R. Merkl

Grundlegende Algorithmen der Bio-Informatik

Kurseinheiten 1 - 7

mathematik und informatik

FernUniversität in Hagen

001 756 486 (10/12) 01738- 4 - 01 - S 1

Inhaltsverzeichnis

V Vorwort

V.1 Allgemeine Information

Der Kurs "Grundlegende Algorithmen der Bioinformatik" basiert auf dem Buch

R. Merkl

Bioinformatik

Grundlagen, Algorithmen, Anwendungen

3., vollständig überarbeitete und erweiterte Auflage, 2015

ISBN Nummer 978-3-527-33820-7 - Wiley-VCH, Weinheim

Das Lehrmaterial ist aufgeteilt in sieben Kurseinheiten, die sich aus den folgenden Kapiteln des Buches zusammensetzen:

Kurseinheit	Buch-kapitel	Inhalt
1	1, 2, 3	Biologische Grundlagen, Sequenzen und ihre Funktion, Datenbanken
2	4, 9	Grundbegriffe der Stochastik, paarweiser Sequenzvergleich
3	5, 10, 11	Bayessche Entscheidungstheorie und Klassifikatoren, Sequenzmotive, Scoring-Schemata
4	12, 13	FASTA und die BLAST-Suite, multiple Sequenzalignments und Anwendungen
5	6, 14	Klassische Cluster- und Klassifikationsverfahren, Grundlagen phylogenetischer Analysen
6	7, 18, 17	Neuronale Netze, Vorhersage der Sekundärstruktur, Support-Vektor Maschinen
7	15, 16	Markov-Ketten und Hidden-Markov-Modelle, Profil-HMMs

Der Basistext bietet eine breite Darstellung der meisten Arbeitsgebiete der Bioinformatik, die wir in diesem Kurs allerdings nicht alle bearbeiten können. Wie im Basistext erläutert, werden auf einer Webseite des Verlages Wiley-VCH begleitende Übungen angeboten. Diese dienen dazu, den praktischen Umgang mit solchen Tools zu erlernen, die aus den vorgestellten Algorithmen entstanden sind. Sie sollten dieses Angebot zur Festigung des Stoffes wahrnehmen.

V.2 Motivation

Es ist sicherlich nicht übertrieben, zu behaupten, dass die moderne biologische Forschung ohne informatische Unterstützung nicht mehr möglich ist. Dies gilt insbesondere für die Molekularbiologie, mit der Lebensvorgänge auf molekularem Niveau studiert werden. Der Bedarf und die Bedeutung der Informatik sind spätestens mit Aufnahme der großen Sequenzierprojekte (Stichwort: Entschlüsselung des menschlichen Genoms) allgemein anerkannt. Synchron zum Bedarf und basierend auf der sich stetig verbessernden Datengrundlage hat sich die Bioinformatik in den letzten Jahrzehnten als eigenständiges Fachgebiet etabliert; Konzepte sowie Algorithmen und Datenstrukturen orientieren sich daher an biologischen Fragestellungen. Deswegen werden Modelle für biologische Objekte wie Sequenzen, Makromoleküle, Stoffwechselvorgänge oder Stammbäume entwickelt, auf denen mit speziellen Algorithmen gerechnet wird. Die bisher entstandenen informatischen Konzepte und Methoden sind jedoch bereits so speziell, dass sich ein Verständnis nur nach intensiver Beschäftigung einstellt. Zudem sind für die Modellierung und das Verstehen der Algorithmen biologische Grundkenntnisse notwendig; es wird aber auch ein fundiertes Repertoire an informatischen Methoden verlangt. Es ist gerade diese Interdisziplinarität, die den Reiz dieses Faches ausmacht. Sie werden sehen, die Beschäftigung mit den Verfahren der Bioinformatik eröffnet Ihnen ein aufregendes und sich mit großer Dynamik entwickelndes Spezialgebiet der Informatik.

