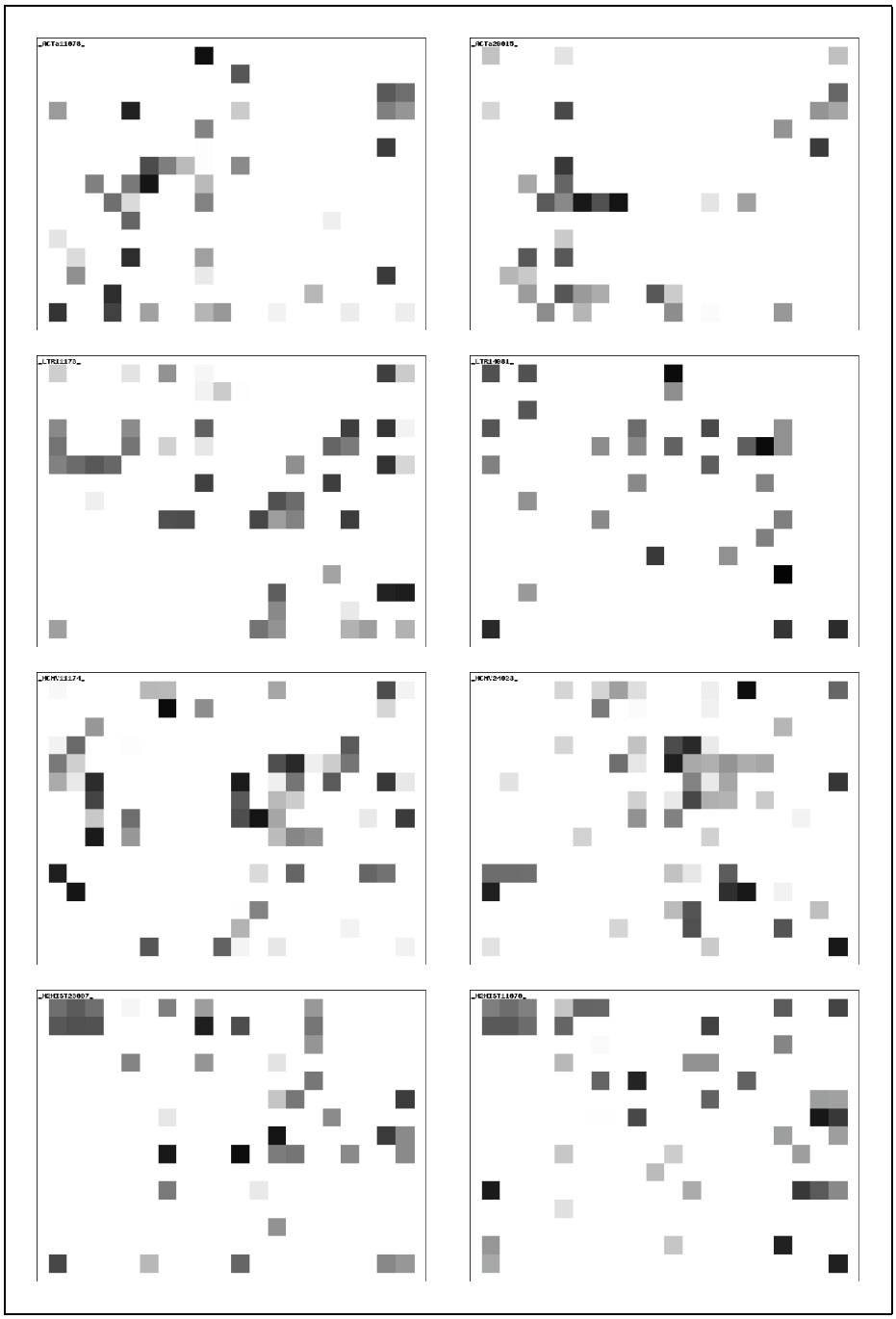

Abbildung 5.4: Vergleich ähnlicher Promotoren aus der EPD [54]. Es handelt sich, von oben nach unten, um jeweils zwei Aktin-Promotoren (EP11078 und EP29015), zwei LTRs (EP11173 und EP14081), zwei HCMVs (EP11174 und EP24023) und zwei Histon-Promotoren (EP23007 und EP11070). Wie man sieht, ergeben sich für die unterschiedlichen Gruppen jeweils verschiedene Visualisierungen, die untereinander in bestimmten Bereichen jedoch Ähnlichkeiten aufweisen.

tionale Ähnlichkeit der Gene hinaus auch die Gemeinsamkeiten zwischen den Promotoren belegt sind. Für die HCMVs[13] existiert bisher noch kein funktionell verifiziertes Promotor-Modell.

Ein bedeutsamer Faktor bei der Erstellung der SARDNET-Karten ist die Parameterwahl für die MatInspector-Analysen. Hier können zwei Schwellwerte für die 'matrix similarity' (d. h. die Ähnlichkeit des betrachteten Sequenzstücks mit der gesamten Matrix) und die 'core similarity' (für den Vergleich des am besten konservierten Teils der Matrix mit der Sequenz) angegeben werden. Beide Werte haben entscheidenden Einfluss sowohl auf die Qualität als auch die Quantität der Ergebnisse. Werden sie zu niedrig angesetzt, steigt die Anzahl der falsch positiven Treffer stark an; sind die Werte zu hoch, werden unter Umständen potentielle Bindungsstellen vom Programm übergangen. Abgesehen von der Möglichkeit, denselben Wert für die 'matrix similarity' aller zu suchenden Matrizen anzugeben, bietet MatInspector auch die Möglichkeit, individuelle Werte für die Matrizen zu verwenden, um wahlweise die Anzahl der falsch positiven bzw. negativen Treffer zu minimieren.

Die Wahl der Parameter wirkt sich insbesondere für eine Reihe von Matrizen mit relativ niedriger Spezifität aus, die ohne die Verwendung individueller 'matrix scores' unverhältnismäßig oft in vielen Promotoren gefunden werden. Abgesehen davon, dass diese Treffer in Hinsicht auf ihre biologische Funktionalität kritisch zu betrachten sind, stören sie in erheblichem Maß die Merkmalsgenerierung für die Promotor-Visualisierung. Statistisch gesehen tragen sie durch die Tatsache, dass sie unter Umständen in fast jeder Sequenz gefunden werden, nichts zur Unterscheidbarkeit der Promotoren bei. Zusätzlich wirkt sich die Besonderheit des SARDNET-Algorithmus, einmal aktivierte Knoten vom Lernen auszuschließen, hier negativ aus. Für öfter vorkommende Matrizen spezialisiert sich ja nicht nur ein einzelner Knoten, sondern ein ganzer Bereich der Karte. Handelt es sich dabei um Bindungsstellen die in einem Großteil der Promotoren als falsch positive Treffer gefunden werden, reduziert sich die Anzahl der Knoten, die für echte Treffer verbleiben, so dass sich unter Umständen unterschiedliche Bindungsstellen einen Knoten oder Bereich teilen müssen. Dadurch können für die Charakteristik des Promotors maßgebliche Bindungsstellen eventuell nicht richtig erkannt werden.

Für unsere Zwecke erwies sich die Auswahl einer minimalen Anzahl falsch po-

Vertebraten zu finden sind.

[13] 'human cytomegalovirus': hierbei handelt es sich ebenfalls um virale Promotoren.

sitiver Treffer als am besten geeignet. Der Nachteil, dass einzelne Bindestellen dann eventuell nicht gefunden werden, wird durch die deutliche Reduzierung an Gesamttreffern, insbesondere bei Matrizen mit niedriger Spezifität, mehr als aufgewogen. In den Fällen, bei denen Modelle für die Promotoren bekannt waren, gingen nur etwa in 10-25% der Fälle[14] im Modell verwendete Matrizen verloren. Andererseits konnte die Anzahl der Gesamttreffer um bis zu 40 % reduziert werden, so dass nur noch die für den jeweiligen Promotor charakteristischen Bindungsstellen übrig blieben und dementsprechend die Repräsentation durch die Visualisierungen exakter wurde.

Von den zur Verfügung stehenden Promotoren aus der EPD [54] wurden nun einige Gruppen ausgewählt, bei denen wegen der ähnlichen oder übereinstimmenden Funktionalität der Gene auf Gemeinsamkeiten innerhalb der Promotoren geschlossen werden konnte. Insgesamt wurden 139 Promotoren aus verschiedenen Bereichen extrahiert, für die sich in der EPD jeweils eine größere Anzahl an Exemplaren finden ließ. Verwendet wurden Promotoren für Histone, Aktine, Myosine, Collagene, Aldolasen, Wachstumsfaktoren, Interleukine, 'herpes simplex' Viren (HSVs), 'human cytomegalo' Viren (HCMVs) und 'long terminal repeats' (LTRs). Bei den drei letzteren handelt es sich um in Vertebratengenome eingefügte virale Promotoren. Insgesamt wurden also Gene für Strukturproteine (wie etwa die Aktine) ebenso verwendet wie Botenstoffe (Interleukin) oder auch retrovirale Promotorstrukturen (LTRs), um bei der Auswahl der Testdaten eine möglichst große Bandbreite zu berücksichtigen. Jedoch ist bisher lediglich für die Histone, die Aktine und die LTRs bereits bekannt, dass ihre Promotoren jeweils gemeinsame Strukturen beinhalten.

Diese 139 Promotoren wurden, wie oben beschrieben, mit dem MatInspector auf potentielle Bindungsstellen untersucht. Die erhaltenen Ergebnisse dienten dann als Trainingsdaten für eine SARDNET-Karte mit insgesamt 300 Knoten (15 Reihen mit jeweils 20 Knoten). Anschließend wurden die gefundenen Bindungsstellen für jeden Promotor auf dieser Karte visualisiert, und unter Anwendung von (4.7) wiederum in Merkmalsvektoren verwandelt, wobei als Fensterbreite $b = 3$ gewählt wurde.[15] Auf diese Weise ergab sich für jeden Promotor ein 234-dimensionaler Merkmalsvektor. Mit Hilfe dieser Vektoren wurde dann abschließend eine herkömmliche Kohonenkarte trainiert. Die

[14] Dieser Wert hängt von der 'core similarity' ab, die in diesem Fall 0.80 betrug.

[15] Größere Werte für b führen in dicht mit Gewinner-Knoten besetzten Bereichen der Karte zu einer unerwünschten Nivellierung der Merkmale. Mit $b = 2$ sind die betrachteten Fenster oft zu klein, um Ähnlichkeiten zu erfassen.

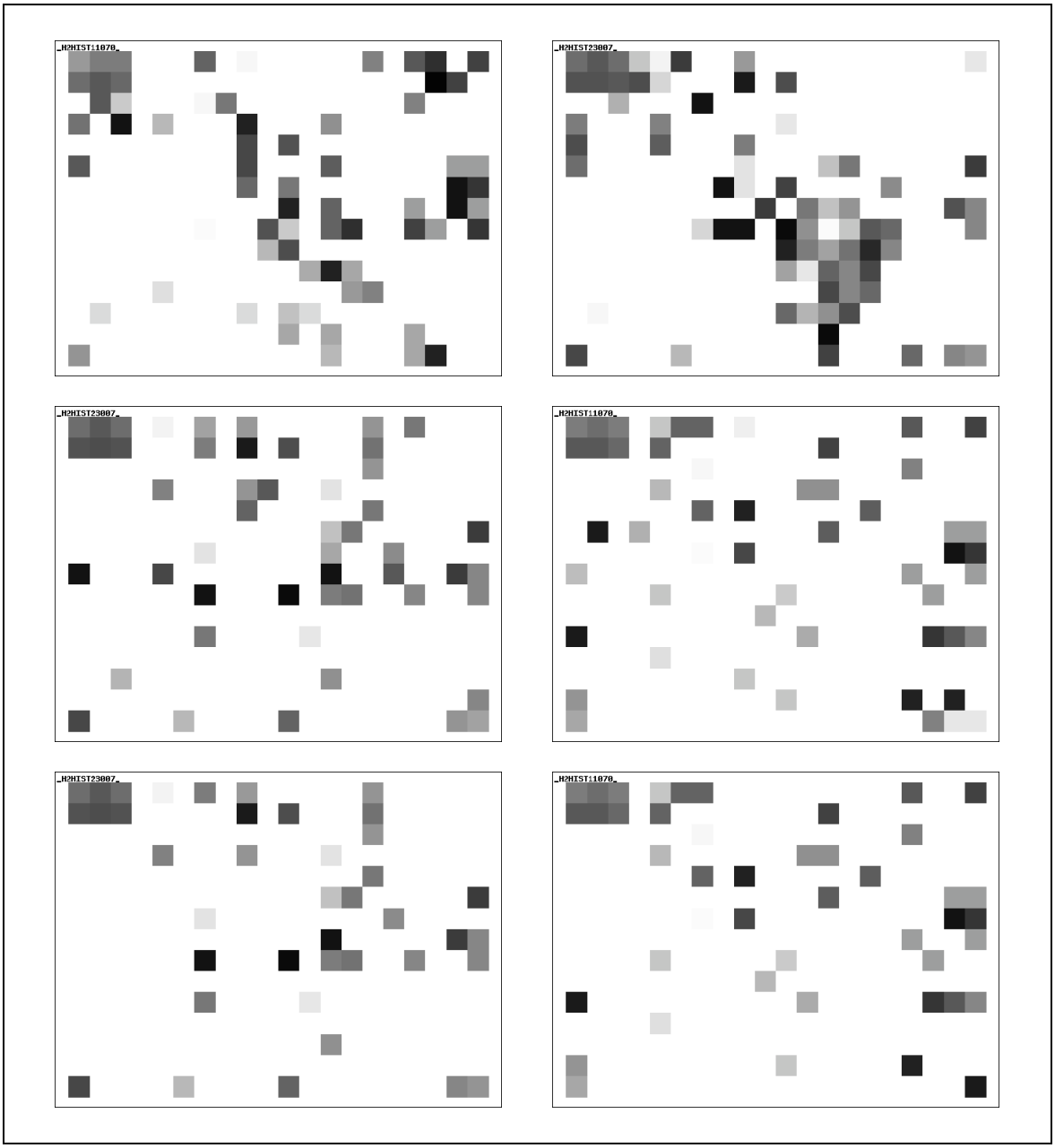

Abbildung 5.5: Die Auswirkungen unterschiedlicher MatInspector Parameter auf die Visualisierung zweier Histon-Promotoren (links:EP11070, rechts: EP23007). Oben wurden als 'matrix similarity' ein Wert von 0.80, als 'core similarity' ein Wert von 0.88 verwendet. In der Mitte wurden für die 'matrix similarity' individuelle Werte für die einzelnen Matrizen verwendet, um die Anzahl der falsch positiven Treffer zu minimieren. Dabei wurde eine core similarity von 0.75 gewählt. Für die beiden untersten Visualisierungen wurde der 'core similarity' Wert nochmals auf 0.80 erhöht. Wie man sieht nimmt die Gesamt-Anzahl der Treffer bei der Verwendung der individuellen 'matrix similarities' erheblich ab (bei diesem Vergleich etwa 20-35 %), die nochmalige Erhöhung der 'core similarity' reduziert die Anzahl nochmals um 10-20 %.

Parameter entsprachen wieder den Vorgaben von Kohonen [39], allerdings wurden entsprechend der geringeren Anzahl von Merkmalsvektoren nur 200 Knoten verwendet. Auch hier ergaben sich, unabhängig von der Anfangsinitalisierung mit wenigen Ausnahmen immer wieder dieselben Gruppierungen auf den Karten. In Abbildung 5.6 ist ein Ergebnis dargestellt.

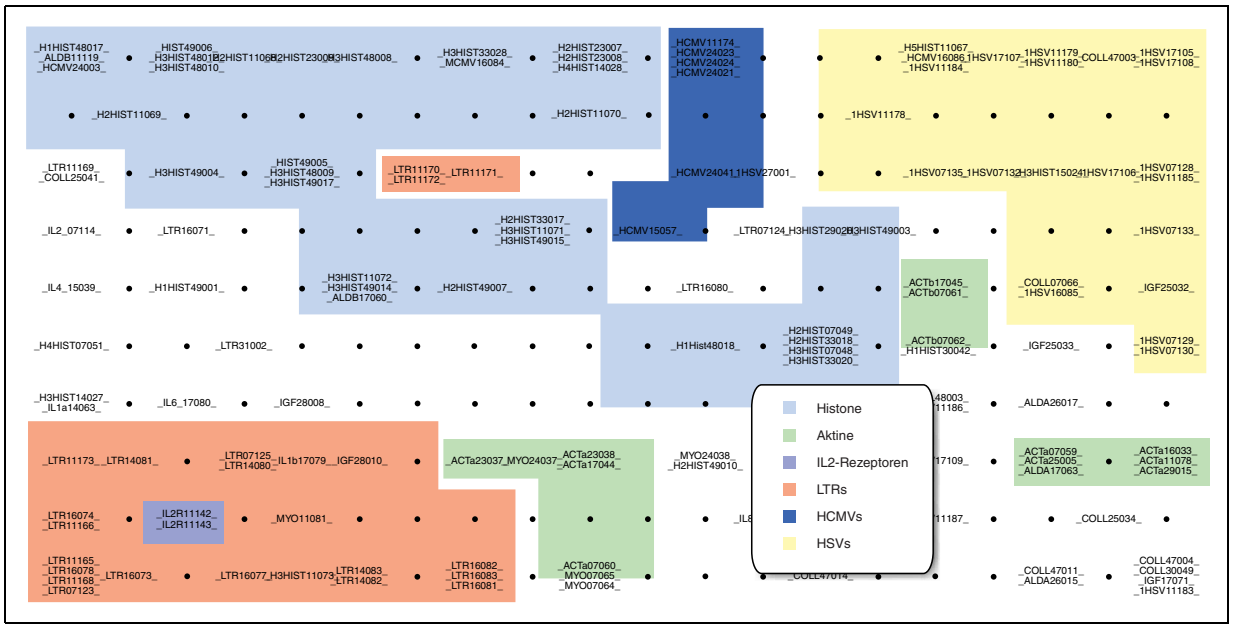

Abbildung 5.6: Kohonenkarte einiger ausgewählter Promotoren. Ingesamt wurden 139 Promotoren auf der Karte angeordnet. Die unterschiedlichen Klassen beinhalten Histone, Aktine, Collagene, Aldolasen, Interleukine und Interleukinrezeptoren, LTRs, HCMVs, HSVs, Myosine und Wachstumsfaktoren. Wie man sieht werden nicht alle Klassen korrekt erfasst. Die Promotoren, für die Modelle existieren (Aktine, LTRs, Histone), finden sich jedoch, von einzelnen Exemplaren abgesehen, als Gruppen auf der Karte wieder. Zusätzlich zu diesen lassen sich auch bei den HSVs, HCMVs und den Interleukin-Rezeptoren Cluster identifizieren. Die anderen Beispiele sind über die ganze Karte verteilt zu finden, allerdings sind hier auch noch keine Gemeinsamkeiten zwischen den Promotoren bekannt.