V.3 Weshalb und wie werden Algorithmen studiert?

Ehe Sie sich mit Elan und Wissbegierde (warum sonst sollten Sie sich für diesen Kurs interessieren?) auf das Material stürzen, müssen Sie sich nochmals klar machen, was es bedeutet, Algorithmen zu studieren.

Das Erarbeiten neuer Algorithmen ist eine intellektuell äußerst anstrengende Tätigkeit, die Ausdauer und Fleiß erfordert. Leider ist es für die Wenigsten damit getan, die Beschreibung eines Algorithmus einfach nur mehrmals durchzulesen. Diese Einschränkung gilt aufgrund der kompakten Notation ganz allgemein für mathematische oder informatische Darstellungen. Es wird meist notwendig sein, das Präsentierte mit Papier und Bleistift aufzuarbeiten, Algorithmen mit Hilfe von Schemata und Flussdiagrammen zu skizzieren und kleine Übungsaufgaben durchzurechnen. Dem Verständnis hilft auch, sich Alternativmethoden zu überlegen und zu versuchen, die Algorithmen zu verbessern; oft lässt sich der meiste Nutzen aus den dabei gemachten Fehlern ziehen. Sicherlich ist es am Anfang schwierig,

sich von einer vorliegenden Darstellung zu lösen. Ihr Ziel sollte es jedoch sein, zu abstrahieren und zu versuchen, mit eigenen Worten einen Algorithmus zu beschreiben. Dabei kommt es darauf an, die zugrunde liegenden Konzepte und Ideen zu verstehen und klar herauszuarbeiten. Weniger relevant ist die Akkumulation von reinem Faktenwissen. **Im Vordergrund steht das Verständnis der Methoden** und darauf sollten Sie sich konzentrieren. Weshalb ist diese Zielsetzung erstrebenswert? Erst nachdem Sie ein Verfahren verstanden haben, können Sie beispielsweise dessen Grenzen abschätzen oder Algorithmen an neue Gegebenheiten anpassen. Ihren Kenntnisstand können Sie mit den folgenden Ansätzen leicht überprüfen:

- Versuchen Sie, Algorithmen schematisch darzustellen.
- Bemühen Sie sich, Stärken und Schwächen der Ansätze herauszuarbeiten.
- Versuchen Sie zu begründen, weshalb für eine bestimmte Aufgabe genau der vorgestellte Algorithmus verwendet wird.
- Überlegen Sie sich, welche Kriterien für die Entwickler wohl den Ausschlag gaben.
- Falls Sie einen besseren Vorschlag parat zu haben, machen Sie sich an eine Implementierung und bestimmen Sie die Qualität Ihrer Lösung.

Zusätzlich sollten Sie sich bemühen, gegenseitige Abhängigkeiten im Dargestellten zu erkennen. Manche Zusammenhänge lassen sich leider nicht streng linear präsentieren. Bezüge auf Kommendes sind manchmal unvermeidlich. Da hilft nur, Geduld mit dem Autor und sich selbst zu haben.

Falls Sie mit der vorliegenden Darstellung überhaupt nicht zurechtkommen, sollten Sie nach einer Alternative Ausschau halten. Häufig fördert die Betrachtung einer Problematik unter einem anderen Blickwinkel den Lernfortschritt enorm. Nutzen Sie das Internet als Informationsquelle. Gerade zur Bioinformatik finden Sie dort eine Fülle von Material.

V.4 Zum Kursinhalt

Der Kurs stellt Kernalgorithmen der Bioinformatik vor. Einen großen Raum nehmen Verfahren ein, die auf Sequenzen arbeiten. Da es relativ einfach ist, mit nasschemischen Verfahren RNA- oder DNA-Sequenzen und damit indirekt auch die von Proteinen zu bestimmen, spielen Sequenzen sowie die zugehörigen Algorithmen in der Bioinformatik eine wichtige, vielleicht die bedeutendste Rolle. Daneben werden zur Lösung bioinformatischer Fragestellungen aber auch solche Methoden verwendet, die in anderen informatischen Spezialdisziplinen wichtig sind, wie *Data-Mining* oder statistische Ansätze. In diesem Kurs geht es jedoch im Wesentlichen um eine Einführung in diejenigen Algorithmen, die in der Bioinformatik von besonderer Relevanz und allgemeinem Interesse sind.