Für sechs der der elf Promotorgruppen (LTRs, Histone, HCMVs, HSVs, Aktine und IL2-Rezeptoren) ergeben sich dabei Häufungen in einzelnen Bereichen der Karte. Darunter sind auch die drei Promotorklassen (LTRs, Histone, Aktine), für die die strukturelle Ähnlichkeit bereits bekannt ist. Die Aktine sind in Abbildung 5.6 zwar auseinandergerissen, bilden aber entsprechend ihres unterschiedlichen Typs (α- und β-Aktine) von den anderen Klassen separierte Gruppen auf der Karte. Bei der Verteilung der übrigen Promotoren über die gesamte Karte können unterschiedliche Faktoren eine Rolle spielen.

Zum einen müssen funktionell ähnlichen Gene nicht zwangsläufig auch ähnliche Promotoren voranstehen. Zum anderen sind auch nicht alle Transkriptionsfaktoren bzw. deren Bindungsstellen bekannt; möglicherweise fehlen für einige der Promotoren genau die Motive, die hier für die Ausprägung charakteristischer Merkmale sorgen würden. Die trotz der Verkürzung auf 500 Basenpaare immer noch auftretenden Längenunterschiede bei den Sequenzen können ebenfalls eine Rolle spielen. Bei den in Abbildung 5.6 von der Hauptgruppe abgeteilten LTRs (EP11070, EP11071, EP11072) handelt es sich beispielsweise um Sequenzen, die nur etwa 350 Basenpaare lang sind, so dass hier ein Teil der Promotorstruktur verloren geht.

Die Anwendbarkeit von *PromoterMap* für die Gruppierung regulatorischer Sequenzen ist damit für die momentan verfügbaren Daten nachgewiesen. Die Methode ist also in der Lage, von biologischen Strukturen ausgehend eine funktionale Anordnung der Promotoren vorzunehmen, für die abgesehen von den DNA-Sequenzen der Promotoren und den Matrix-Beschreibungen der Bindungsstellen keine weiteren biologischen Daten benötigt werden.

Geht man davon aus, dass sich sowohl die Datenlage bei den Transkriptionsfaktoren wie auch bei den Promotorsequenzen in den nächsten Jahren deutlich verbessern wird, kann davon ausgegangen werden, dass sich auf die Gruppierung auf der Karte noch erheblich verbessert. Insbesondere würde sich eine Verbesserung der Qualität bei der Transkriptionsfaktorsuche positiv auf das Ergebnis auswirken. *PromoterMap* ist jetzt schon in der Lage, aus den immer noch hoch-redundanten Bindungsstellenfolgen für einen Großteil der Promotoren die gemeinsamen Strukturen abzuleiten und dementsprechende Klassen zu finden. Eine Verminderung der Redundanz kann sich dabei nur positiv auf die Ergebnisse auswirken. Die Vervollständigung der Promotorsequenzen wird ebenfalls zur Verbesserung der Resultate beitragen. Teile der Promotorstruktur, die zur Zeit eventuell noch gar nicht in den Sequenzen enthalten sind, werden dann ebenfalls für den Vergleich zur Verfügung stehen. Die Ergebnisse werden sich also im Rahmen der Zunahme des allgemein verfügbaren biologischen Wissens stetig verbessern, ohne dass dafür grundsätzliche Veränderungen an der Methodik vorgenommen werden müssen. Bei einer Zunahme der Daten muss lediglich die Kartengröße in den einzelnen Schritten angepasst werden.

5.2 Zusätzliche Anwendungsmöglichkeiten der Einzelschichten

Zusätzlich zu den von *PromoterMap* erzielten Ergebnissen bei der Promotorgruppierung lassen sich zwei der in den einzelnen Teilschritten des Verfahrens erzielten Resultate generell nutzbringend bei der Promotoranalyse verwenden. Zum einen handelt es sich dabei um die Reduktion von Daten bei der Bindungsstellensuche, zum anderen um die einfache automatische Erstellung grundlegender Promotormodelle aus den SARDNET-Karten funktionell verwandter Promotoren. Diese beiden Anwendungen sollen im folgenden jeweils kurz erläutert werden.

5.2.1 Datenreduktion bei der Bindungsstellensuche

Bei der *in silico* Suche von Transkriptionsfaktorbindungsstellen mit Hilfe von Gewichtsmatrizen besteht, wie bereits in Abschnitt 4.2.1 erwähnt, ein Problem darin, dass oft für dieselbe Bindungsstelle mehrere Matrix-Modelle vorliegen, die das entsprechende DNA-Motiv beschreiben ohne jedoch vollständig übereinzustimmen. Dies kann dazu führen, dass bei der Analyse einer DNA-Sequenz an ein und derselben Bindungsstelle mehrere dieser Matrizen einen Treffer erzeugen. Dies resultiert in einer Vielzahl redundanter Ergebnisse. Um diese reduzieren zu können, wäre es wünschenswert, wenn von der Gruppe von Treffern, die an einer Bindungsstelle gefunden werden nur ein Repräsentant im Ergebnis verbliebe. Theoretisch könnte man von vornherein die Anzahl der verwendeten Matrizen verringern, indem für jede Bindungsstelle lediglich ein Exemplar verwendet würde. Allerdings ginge dadurch die Sensitivität des Verfahrens verloren, denn die Unterschiede der einzelnen Matrizen entsprechen der Variabilität der Bindungsstellen-Motive in der DNA. Für die *MatInspector professional* Software wurde daher eine andere Vorgehensweise gewählt. Zunächst wird nach allen Matrizen gesucht. Werden innerhalb eines vorgegebenen Abstandsintervalls mehrere Treffer ähnlicher Matrizen gefunden, wird als Ergebnis nur die am besten passende ausgegeben. Auf diese Weise kann eine Reduktion der Daten um den Faktor 2-4 erreicht werden, was die Weiterverwendung der Ergebnisse in vielen Fällen, etwa bei der Suche nach Bindungsstellen-Modulen [36], erst ermöglicht.

Bei der Entwicklung dieser *Familien* genannten Gruppen ähnlicher Matrizen stellte die Möglichkeit, diese durch die Anordnung auf einer Kohonen-Karte nahezu automatisch erstellen lassen zu können einen wichtigen Schritt dar. Auch wenn ein Abgleich anhand der zusätzlich zur Matrix bekannten biolo-

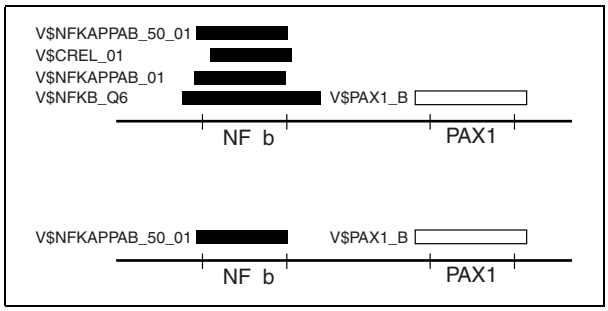

Abbildung 5.7: Ein Beispiel für die Vermeidung von Redundanz bei der Bindungsstellensuche. Für die Bindungsstelle des NFκB Proteins sind mehrere verschiedene Matrizen bekannt, für die PAX1 Bindestelle bisher nur eine. Wie im oberen Teil der Grafik zu sehen ist, passen an an der NFκB Bindestelle unter Umständen mehrere unterschiedliche Matrizen und würde bei diesem Beispiel würde man so insgesamt fünf Treffer erhalten. Wird lediglich nach dem besten Vertreter — in diesem Fall die Matrix V$NFKAPPAB_50_01 — aus einer Gruppe ähnlicher Matrizen gesucht, reduziert sich das Ergebnis auf zwei Treffer (siehe unterer Teil der Grafik).

gischen Daten der Bindungsstelle noch immer nötig ist, um eventuell fehlplazierte Matrizen richtig zuzuordnen, geben Kohonenkarten wie in Abbildung 5.2 eine verlässliche Gesamtübersicht, anhand derer die Familieneinteilung relativ schnell vorzunehmen ist. Die Zuordnung neuer, bisher unbekannter Matrizen lässt sich mit Hilfe einer einmal trainierten Kohonenkarte problemlos automatisieren, so dass der Zeitaufwand, der ansonsten für die Literaturrecherchen anfallen würde, um die entsprechende Matrix-Familie zu finden, auf ein Minimum reduziert wird. Lediglich bei der Analyse größerer Anzahlen neuer Matrizen kann es notwendig werden, die Karte neu anzulegen, um eventuell neu hinzugekommene Familien zu berücksichtigen.

Abgesehen von der Bestätigung, dass es sich bei der in Abschnitt 4.2.1 gefundenen Beschreibung der Matrizen durch Polynukleotidprofile um ein geeignetes Ähnlichkeitsmaß für die weitere Verwendung innerhalb des *PromoterMap*-Verfahrens handelt und dem Hinweis auf den generellen 'recognition code' der Bindeproteine, können die gefundenen Kohonenkarten also auch für eine automatische Gruppierung von Bindungsstellen-Matrizen verwendet werden. Mit Hilfe einer solchen Gruppierung lassen sich bei der Computer-gestützten Suche nach Bindungsstellen in DNA-Sequenzen erhebliche Verbesserungen bei der Quantität und damit letztendlich auch der Qualität der erhaltenen Ergebnisse erzielen.