Die Bioinformatik ist eine sich rasch wandelnde angewandte Informatik, die sich häufig an den Bedürfnissen der biologischen Forschung orientiert und die oft Richtung und Ansprüche vorgibt. Ein einführender Kurs wird immer nur einen Teil relevanter Algorithmen abdecken können. Allerdings schaffen Sie sich mit der sorgfältigen Bearbeitung des Kursmaterials Grundlagen, die es Ihnen leicht machen, sich auch in solche Algorithmen und Methoden einzuarbeiten, die hier nicht behandelt werden können. Das Kursmaterial, oder genauer, die zu bearbeitenden Themen sind wie folgt auf sieben Kurseinheiten aufgeteilt.

In der ***ersten Kurseinheit*** werden zunächst biologische Zusammenhänge vorgestellt, die notwendig sind, um die Algorithmen zu verstehen. Zur Modellierung werden in der Regel biologische Konzepte verwendet; daher ist es wichtig, die zugrunde liegenden biologischen Fakten zu kennen. Zusätzlich werden Sequenzen und einige der speziellen Datenbanken eingeführt.

In der ***zweiten Kurseinheit*** wiederholen wir zunächst einige Grundlagen der Stochastik und führen spezielles Wissen ein, das in der Bioinformatik eine wichtige Rolle spielt. Anschließend lernen Sie erste Algorithmen der Bioinformatik kennen. Wir beginnen mit dem Studium einfacher Verfahren, ehe wir uns dem wohl bedeutendsten und am intensivsten bearbeiteten Teilproblem der Bioinformatik zuwenden, dem Sequenzvergleich. Die klassischen Methoden beruhen auf der Berechnung einer Editierdistanz, d. h. einem Konzept, das wir in mehreren Runden verfeinern und so der biologischen Problemstellung anpassen.

In der ***Kurseinheit drei*** beschäftigen wir uns zunächst mit der Bayesschen Entscheidungstheorie, auf der viele bioinformatische Ansätze beruhen. Dies gilt insbesondere dann, wenn eine Entscheidung zwischen zwei Alternativen zu treffen ist. Bayessche Ansätze bilden auch die Grundlage für *Scoring-Schemata*, die für den Sequenzvergleich benötigt werden. Scores geben für jedes Paar von Symbolen die Ersatz- d. h. die Substitutionshäufigkeit an, die je nach Fragestellung bestimmt und gewählt werden muss. Unter Sequenzmotiven, einem weiteren Thema dieser Einheit, werden biologische Objekte zusammengefasst, die jeweils dieselbe biologische Funktion besitzen.

Der Sequenzvergleich spielt auch in ***Kurseinheit vier*** die dominante Rolle. Wir werden uns zunächst mit Heuristiken beschäftigen, mit denen die in Kurseinheit zwei eingeführten exakten Methoden des Sequenzvergleichs approximiert werden. Dieses Vorgehen ist beim paarweisen Vergleich mittlerweile zwingend notwendig, da die optimalen Methoden aufgrund der Größe der Datenbanken für das Durchmustern zu zeitaufwendig sind. Mit solchen Programmen, die zwar nicht die Empfindlichkeit der exakten Algorithmen erreichen, können große Sequenzdatenbanken, die Sie in Einheit eins kennengelernt haben, jedoch wesentlich schneller durchsucht werden. Eine Steigerung der Empfindlichkeit der Sequenzvergleichsmethoden wurde durch den Ersatz einer einfachen Sequenz durch ein sogenanntes *Profil* erreicht. Dieses gibt für eine Sequenzmenge in Form einer Matrix an, wie häufig die Symbole an den einzelnen Positionen vorkommen. Eine

präzise Beschreibung von Sequenzmotiven und insbesondere von Proteinfamilien resultiert aus den multiplen Sequenzalignments. Wir werden uns mit einigen grundsätzlichen Problemen sowie mit implementierten Heuristiken zur Konstruktion von multiplen Alignments beschäftigen.