5.2.2 Automatische Modellbildung

Zusätzliches Potential bietet der Vergleich der Visualisierungen der Bindungsstellen-Folgen innerhalb der Promotor-Sequenzen. Hier können auf relativ einfache Art und Weise aus den Gemeinsamkeiten funktionell verwandter Promotoren Modelle für die Analyse unbekannter DNA-Sequenzen generiert werden. Die Vorgehensweise entspricht dabei einem direkten Vergleich der einzelnen Knoten der verschiedenen Visualisierungen (siehe Abbildung 5.8).

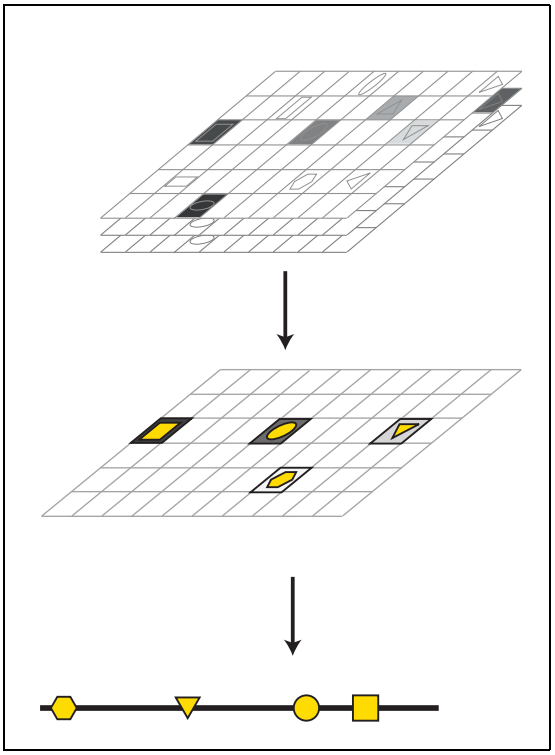

Abbildung 5.8: Schematische Darstellung des Konzepts der automatischen Promotor-Modell-Generierung. Die unterschiedlichen Visualisierungen werden „übereinandergelegt“, um so Gemeinsamkeiten zu erkennen. Findet sich ein Knoten mit vergleichbarer Aktivierung auf mehreren Karten, geht die entsprechende Bindungsstelle in das Modell ein.

Der Vergleich einzelner Knoten ist jedoch nicht ausreichend, da durch das unterschiedlich häufige Auftreten derselben Bindungsstelle innerhalb der verschiedenen Promotoren die Aktivierungsmuster auf der Karte variieren können. Für den Vergleich werden daher auch hier (analog zu (4.7)) Fenster betrachtet, die mehrere Neuronen enthalten (siehe Abbildung 5.9).
Im Gegensatz zu einem herkömmlichen Alignment der Bindungsstellen, bei

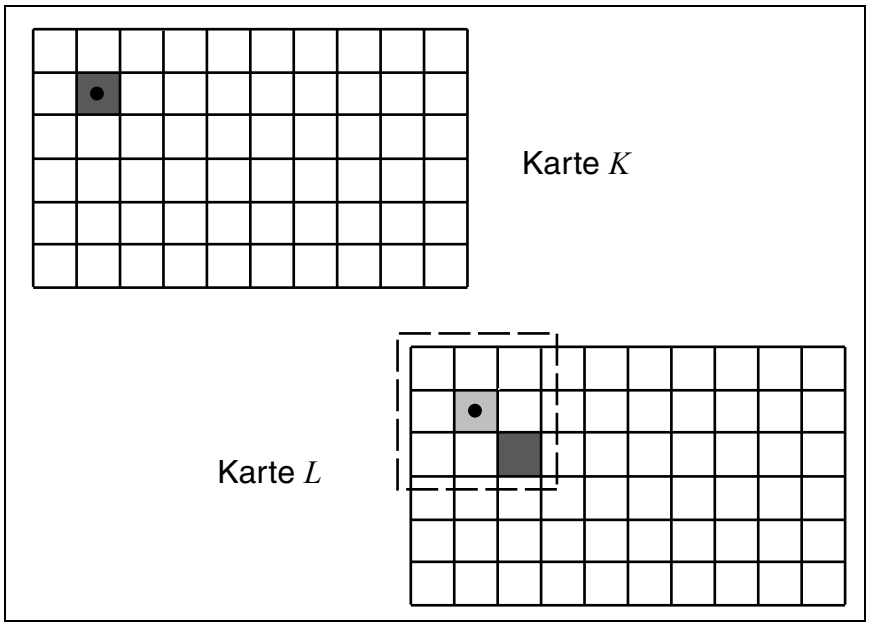

Abbildung 5.9: Vergleich zweier Karten K und L. Geprüft wird, ob zum markierten Knoten auf K ein ähnlicher Knoten auf L vorhanden ist. Die Aktivierungen aller Knoten auf L, die im (durch die gestrichelte Linie) markierten Fenster F liegen, werden mit denen des entsprechenden Fensters auf K verglichen. So wird, obwohl der Knoten auf L bereits eher aktiviert worden ist, der benachbarte Knoten gefunden, der der vorgegebenen Aktivierung entspricht.

dem positionsweise nach Ähnlichkeiten gesucht wird, findet der Vergleich hier auf der Basis des Bindungsstellentyps statt. Anstatt sequentiell nach Gemeinsamkeiten in einer Folge von Bindungsstellen zu suchen, wird hier zunächst für jeden Typ überprüft, ob das entsprechende Motiv innerhalb der Promotorsequenz gefunden wurde. Danach können anhand der Aktivierung die Positionen verglichen werden. Auch hier wird nicht auf exakte Gleichheit überprüft; zwei Aktivierungen werden als äquivalent betrachtet, wenn beide innerhalb eines vorgegebenen Intervalls I liegen.

Da die Anzahl der einzelnen Visualisierungen für jede Promotorklasse verhältnismäßig klein ist, und auch die Anzahl der Knoten mehrere hundert nicht übersteigt, kann der Vergleich der Visualisierungen als einfacher 'brute force'-Ansatz realisiert werden, der für jedes Fenster überprüft, ob die Anzahl der Knoten, deren Aktivierungen innerhalb von I liegen, einen ebenfalls vorzugebenden Grenzwert l überschreitet. Ist dies der Fall, wird die entsprechende Bindungsstelle in das Modell übernommen. Sind auf diese Weise alle Elemen-

te des Modells gefunden, lassen sich anhand der Differenzen der jeweiligen Aktivierungen die Abstände zwischen den Bindungsstellen innerhalb des Modells festlegen.

Für zwei der Promotorgruppen, für die schon Modelle bekannt waren, wurde überprüft, ob sich diese auch mit Hilfe der beschriebenen Methode finden lassen. Für das Aktinmodell ($\mathcal{M}_{Aktin}$ = {(USF, CAAT, SRF, TATA, INI, SP1), (18, 31, 31, 20, 7), (161, 99, 193, 49, 90)} [16] wurden sieben Sequenzen miteinander verglichen (EP25005, EP16033, EP11078, EP29015, EP17045, EP07061), der Wert l wurde dabei auf 4 gesetzt, die Intervallbreite von I wurde zu 0.1 gewählt[17] Als Resultat ergab sich $\mathcal{N}_{Aktin}$ = {(CAAT, SRF, TATA, INI, SP1), (40, 34, 21, 10), (95, 99, 49, 85)}. Die vorderste Bindungsstelle wurde also nicht gefunden und bei den Abständen ergaben sich kleine Abweichungen. Für die die Untergruppe der 'C-Type'-LTRs konnte ebenfalls ein Modell erstellt werden: $\mathcal{N}_{LTR}$ = {(XBBF, CAAT, TATA, MINI, RPOA), (53, 25, 13, 23), (102, 85, 42, 72)}[18] (basierend auf den Sequenzen EP01723, EP11165, EP11168, EP16073, EP16078). Das originale ModelInspector Modell verwendet für diese Promotoren anstelle von Matrizen sogenannte 'Consensus-Profile'[19] für die Beschreibung des Promotors [18]. Außerdem werden zusätzlich strukturelle Elemente wie 'hairpins'[20] im Modell verwendet. Von den insgesamt elf Elementen des Originalmodells stimmen CAAT, TATA und RPOA mit dem Modell $\mathcal{N}_{LTR}$ überein.

Zusätzlich wurde versucht, für die HCMVs (EP11174, EP24021, EP24024, EP24025) ein Modell zu finden. Hier konnte $\mathcal{N}_{HCMV}$ = {(CREB, CREB, EBOR, NFKB), (11, 24, 20), (98, 96, 98)} realisiert werden. Eine damit durchgeführte Suche innerhalb der EPD ergab bei insgesamt 1.386 durchsuchten Sequenzen (mit 831.600 Nukleotiden) nur einen zusätzlichen Treffer, bei dem es sich ebenfalls um einen viralen Promotor handelt. Beim Durchsuchen der 'human section' der EMBL-Datenbank[21] [75] ergaben sich insgesamt nur 606 Treffer in insgesamt 136.626 Sequenzen mit rund 1,4 Milliarden Nukleotiden.

[16] USF='upstream stimulating factor', CAAT=CAAT-Box, SRF='serum response factor', INI=Initiator, SP1='stimulating protein 1'

[17] Dies entspricht bei einer Gesamtsequenzlänge von 500 Basen einem 50 Nukleotide breiten Fenster.

[18] XBBF= 'x-box binding factor', RPOA = 'retroviral poly-A site'

[19] Hierbei handelt es sich ebenfalls um eine statistische Beschreibungen für Transkriptionsfaktor-Bindungsstellen.

[20] Mit 'hairpin' bezeichnet man Bereiche, in denen die DNA eine Schleife nach außen bildet. Die Ähnlichkeit dieser Verformung mit einer Haarklammer führt zur Namensgebung.

[21] Version 66

Das Modell ist damit zumindest hochspezifisch (etwa 1 Treffer pro 2.200.000 Nukleotide), allerdings handelt es sich bei den Sequenzen, in denen Treffer erzielt wurden, um bisher nicht vollständig annotierte Sequenzstücke, so dass eine Überprüfung inwieweit an den gefundenen Positionen tatsächlich HCMV-Gene bzw. Promotoren vorhanden sind, noch aussteht.

Das Verfahren ist also durchaus geeignet, spezifische Modelle für Gruppen von Promotoren zu finden. Es benötigt im Unterschied zu den herkömmlichen Vorgehensweisen keinerlei Informationen aus der Literatur. Außerdem werden aufgrund des Vergleichs der Visualisierungen auch Modelle aus Sequenzen extrahiert, die lediglich ähnliche, nicht exakt übereinstimmende Bindungsstellen enthalten. Die auf diesem Wege erhaltenen Ergebnisse sind im Vergleich mit den bekannten Modellen qualitativ unterlegen, jedoch lassen sich auf diese Weise ohne den Aufwand der Literatursuche einfach und schnell Grundgerüste für Modelle erstellen, die anschließend beispielsweise mit der ModelGenerator Software verfeinert werden können.

5.3 Zusammenfassung

Das Verfahren *PromoterMap* bietet zum ersten Mal die Möglichkeit, regulatorische Sequenzen anhand der in ihnen enthaltenen biologischen Strukturen in funktionaler Weise anzuordnen. Damit wurde eine Methode konstruiert, die in der Lage ist, ohne direktes biologisches Vorwissen über Promotoren diese einander zuzuordnen und somit eine Klassifizierung zu ermöglichen. Im Gegensatz zu den bisher bekannten Verfahren wurden bei der Entwicklung von *Promotormap* die biologischen Gegebenheiten konsequent berücksichtigt. Dementsprechend spiegelt der Aufbau der Methode die Abläufe bei der Promotererkennung in der Natur wider. Zunächst werden auf der Ebene der Nukleotidsequenz die Bindungsstellen analysiert. Dies entspricht bei der Bildung des Transkriptionskomplexes der Bindung der einzelnen Transkriptionsfaktoren. Die Kombination aus diesen führt dann schließlich zur Promotorerkennung bzw. -klassifizierung und bei der Genregulation zur Anlagerung der Polymerase II und schließlich zur Transkription des Gens. In dieser Hinsicht unterscheidet sich das Verfahren deutlich von den bisher bekannten Methoden, wie in Abbildung 5.10 nochmals verdeutlich wird.

Das Verfahren ist skalierbar und profitiert direkt von Verbesserungen der allgemein verfügbaren biologischen Daten. Eine Zunahme verfügbarer Daten wird also zu einer Qualitätssteigerung bei den Resultaten führen.

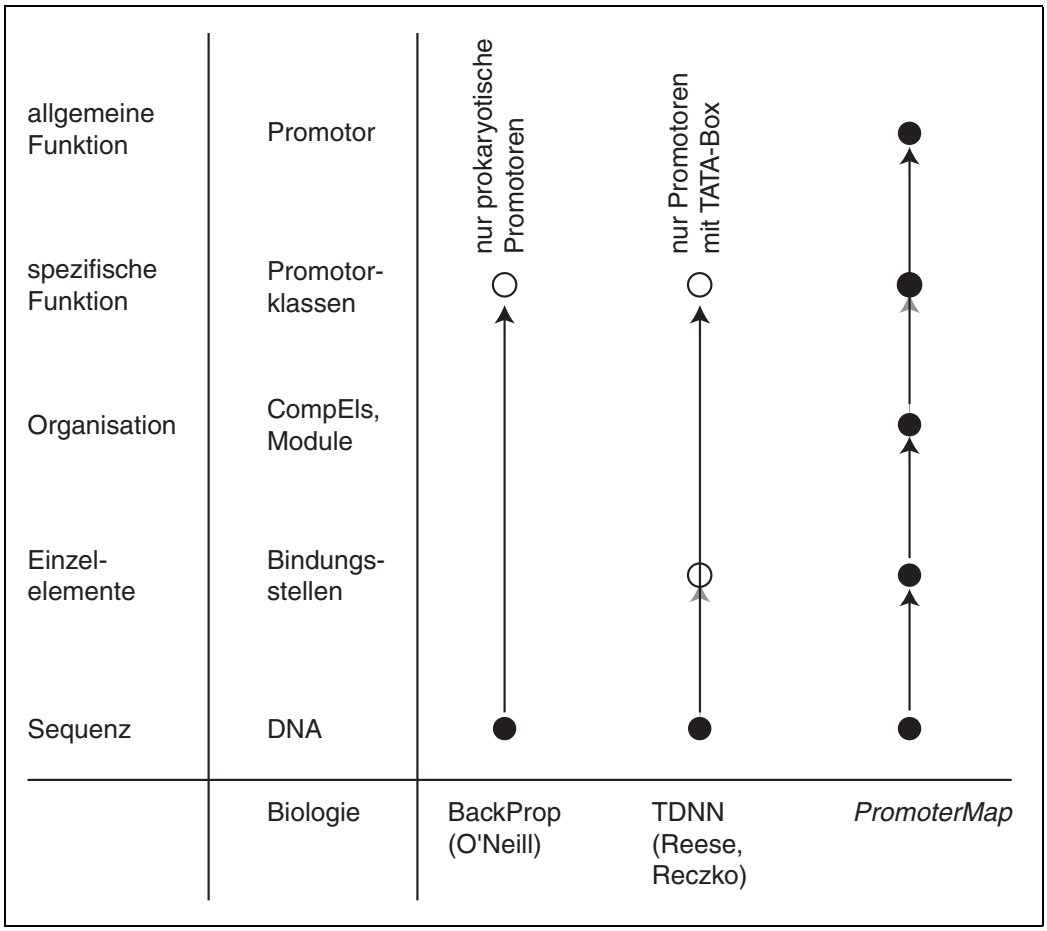

Abbildung 5.10: Vergleich der in Kapitel 3 vorgestellten Verfahren mit der *PromoterMap*-Methode hinsichtlich der Modellierung biologischer Gegebenheiten. Ein ausgefüllter Punkt bedeutet, dass die Erkennung der links aufgeführten Elemente funktioniert, ein leerer Punkt steht für eingeschränktes Erkennen. Das Verfahren von O'Neill [50] erkennt nur Muster in der DNA-Sequenz und ist daher nur in der Lage, eine ganz bestimmte Gruppe von Promotoren, nämlich die des Bakteriums Escherichia coli, zu erkennen. Von einer Promotor-Erkennung im eigentlichen Sinne kann also nicht geprochen werden. Die Verfahren von Reese und Reczko [44, 45] erkennen und benutzen schon bestimmte Bindungsstellen bzw. strukturelle Signale. Allerdings handelt es sich um eine enge Auswahl, so dass auch hier die Erkennung von Promotoren auf die diejenigen beschränkt ist, die genau diese beiden DNA-Muster beinhalten. *PromoterMap* verwendet die durch die Biologie vorgegebene Strukturierung, um so vollständige formale Promotor-Modelle vergleichen zu können.

Darüberhinaus ergeben sich aus den einzelnen Teilen der Methode interessante Ergebnisse. So lassen sich durch die Anordnung der die Transkriptionsfaktorbindungsstellen beschreibenden Matrizen unmittelbar Familien verwandter Matrizen ableiten, die zur erheblich Reduzierung redundanter Ergebnisse bei der Suche nach potentiellen Bindungsstellen führen. Außerdem bestärkt das Ergebnis, dass sich die Matrizen entsprechend der den Bindungsstellen zugehörigen Proteinklassen anordnen, die Vermutung, dass ein genereller 'recognition code' für Proteine und ihre Bindungsstellen existiert. Darüberhinaus können die Visualisierungen der in den einzelnen Promotoren gefundenen Bindungsstellenfolgen für die einfache und schnelle Konstruktion grundlegender Promotormodelle genutzt werden.

Kapitel 6

Schlußbetrachtung

In diesem Kapitel werden zunächst die wichtigsten Ergebnisse der Arbeit noch einmal zusammengefasst. Anschließend sollen einige offene Fragen zum Verfahren diskutiert werden.

6.1 Zusammenfassung

Die Motivation für die vorliegende Arbeit war das Problem der in silico Promotorerkennung bzw. -analyse in eukaryotischer DNA. Dabei sollte insbesondere ein Ansatz gefunden werden, der biologische Strukturen bereits beim Entwurf des Verfahrens berücksichtigt, darüberhinaus aber möglichst wenig Expertenwissen benötigt. Das Ziel war die automatische Klassifizierung von Promotorsequenzen in funktionell ähnliche Gruppen.

Bei der Fragestellung handelt es sich um ein Problem der Mustererkennung für dessen Lösung – aufgrund der bisher nur spärlich vorhandenen Daten über Promotor-Klassen – nur Methoden aus dem Bereich der nicht überwachten Lernverfahren in Frage kamen. Dementsprechend wurden zwei Methoden zur Selbstorganisation ausgewählt und in einem hierarchischen Ansatz zu dem *PromoterMap* getauften Verfahren kombiniert. Konkret wurde eine Variante von Kohonens Algorithmus mit dessen herkömmlicher Version verknüpft. Auf diese Weise konnten zunächst Folgen von Transkriptionsfaktorbindungsstellen visualisiert werden; mit Hilfe dieser Repräsentationen wurde dann eine Anordnung der Promotersequenzen erreicht. Außerdem eignen sich Teilschritte des Verfahrens als Einzelanwendungen für die Verbesserung bereits bestehender Software sowie zur automatischen Erzeugung grundlegender Promotormodelle verwendet werden.