Die Berechnung von Stammbäumen steht im Mittelpunkt von ***Kurseinheiten fünf***. Multiple Sequenzalignments, die wir bereits in Kurseinheit vier kennengelernt haben, bilden die Grundlage für *phylogenetische Verfahren*. Seit Darwins Theorie von der Entwicklung der Arten gilt als gesichert, dass sich Spezies durch Mutation und Selektion, d. h. durch Abspaltung aus einem gemeinsamen Vorfahren entwickelt haben. In dieser Kurseinheit werden Verfahren und Konzepte vorgestellt, mit denen solche Verwandtschaftsbeziehungen modelliert und abgeleitet werden können. Je nach vorliegendem Datenmaterial werden hierbei unterschiedliche Algorithmen eingesetzt und wir werden auf die wichtigsten Konzepte eingehen. Am Anfang der Einheit steht die Beschäftigung mit klassischen Cluster- und Klassifikationsverfahren.

In ***Kurseinheit sechs*** werden zwei Methoden zur Vorhersage der Sekundärstruktur eingeführt. Zwischen der Proteinsequenz, die manchmal auch als Primärstruktur bezeichnet wird und der 3D-Struktur eines Proteins ist die Proteinsekundärstruktur (2D-Struktur) angesiedelt. Diese besteht aus regelmäßigen Anordnungen von Atomen des Hauptkettenverlaufs (auch *backbone* genannt). Die besten Methoden zur Vorhersage der 2D-Struktur beruhen auf der Auswertung eines Profils (siehe Einheit drei) durch ein System von neuronalen Netzen. Daher werden wir uns, ehe Verfahren zur Proteinsekundärstrukturvorhersage vorgestellt werden, zunächst mit *neuronalen Netzen* beschäftigen. Als zweite, wichtige Sekundärstruktur wird die von RNA-Molekülen eingeführt und erläutert, wie sie mithilfe *dynamischer Programmierung* vorhergesagt werden kann. Neben den neuronalen Netzen sind Support-Vektor Maschinen wichtige und häufig verwendete Verfahren des maschinellen Lernens. Wir werden uns mit den grundlegenden Konzepten dieser Verfahren beschäftigen, die meist als Klassifikatoren dienen.

In den ***Kurseinheiten sieben*** werden wir uns Theorie und Anwendung von Hidden-Markov-Modellen erarbeiten. Diese statistischen Modelle wurden zuerst erfolgreich in die Spracherkennung eingeführt, ehe sie für die Bioinformatik entdeckt wurden. Mittlerweile sind sie fest etabliert, um z. B. Proteinfamilien zu modellieren. Wir werden intensiv die Konzepte, sowie die Algorithmen studieren, um zu lernen, derartige Modelle zu generieren, zu befragen und Parameter zu schätzen. Daneben werden Anwendungen diskutiert, wie das Erkennen von CpG-Inseln und das Beschreiben von Proteinfamilien.