Kapitelübersicht Im zweiten Kapitel standen die biologischen Grundlagen für die Promotoranalyse im Mittelpunkt. Insbesondere wurde auf den modularen Aufbau der Promotoren eingegangen, der die biologische Grundlage für diese Arbeit bildet. Um das Verständnis über die Herkunft verwendeter Daten zu erleichtern und den nötigen Aufwand zu verdeutlichen, wurden einige im Zusammenhang mit der Untersuchung von Promotoren wichtige molekularbiologische Labormethoden vorgestellt. Die Beschreibung von Softwaremethoden für das Auffinden struktureller Elemente und daraus konstruierten vorgegebenen Promotormodellen und ihrer theoretischen Grundlagen bildete den Schluss dieses Abschnitts.

Den Schwerpunkt des dritten Kapitels bildete die Theorie der im Rahmen der Arbeit interessanten Klassifikationsverfahren. Dabei wurden sowohl die Grundlagen für bereits existierende Verfahren als auch die für die neue Methode benötigten Algorithmen erläutert. Von grundsätzlichem Interesse für unser Problem waren dabei Verfahren, die in der Lage sind, Folgen von Einzelmustern zu klassifizieren. Außerdem wurden Architektur und Ergebnisse dreier Verfahren für die Promotorerkennung vorgestellt, die auf vorher beschriebenen überwachten Algorithmen basieren.

Im vierten Kapitel wurden zunächst die existierenden Verfahren im Hinblick auf die Verwendung biologischen Wissens analysiert und dabei festgestellt, dass die in Promotoren vorkommenden Strukturen nur ansatzweise berücksichtigt werden. Davon ausgehend wurde dann die Architektur des neuen Verfahrens *Promotormap* entwickelt. Im Gegensatz zu den anderen Methoden wurden hier nicht überwachte Algorithmen verwendet, da die aktuelle Datenlage ein effizientes Training etwa von Backprop-Netzen nicht ermöglicht. Dabei waren drei Hauptprobleme zu lösen:

- Für die den Transkriptionsfaktorbindungsstellen zugeordneten Matrizen musste eine Möglichkeit gefunden werden, adäquate Merkmale zu generieren, um so die vorher nicht grundsätzlich bekannten Ähnlichkeiten zwischen den einzelnen Motiven zu erfassen.

- Anstelle eines Verfahrens zur Klassifizierung von Einzelmustern wurde eine Methode für die Erfassung von Mustersequenzen benötigt, die darüberhinaus in der Lage ist, nicht nur die Reihenfolge, sondern auch den Abstand zwischen einzelnen Mustern zu konservieren.

- Die so gefundenen Visualisierungen jeder einzelnen Folge von Mustern mussten für die weitere Gruppierung ebenfalls durch einen geeigneten

Merkmalsvektor beschreibbar sein. Da Strukturähnlichkeiten bei den Promotoren sich in vergleichbaren Teilbereichen der generierten Karten widerspiegeln, ist es besonders wichtig, die Merkmale so zu wählen, dass diese Eigenschaft nicht verloren geht.

Die für die Charakterisierung der Matrizen gefundene Beschreibung stellt eines der wichtigsten Ergebnisse der Arbeit dar. Lediglich aufgrund der statistischen Beschreibung der Bindungsstellen konnte hier ein biologisch relevanter Merkmalsvektor gefunden werden, der eine funktionell sinnvolle Anordnung der Matrizen erlaubt. Auf diese Weise lassen sich auch bisher unbekannte Bindungsstellen bereits etablierten Motiven zuordnen, ohne dass weiteres Wissen beispielsweise über ihre Struktur oder biochemische Eigenschaften benötigt wird.
Das SARDNET-Verfahren [31] stellt für die Umsetzung der in den Promotoren gefundenen Sequenzen von Transkriptionsfaktorbindungsstellen in vergleichbare Visualisierungen ein geeignetes Verfahren dar. Einerseits werden die Ähnlichkeiten zwischen den einzelnen DNA-Motiven berücksichtigt, andererseits lassen sich, durch minimale Modifikation des Algorithmus, auch die Positionen der Bindungsstellen innerhalb des Promotors berücksichtigen. Die Tatsache, dass während des Lernens den öfter auftretenden Mustern mehr Platz eingeräumt wird, als den seltener vorkommenden, kann insgesamt ebenfalls als Vorteil des Verfahrens betrachtet werden.
Die Aufteilung der im vorangeganenen Schritt erhaltenen Karten in überlappende Fenster liefert dann die Merkmalsvektoren für die Gruppierung der untersuchten Promotoren. Dabei entspricht diese Vorgehensweise genau der Anforderung, insbesondere lokale Teilbereiche der Karte vergleichbar zu machen. Die für die Kohonenkarten wichtige Eigenschaft der Nachbarschaftserhaltung wird dadurch ebenfalls in höherem Maße beibehalten, als dies beispielweise durch direkte Erfassung der Neuronen der SARDNET-Karte durch einen Merkmalsvektor möglich wäre.
Das Verfahren *PromoterMap* besteht damit aus der hierarchischen Schichtung zweier selbstorganisierender Karten. Es basiert, im Vergleich zu anderen Verfahren, nicht auf der Analyse der zugrundeliegenden DNA, sondern auf dem Vergleich darin enthaltener Transkriptionsfaktorbindungsstellen. Damit ist es von der Konstruktion her näher an der biologischen Realität als Verfahren, die auf der direkten Untersuchung der Nukleotid-Sequenzen basieren, da für die molekularbiologischen Mechansimen am Promotor letztendlich diese Motive verantwortlich sind. Ein weiterer Vorteil des Verfahrens ist die „ungenaue“ Betrachtung dieser Bindungsstellen. So wie in der Natur innerhalb des Kontextes eines Promotors Bindungsstellenmotive durch ihnen ähnliche Muster, die eigentlich für die Erkennung eines anderen Proteins geeigneter

wären, ersetzt werden können, ohne dass die Funktionalität dadurch beeinträchtigt ist, werden auch hier ähnliche Matrizen durch die Anordnung auf der Karte in funktionelle Gruppen zusammengefasst. Anstelle des oft ergebnislosen Vergleichs von Folgen individueller Muster findet also eher eine Beurteilung biologisch verwandter Gruppen von Proteinbindungs-Signalen statt.

Im fünften Kapitel wurde *PromoterMap* in der Praxis überprüft.
Nach der Vorstellung der Lösungsansätze für die drei oben aufgeführten Teilprobleme wurde deren Eignung durch die Untersuchung ausgewählter Beispiele verifiziert. Dabei zeigte sich, dass die Methode insgesamt in der Lage ist, Promotoren ähnlicher Funktionalität in Clustern auf einer Kohonenkarte darzustellen, womit der Nachweis für die generelle Funktionsfähigkeit der vorgestellten Architektur erbracht wurde. *PromotorMap* ist das bisher einzige bekannte Verfahren, dass eine solche Anordnung ermöglicht. Zum ersten Mal ist es damit möglich, eine funktionelle Klassifizierung von Promotoren ohne die Verwendung biologischen Vorwissens über die jeweiligen Sequenzen oder die zugeordneten Gene vorzunehmen. Auf diese Weise lassen sich Rückschlüsse über die Funktionalität unbekannter Sequenzen ziehen, die deren weitere Analyse vereinfachen und in großem Maßstab Zeit und Kosten sparen helfen können.

Die Einzelanwendungen der im Verfahren enthaltenen Schichten liefern zusätzliche Impulse bei der in silico Analyse von DNA:

- Die Gruppierung verwandter Bindungsstellen-Matrizen auf einer Kohonenkarte vereinfacht deren ansonsten aufwändige Unterteilung in disjunkte Mengen biologisch gleichwertiger Motive, sogenannte *Familien*. Auch neu gefundene Matrizen lassen sich auf diese Weise bereits bekannten Familien zuordnen. Diese Art der Einteilung ermöglicht eine stark reduzierte Datenausgabe bei Verfahren, die solche Muster in Nukleotid-Sequenzen suchen.

- Die Visualisierungen auf den SARDNET-Karten können für die direkte Bestimmung expliziter Promotor-Modelle benutzt werden. Dabei bieten die Vorsortierung der ähnliche Bindungsstellen innerhalb der gleichen Bereiche der Karte sowie die durch die Unterschiede in der Aktivierung der einzelnen Neuronen dargestellten Positionen von Positionen innerhalb des Promotors eine Basis, auf der Vergleiche von Muster-Sequenzen vergleichsweise einfach zu bewerkstelligen sind.

Damit ergeben sich zusätzlich zur Möglichkeit, unbekannte Promotoren in funktionelle Gruppen einzuteilen, zwei weitere wichtige Anwendungen des Verfahrens. Zum einen die Reduktion der von Software-Methoden bereitgestellten Ergebnisse und damit eine weitere Verbesserung der Verwertbarkeit dieser Daten, zum anderen die Möglichkeit, bei Promotoren, deren funktionelle Äquivalenz bereits bekannt ist, in einfacher Weise enthaltene gemeinsame Strukturen zu extrahieren. Die Relevanz beider Vorgehensweisen wurde ebenfalls anhand ausgewählter Beispiele dargestellt.

Ein aus biologischer Sicht weiteres wertvolles Ergebnis stellt die aus der Anordnung der Bindungsstellen-Matrizen abzuleitende analoge Gruppierung der zugehörigen Transkriptionsfaktoren dar. Die Erkenntnis, dass die lediglich von den Matrizen aus hergeleitete Anordnung dieser Proteine im wesentlichen den bekannten Proteinklassen entspricht, kann als wichtiges Indiz für die vermutete Existenz eines generellen 'recognition codes' bei den Proteinen aufgefasst werden.

6.2 Ausblick

Über das in dieser Arbeit vorgestellte generelle Funktionsprinzip von *PromoterMap* hinaus sind eine Reihe von Verbesserungen denkbar, die vorhandene Schwächen der Methode reduzieren bzw. die erzielbaren Ergebnisse noch verfeinern könnten. Eine besonders große Rolle spielt dabei die fortwährende Zunahme der verfügbaren biologischen Daten. Durch die für solche Veränderungen offene Architektur der Methode sollten sich vor allem qualitative Steigerungen unmittelbar auf die Ergebnisse auswirken.

Die Vervollständigung und Verbesserung der Bindungsstellen-Matrizen-Sammlungen ist eine der Änderungen, die in näherer Zukunft zu erwarten sind. Die dadurch in zunehmendem Maße spezifischer werdenden Vorhersagen dieser Motive führen zu individuelleren Visualisierungen im ersten Schritt des Verfahrens. Auf diese Weise können im zweiten Schritt Promotoren einander zugeordnet werden, bei denen bisher aufgrund fehlender Matrizen keine Ähnlichkeit festgestellt werden konnte. Auch ist zu erwarten, dass die Wahl der Parameter für die Vorhersage dieser Muster nicht mehr so stark ins Gewicht fällt, wenn qualitativ höherwertige Matrizen verwendet werden. Insgesamt sollte sich damit die Gruppierung auf der zweiten Karte verbessern und so zu einer Qualitätssteigerung beim Ergebnis führen.

Auch bei den Promotoren ist zu erwarten, dass in nächster Zeit – etwa durch die in Kapitel 2 vorgestellten Micro-Array Techniken – zunehmend Beispiele vorhanden sein werden, bei denen sowohl die Struktur als auch die biologische Funktionalität bekannt sind. Findet man solche Exemplare in einem bestimmten Cluster auf der Karte, lassen sich die anderen dort gruppierten Promotoren gezielt auf diese Funktionalität hin untersuchen. Mit der Existenz ausreichend vieler solcher bekannter Promotoren ließe sich die zweite Schicht des Verfahrens eventuell in eine überwachte Methode umwandeln, so dass eindeutige Zuweisungen für unbekannte Promotoren denkbar wären. Dabei böte sich aufgrund ihrer Ähnlichkeit zur bisher verwendeten Methodik das 'learning vector quantization' Verfahren an, das von Kohonen in Anlehnung an die selbstorganisierenden Karten entwickelt wurde [39].

Des weiteren könnte das zunehmende Wissen über die Promotoren dazu führen, dass die software-basierte Vorhersage der Bindungsstellen, die jetzt immer noch viele Redundanzen aufweist, wegfallen kann, und lediglich die für die Bildung des Transkriptionskomplexes relevanten Motive übrig bleiben. Dies wäre der ideale Fall für *Promotormap*. Auch Einschränkungen in der Verfügbarkeit von Transkriptionsfaktoren – etwa in bestimmten Geweben oder zu bestimmten Zeitpunkten der Entwicklung eines Organismus – können bei der Erstellung der SARDNET-Karten berücksichtigt werden. Auf diese Weise lassen sich eventuell Unterschiede in der Wirkungsweise einzelner Promotoren in verschiedenen Regulationsszenarien finden.

Änderung im biologischen Datenbestand können also mehr oder weniger direkt in das Verfahren eingehen. Dabei sind keine grundlegenden Änderungen an der Architektur vonnöten; der Wissenszuwachs kann sofort verwertet werden.

6.3 Fazit

Die Verwendung nichtüberwachter Lernverfahren ist ein geeigneter Ansatz für die Analyse von Promotoren mit Hilfe des Computers. Mit *PromoterMap* wurde eine Methode vorgestellt, die in der Lage ist, Promotoren lediglich auf der Basis der zugrundeliegenden Sequenz biologisch relevant anzuordnen. Wie in anderen Bereichen der Molekularbiologie auch [14, 15, 25, 26] lassen sich also auch ohne ausreichendes Detailwissen auch bei der Promotorerkennung mit solchen Verfahren Erfolge erzielen, wenn man die zugrundeliegende Struktur der betrachteten biologischen Mechanismen miteinbezieht.

Literaturverzeichnis

[1] G. Andreu, A. Crespo, J.M. Valiente (1997): „Selecting the Toroidal Self-Organizing Feature Maps (TSOFM) best organized to object recognition.“, *Proc. of ICNN (International Conference on Neural Networks) 1997*, Vol. II, 1341-6

[2] C.M. Bishop (1995): „Neural Networks for Pattern Recognition“, Oxford, Clarendon

[3] S. Becker, B. Groner, C.W. Müller (1998): „Three-dimensional structure of the Stat3β homodimer bound to DNA“,*Nature* 394, 145-51

[4] D.A. Benson, et al. (2000): „GenBank“, *Nucleic Acids Res.* 28, 15-8

[5] T.A. Brown (1999): „Genomes“, New York, John Wiley & Sons

[6] G.J. Chappell, J.G. Taylor (1993): „The temporal Kohonen map“, *Neural Networks* 6, 1993, 441-5

[7] B. Demeler, G. Zhou (1991): „Neural network optimization for E.coli promoter prediction“, *Nucleic Acids Research* 19, 1593-9

[8] I. Dunham et al. (1999): „The DNA sequence of human chromosome 22“, *Nature* 402, 489-95

[9] J.-M. Claverie, I. Sauvaget (1985): „Assessing the biological significance of primary structure consensus patterns using sequence databanks. I. Heat-shock and glucocorticoid control elements in eukaryotic promoters.“, *Comp. Appl. Biosci.* 2, 95-104

[10] N. Euliano, J. Principe (1996): „Spatio-temporal self-organizing feature maps“, *ICNN* 4, Washington D.C., 1900-5
`http://cnel.ufl.edu/bib/papers/euliano96icnn.ps.gz`

[11] C.L. Fancourt, J. Principe (1996): „A neighborhood map of competing one step predictors for piecewise segmentation of time series“, *ICNN* 4, Washington D.C., 1906-11
`http://cnel.ufl.edu/bib/papers/fancourt96icnn.ps.gz`

[12] C.L. Fancourt, J.C. Principe (1997): „Temporal self-organization through competitive prediction“, *ICASSP'97*, 8-12
`http://cnel.ufl.edu/bib/papers/fancourt97icassp.ps.gz`

[13] D.F. Feng, R.F. Doolittle (1987), *J. Mol. Evol.* 25, 351-60

[14] E.A. Ferrán, P. Ferrara (1991): „Topological maps of protein sequences“, *Biological Cybernetics* 65, 451-8

[15] E.A. Ferrán, P. Ferrara (1991): „Unsupervised clustering of proteins“, in *Artificial Neural Networks*, Proceedings of the First International Conference on Artificial Neural Networks, ed. by T. Kohonen, K. Mäkisara, O. Simula, J. Kangas, North-Holland, Vol. 2, 1341-4

[16] E.A. Ferran, B. Pflugfelder (1993): „A hybrid method to cluster protein sequences based on statistics and artificial neural networks“, *Comp. Appl. Biosci.* 9, 671-80

[17] J.W. Fickett (1996): „Coordinate positioning of MEF2 and myogenin binding sites“, (reprinted from Gene-Combis, 172, GC19-GC32, 1996) *Gene* 172, GC19-GC32

[18] K. Frech, J. Danescu-Mayer, T. Werner (1997): „A novel method to develop highly specific models for regulatory units detects a new LTR in GenBank which contains a functional promoter“, *Journal of Molecular Biology* 270, 674-87

[19] K. Frech, K. Quandt, T. Werner (1997): „Software for the analysis of DNA sequence elements of transcription“, *CABIOS* 13, 89-97

[20] K. Frech, K. Quandt, T. Werner (1998): „Muscle actin genes: A first step towards computational classification of tissue specific promoters.“, *In Silico Biology* 1, 0005. `<http://www.bioinfo.de/isb/1998/01/0005/>`

[21] K. Fukushima (1975): „Cognitron: A self-organizing multilayered neural network“, *Biological Cybernetics* 20, 121-36

[22] M. Glickstein (1988): „The discovery of the visual cortex“, *Scientific American* 259, 84-91

[23] S. Grossberg (1976): „Adaptive pattern classification and universal recording, I: Parallel development and and coding of neural feature detectors.“*Biological Cybernetics* 23, 121-34

[24] H. Gutfreund, M. Mezard (1988): „Processing of temporal sequences in neural networks“, *Physical Review Letters 61* 1988, 235-8

[25] J. Hanke, J.G. Reich (1996): „Kohonen map as a visualization tool for the analysis of protein sequences: multiple alignments, domains and segments of secondary structures“, *CABIOS* 12, 447-54

[26] J. Hanke, G. Beckmann, P. Bork, J.G. Reich (1996): „Self organizing hierarchic networks for pattern recognition in protein sequence“, *Protein Science* 5, 72-84

[27] T. Heinemeyer et al. (1998): „Databases on transcriptional regulation: TRANSFAC, TRRD and COMPEL“, *Nucleic Acids Research* 26, 362-7