V.5 Rahmenbedingungen bioinformatischer Verfahren

Die Beschäftigung mit biologischen Objekten bedingt gewisse Techniken oder Ansätze, die anfangs möglicherweise etwas befremdlich wirken. Üblicherweise werden in der Informatikausbildung Algorithmen vorgestellt, die optimale Lö-

sungen berechnen und deren Korrektheit bewiesen werden kann. In der Bioinformatik ist die Performanz mancher Methoden eher dürftig. Zudem werden häufig Heuristiken verwendet, also Verfahren, die *in der Regel* gute Lösungen finden. *Optimale* Lösungen sind hierbei jedoch nicht garantiert. Weshalb gibt es für manche Problemstellungen keine 100 % präzisen Algorithmen? Zum einen ist die Komplexität des Lösungsraumes in vielen Fällen enorm groß, sodass nicht alle Möglichkeiten rechnerisch bewertet werden können. Zum anderen mangelt es den Modellen, die den Rechnungen zugrunde liegen, an Präzision. Vor allem aber sind biologische Objekte wie ein Enzym nicht auf einen einzigen Parameter hin optimiert. Zudem wird bei jeder evolutionären Entwicklung die Optimierung nur soweit getrieben, wie es notwendig ist, um zu überleben. All diese Randbedingungen machen eine informatische Bearbeitung enorm schwierig. Im letzten Kapitel des Basistextes wird auf diese spezielle Situation anhand von Beispielen eingegangen.

V.6 Weitere Literatur

Eine breite Darstellung von Algorithmen für unterschiedliche bioinformatische Probleme finden Sie in

I. Mandoiu und A. Zelikovsky, *Bioinformatics Algorithms: Techniques and Applications*, Wiley, 2008.

Algorithmen auf Sequenzen werden ausführlich dargestellt in

D. Gusfield, *Algorithms on strings, trees and sequences*, Cambridge University Press, 1997.

Eine eher statistisch orientierte Herangehensweise verfolgen

R, Durbin, S.R. Eddy, A. Krough, G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press, 1998.

P. Baldi und S. Brunak, *Bioinformatics, the machine learning approach*, MIT Press, 2001.

W.J. Ewens und G.R. Grant, *Statistical Methods in Bioinformatics*, Springer, 2005.

Mehr anwendungsorientiert ist

A. Lesk, *Introduction to Bioinformatics*, Oxford University Press, 2014.

Auf Strukturen fokussiert ist

J. Gu, P.E. Bourne, (Hrsg.), *Structural Bioinformatics*, Wiley-Blackwell, 2011.

Einen Einblick in die aktuelle Bioinformatikforschung bieten die Zeitschrift *Bioinformatics* des Verlages Oxford University Press, sowie *BMC Bioinformatics*, das als *open access* Journal auf alle Artikel unbeschränkten Internet-Zugriff gewährt.

1 Biologische Grundlagen, Datenstrukturen und -bestände

Algorithmen der Bioinformatik operieren auf Datenstrukturen, die wiederum biologische Objekte modellieren. Zu diesen Objekten gehören Makromoleküle wie die DNA, die RNA oder Proteine. Dazu kommen abstraktere, wie z. B. Verwandtschaftsbeziehungen, die in Taxonomiebäumen nachgestellt werden. Zunächst beschäftigen wir uns mit der Modellierung der genannten Makromoleküle und mit Algorithmen, die auf diesen Modellen arbeiten. In der Kurseinheit eins werden zunächst die biologischen Grundlagen gefestigt, sowie die wichtigste Datenstruktur - die Sequenz - vorgestellt, so wie sie im biologischen Kontext eingeführt ist. Anschließend lernen Sie einige wichtige Datenbanken kennen, in denen biologisches Wissen deponiert ist.

1.1 Arbeitspensum

Bitte bearbeiten Sie im Basistext die Kapitel

1 Biologische Grundlagen

2 Sequenzen und ihre Funktion

3 Datenbanken

1.2 Lernziele

Nach der Bearbeitung der Kurseinheit eins sollten Sie

- die Bausteine kennen, aus denen DNA, RNA und Proteine aufgebaut sind,
- die wichtigsten Sekundärstrukturelemente und das Konzept der Proteindomäne verstanden haben,
- die Bedeutung des genetischen Codes erfasst haben,
- erkannt haben, dass Codonen, Aminosäuren und andere Objekte mit unterschiedlichen Häufigkeiten auftreten,
- die Datenstruktur Sequenz verinnerlicht haben,
- wichtigste Sequenzdatenbanken nennen können,
- einige Ansätze kennen, die es erlauben, die Zusammensetzung von Sequenzen zu vergleichen,
- den Aufbau von Ontologien erläutern können,
- die wichtigsten Eigenschaften der Gen-Ontologie verstanden haben.

1.3 Der Aufbau von DNA, RNA und Proteinen

Viele Algorithmen, mit denen wir uns beschäftigen werden, operieren auf Sequenzen. Diese sind ein sehr abstraktes Modell für bestimmte Makromoleküle, die *in vivo* eine höchst komplexe dreidimensionale Struktur einnehmen, in der Sequenz jedoch als *Zeichenkette* angegeben werden. Studieren Sie bitte die physikalisch-chemischen Eigenschaften der Bausteine, die der Synthese der genannten Makromoleküle dienen. Wir beschreiben diese Bausteine später nur noch durch Symbole; die Algorithmen müssen die Eigenschaften der Bausteine hinreichend präzise modellieren. Der GC-Gehalt der DNA ist ein wichtiger Parameter, da zwischen G:C-Basenpaarungen drei Wasserstoffbrückenbindungen ausgebildet werden, zwischen A:T-Paaren nur zwei, vergleiche Abbildung 1.2.

In den letzten Jahren hat sich herausgestellt, dass RNA-Moleküle wichtige, bisher nicht bekannte Funktionen besitzen. Die bioinformatische Beschäftigung mit der RNA ist deswegen besonders relevant geworden, da mithilfe neuester Sequenzierverfahren Menge und Vorkommen spezifischer RNA-Sequenzen zu wohldefinierten Zeitpunkten bestimmt werden kann.

Die Eigenschaften der Proteine resultieren aus dem Zusammenspiel der Aminosäurereste, mit denen die Positionen in der Proteinstruktur besetzt sind. Wichtig ist die Bedeutung der Begriffe, mit denen die Eigenschaften der Reste (d. h. der Seitenketten) beschrieben werden. Beachten Sie auch, dass im Venn-Diagramm (Abbildung 1.6) die Teilmengen überlappen und meist mehr als ein Element besitzen. Daraus lässt sich ableiten, dass Aminosäuren zum Teil ähnliche Eigenschaften besitzen und sich zum Teil substituieren (d. h. ersetzen) können. Wie wir später sehen werden, hat dieser Befund erhebliche Konsequenzen für den Vergleich von Sequenzen und die Entwicklung der Algorithmen.

1.4 Genetischer Code

Sie wissen bereits, dass jedes Protein in Form eines Gens in der DNA codiert ist. Die Abbildung von DNA- auf Proteinsequenzen leistet der genetische Code: In einem Gen codieren jeweils drei direkt aufeinander folgende Basen (diese Einheit wird Codon genannt) für eine Aminosäure. Der Code ist eindeutig, allerdings nicht vollkommen universell. Beachten Sie, dass die DNA im Chromosom als Doppelstrang vorliegt und dass beide Stränge Gene codieren können. Zur Decodierung des Gegenstranges muss das *reverse Komplement* gebildet werden, das in Sequenzdatenbanken nicht deponiert wird. Die gemittelten Häufigkeiten, mit der die einzelnen Codonen in einem Genom vorkommen, unterscheiden sich deutlich voneinander. Eine Ursache für diese "Verzerrung" der *codon usage* ist der unterschiedliche tRNA-Gehalt in den einzelnen Spezies. Diese Variationen im Vorkommen der Codonen erzeugen charakteristische Signaturen, die für die Vorhersage von codierenden Bereichen und zur Charakterisierung der Art genutzt werden können. Tabelle 1.2 listet die Codonhäufigkeiten für ein Bakterium.