[28] T. Heinemeyer, H. Karas (1998): personal communications

[29] G.Z. Hertz, G.W. Hartzell, G.D. Stormo (1990), *Comp. Appl. Biosci.* 6, 81-92

[30] International Human Genome Sequencing Consortium (2001): „Initial sequencing and analysis of the human genome“, *Nature* 6822, 860-921

[31] D.L. James, R. Miikkulainen (1995): „SARDNET: A self-organizing feature map for sequences“, *Advances in Neural Information Processing Systems* 7, ed. by G. Tesauro, D.S. Touretzky, T.K. Leen, 577-84
`ftp://ftp.cs.utexas.edu/pub/neural-nets/papers/james.sardnet.ps.Z`

[32] J. Kangas (1994): „On the analysis of pattern sequences by self-organizing map“, PhD thesis, Helsinki University of Technology, Espoo, Finland
`http://nucleus.hut.fi/ jari/papers/thesis94.ps.Z`

[33] O.V. Kel et al. (1995): „A compilation of composite regulatory elements affecting gene transcription in vertebrates.“, Nucleic Acids Research 23, 4097-103

[34] B. de Ketelaere, D. Moshou, P. Coucke, J. de Baerdemaeker (1997): „A hierarchical Self-Organizing Map for classification problems“, *Proc. of WSOM'97*, Helsinki

[35] C.V. Kirchhamer, C.-H. Yuh, E.H. Davidson (1996): „Modular cis-regulatory organization of developmentally expressed genes: two genes transcribed territorially in the sea urchin embryo, and additional examples.“, *Proc. Natl. Acad. Sci. USA* 93, 9322-8

[36] A. Klingenhoff et al. (1999): „Functional promoter modules can be detected by formal models independent of overall nucleotide sequence similarity“, *Bioinformatics* 15, 180-6

[37] R. Knippers (1997): „Molekulare Genetik“, Stuttgart, Thieme

[38] T. Kohonen (1982): „Self-organized formation of topologically correct feature maps“, *Biological Cybernetics* 43, 59-69

[39] T. Kohonen (1997): „Self-organizing maps“, 2nd edition, Berlin, Springer

[40] H. Kono, A. Sarai (1999): „Structure-based prediction of DNA target sites by regulatory proteins.“, *Proteins* 35, 114-31

[41] F. Larsen, R. Gundersen, R. Lopez, H. Prydz (1992): „CpG islands as gene markers in the human genome“, *Genomics* 13, 1095-107

[42] N. Lassen, D. Ingvar, E. Skinhoj (1988): „Hirnfunktion und Hirndurchblutung“, *Gehirn und Nervensystem 1988*, 135-43

[43] D.S. Latchman (1995): „Eukaryotic Transcription Factors“, Academic Press

[44] N. Mache, M. Reczko, A. Hatzigeorgiou (1995): „Multistate Time-Delay Neural Networks for the recognition of POL II promoter sequences“,

[45] N. Mache, P. Levi (1996): „Detection of eukaryotic POL II promoters with multi-state time-delay neural networks“, *Proceedings of the German Conference on Bioinformatics (GCB) 1996*, 264-7

[46] Y. Mandel-Gutfreund, H. Margalit (1998): „Quantitative parameters for amino acid-base interaction: implications for prediction of protein-DNA binding sites.“*Nucleic Acids Res.* 15,2306-12

[47] R. Miikkulainen, persönliche Kommunikation

[48] B. Morgenstern, A. Dress, T. Werner (1996): „Multiple DNA and protein sequence alignment based on segment-to-seqgment comparison“, *Proc. Natl. Acad. Sci. USA* 93, 12098-103

[49] S.B. Needleman, C.D. Wunsch (1970), *J. Mol. Biol.* 48, 443-58

[50] M.C. O'Neill (1991): „Training back-propagation neural networks to define and detect DNA-binding sites“, *Nucleic Acids Research* 19, 313-8

[51] M.C. O'Neill (1992): „Escherichia coli promoters: neural networks develop distinct descriptions in learning to search for promoters of different spacing classes“,*Nucleic Acids Research* 20, 3471-7

[52] R.C. Périer, T. Junier, P. Bucher (1998): „The Eukaryotic Promoter Database EPD.“, *Nucleic Acids Res.* 26, 353-7

[53] R.C. Périer, T. Junier, C. Bonnard, P. Bucher (1999): „The Eukaryotic Promoter Database EPD: Recent Developments.“, *Nucleic Acids Res.* 27, 307-9

[54] R.C. Périer et al.(2000):„The Eukaryotic Promoter Database (EPD).“, *Nucleic Acids Res.* 28, 302-3

[55] G. Pesole et al.(2000):„UTRdb and UTRsite: specialized databases of sequences and functional elements of 5' and 3' untranslated regions of eukaryotic mRNAs“, *Nucleic Acids Res.* 28, 193-6

[56] E. Petrovick et al.(1998):„Multiple functional domains of AML1:PU.1 ans C/EBPα synergize with different regions of AML1“, *Mol. Cell. Biol.* 18, 3915-25

[57] D.S. Prestridge (1995): „Predicting PolII promoter sequences using transcription factor binding sites“, Journal of Molecular Biology 249, 923-32

[58] K. Quandt et al. (1993): „MatInd and MatInspector: new fast and versatile tools for detection of consensus matches in nucleotide sequence data“, *Nucleic Acids Research* 23, 4878-84

[59] K. Quandt, K. Grote, T. Werner (1996): „GenomeInspector: Basic software tools for analysis of spatial correlations between genomic structures within megabase sequences“, *Genomics* 33, 301-4

[60] K. Quandt, K. Grote, T. Werner (1996): „GenomeInspector: a new approach to detect correlation patterns of elements on genomic sequences“, *CABIOS* 12, 405-13

[61] M.G. Reese, F.H. Eeckmann (1995): „Novel neural network prediction systems for human promoters and splice sites.“, *Proceedings of the Workshop on Gene-Finding and Gene Structure Prediction*, Philadelphia, ed. by D. Searls, J. Fickett, G. Stormo and M. Noordewier

[62] B.D. Ripley, (1996): „Pattern Recognition and Neural Networks“, Cambridge, Cambridge University Press

[63] H. Ritter, T. Martinetz, K. Schulten (1991): „Neuronale Netze“, Bonn, Addison-Wesley

[64] R. Rojas (1993): „Theorie der neuronalen Netze“, Berlin, Springer

[65] H. Rozenberg, et al. (1998): „Structural code for DNA recognition revealed in crystal structures of papillomavirus E2-DNA targets.“, *Proc. Natl. Acad. Sci. USA* 95, 15194-9

[66] D.E. Rumelhart, D. Zipser (1985): „Feature discovery by competitive learning“,*Cognitive Science* 9, 75-112

[67] H. Sakoe, S. Chiba (1987): „Dynamic programming algorithm optimization for spoken word recognition.“ *IEEE Transaction on Acoustics, Speech and Signal Processing* 26, 43-9

[68] J.W. Sammon Jr (1969): „A non-linear mapping for data structure analysis“ *IEEE Transaction on computers* 18, 401-9

[69] M. Scherf (1998): „Distanzbasierte Merkmalsbewertung“, *Dissertation*, TU München

[70] M. Scherf, A. Klingenhoff, T. Werner (2000): „Highly specific localization of promoter regions in large genomic sequences by PromoterInspector: a novel context analysis approach.“, *Journal of Molecular Biology* 297, 599-606

[71] M. Scherf et al. (2001): „First Pass Annotation of Promoters on Human Chromosome 22.“, *Genome Research* 11, 333-40

[72] M. Schena, D. Shalon, R.W. Davis, P.O. Brown (1995): „Quantitative monitoring of gene expression patterns with a complementary DNA microarray“, *Science* 270, 467-70

[73] V. Solovyev, A. Salamov (1997): „The Gene-Finder computer tools for analysis of human and model organisms genome sequences.“ *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology (ISMB)*, 294-302

[74] J.A. Stamatoyannopoulos, C.H. Clegg, Q. Li (1997): „Sheltering of gamma-globin expression from position effects requires both an upstream locus control region and a regulatory element 3' to the (A)gamma-globin gene“, *Molecular Cellular Biology* 17, 240-7

[75] The EMBL Nucleotide Sequence Database G. Stoesser, et al. (2001): „The EMBL Nucleotide Sequence Database“, *Nucleic Acids Res.* 29,17-21

[76] C. Van der Malsburg (1973): „Self-organizing of orientation sensitive cells in the striate cortex.“, *Kybernetik* 4, 85-100

[77] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K.J. Lang (1989): „Phoneme recognition using time-delay neural networks“, *IEEE Transactions on acoustics, speech and signal processing* 37, 328-39

[78] M.S. Waterman, R. Jones (1990), *Methods Encymol.* 183, 221-37

[79] T. Werner (1999): „Models for prediction and recognition of eucaryotic promoters“, *Mammalian Genome* 10, 168-75

[80] C. Windheuser, J. Kindermann (1990): „Competitive-sequence-learning - A new approach to unsupervised sequence learning“ Parallel processing in neural systems and computers, ed. by R. Eckmiller, G. Hartmann, G. Hauske, 217-22

[81] C. Windheuser (1991): „Spracherkennung mit unüberwacht lernenden konnektionistischen Systemen“

[82] `http://www.dhgp.de/general/hgp/hgp1.html`, „Das Humangenomprojekt“

[83] http://www.gene-chips.com, „DNA Microarray (Genome Chip)“

[84] http://www.cis.hut.fi:80/ hynde/lvq/tars „Neural Networks Research Centre“