1.5 Proteinstruktur

Ähnlich wie DNA, sind auch Proteine fädige Makromoleküle, die aus zwanzig, natürlich vorkommenden Grundbausteinen, den Aminosäuren, synthetisiert werden. Die 3D-Struktur wird einerseits durch die Peptidbindung determiniert, andrerseits durch die Aminosäurereste. Wichtig ist, zu verstehen, dass die 3D-Struktur von Proteinen im Wesentlichen durch **schwache** Wechselwirkungen bestimmt wird, die sich aus nichtkovalenten Bindungen ergeben.

Der Verlauf der Hauptkette (des *backbones*) eines Proteins gehört zu einem großen Teil zu regelmäßigen Strukturen, die auf Wechselwirkungen zwischen Atomen des *backbones* zurückzuführen sind. Machen Sie sich bitte klar, wie α-Helices und β-Faltblätter zustande kommen. Von besonderer Wichtigkeit ist auch das Verständnis zu Proteindomänen. Wir werden später Algorithmen kennenlernen, die für das Identifizieren und Lokalisieren von Proteindomänen entwickelt wurden. Weshalb? Nun, die Gesamtfunktion eines Proteins leitet sich ab vom Zusammenspiel seiner Domänen. Diese werden im Rahmen der Evolution häufig als "monolithische" Blöcke in variierenden Kombinationen wiederverwendet.

Zur räumlichen Darstellung von Proteinstrukturen werden Datensätze gehalten, die für jedes Atom dessen 3D-Koordinaten angeben. Die Nummerierung der Atome wird abgeleitet von der, durch die Proteinsequenz vorgegebene Abfolge. Die Menge von Atomen, die zu einem Aminosäurerest gehört, wird häufig auch *Residuum* oder einfach nur *Rest* genannt. Es kann vorkommen, dass Teile der Protein-3D-Struktur experimentell nicht bestimmt werden können. Manche 3D-Datensätze sind daher nicht vollständig, was ihre Verwendung verkompliziert.

Häufig werden Proteine auf den Hauptkettenverlauf reduziert, da dieser die Topologie der Proteinfaltung erkennen lässt. Dieser Verlauf wird durch die Angabe zweier Winkel (ϕ, ψ) pro Residuum vollständig definiert. Wie der Ramachandran-Plot belegt, sind in natürlich vorkommenden Proteinen bestimmte (ϕ, ψ)-Winkelkombinationen verboten und andere aus energetischen Gründen bevorzugt.

1.6 Proteinfamilien

Je nach Sichtweise können Proteine unterschiedlich klassifiziert werden. Steht die Funktion im Vordergrund, so wird die Proteindomäne als Klassifikationskriterium gewählt. Ein alternatives Klassifikationskriterium ist die Menge und Abfolge von Sekundärstrukturelementen, die zu hierarchischen Klassifikationsschemata führt.

1.7 Sequenzen

Ab sofort spielen im Kurs die realen Makromoleküle bei der informatischen Bearbeitung nur noch eine untergeordnete Rolle. Wir betrachten nun häufig nur noch Sequenzen, also Symbolfolgen, mit denen wir eigentlich DNA- oder Proteinmoleküle meinen. In Tabelle 2.3 werden Aminosäurereste hinsichtlich ihres Charakters

gruppiert. Die Alphabete orientieren sich an den wichtigsten Merkmalen der Aminosäuren. Beachten Sie, dass die physikalisch-chemischen Eigenschaften von den Entwicklern der Alphabete unterschiedlich gewichtet werden, sodass die Klassifikation der Aminosäuren variieren kann.

1.8 Vergleich der Sequenzkomposition

Die spezifische Verzerrung des genetischen Codes kann dazu genutzt werden, Unterschiede in der Zusammensetzung codierender DNA mithilfe einer einfachen Distanzfunktion zu berechnen (Gleichung (2.4)). Solche Kennwerte spielen in Metagenomprojekten bei der Zuordnung von DNA zu Arten (Spezies) eine Rolle. Bei diesen Sequenzierprojekten werden DNA-Fragmente analysiert, die von nicht genauer bekannten mikrobiellen Arten stammen. Analog zum Vorgehen bei der DNA kann die Zusammensetzung von Aminosäuresequenzen verglichen werden.

1.9 Ontologien

Von vielen DNA- und Proteinsequenzen kennt man die Funktion, die als *Annotation* die Eigenschaften der Gene bzw. Genprodukte beschreibt. Für eine maschinelle Bearbeitung dieser Terme ist die Verwendung von Ontologien notwendig, die in bioinformatischen Datenbanken Standard geworden sind. Jede Ontologie besteht aus einer präzise definierten Menge von Begriffen und Beziehungen, mit denen die Begriffe verknüpft werden. Für bestimmte Relationen kann die semantische Ähnlichkeit der Objekte berechnet werden wie Gleichungen (2.11) und (2.16) belegen. Diese Techniken erlauben einen *sequenzunabhängigen* Vergleich von Genen und Proteinen.

1.10 Datenbanken

Im Kapitel neun des Basistextes, das wir in der nächsten Kurseinheit studieren, wird gezeigt, wie Funktion oder Struktur von Proteinen durch Ähnlichkeitsbetrachtungen abgeleitet werden können: Hat eine Proteinsequenz eine hinreichende Ähnlichkeit zur Sequenz eines anderen Proteins, dessen Struktur oder Funktion bekannt ist, so kann dessen Annotation (d. h. Funktion) auf die untersuchte Sequenz übertragen werden. Wie später genauer erläutert wird, gilt dieses Paradigma aufgrund der Abstammung ähnlicher Sequenzen von einem gemeinsamen Vorfahren. Aus dieser Anwendung folgt die besondere Bedeutung der vorgestellten Datensammlungen. Der Vergleich unbekannter Sequenzen mit den Einträgen der GenBank- oder InterPro-Datenbank gehört zu den Routinemethoden der Bioinformatik. Die anderen Datenbanken sollten Sie zunächst nur zur Kenntnis nehmen. Sie belegen, welchen Umfang und Grad von Komplexität diese Datensammlungen mittlerweile angenommen haben; wir werden später auf einige der Datenbanken zurückgreifen. Neben den Sequenzdatenbanken gibt es eine Vielzahl weiterer, die sich speziellen Datensätzen widmen.

1.11 Selbsttestaufgaben

Aufgabe 1.1:

Eine wichtige Aufgabe bei der Entschlüsselung von Genomen ist das Identifizieren von Genen, *nachdem* im Labor die Sequenz aufgeklärt wurde. Um in längeren DNA-Sequenzen codierende Bereiche und das Leseraster festzulegen, kann dasjenige Raster bestimmt werden, das zu den Codonhäufigkeiten der biologischen Art am besten "passt". Gegeben sei das folgende Sequenzfragment aus dem Genom von *Escherichia coli*. Bestimmen Sie das "Codierpotential" für die ersten sechs Codonen in den drei Leserastern des Sinnstranges. Verwenden Sie für die Berechnung die Codonhäufigkeiten aus Tabelle 1.2 des Basistextes und multiplizieren Sie die Werte. Welcher Leserahmen ist mit hoher Wahrscheinlichkeit der codierende? Können Sie die Idee für einen Algorithmus skizzieren, mit dem der Leserahmen vorhergesagt werden kann?

```
>genomische Sequenz
TGGCACCCGTCTGTTTAAAGGC
```

Aufgabe 1.2:

Für die folgende Sequenz ist bekannt, dass sie zu einem β-Strang gehört, der an der Proteinoberfläche liegt. Welche Orientierung im Hinblick auf die Zuwendung zum Proteininneren / Lösungsmittel sagen Sie vorher? Benutzen Sie das zweiwertige Alphabet aus Tabelle 2.3 des Basistextes, das die Aminosäuren in hydrophile und hydrophobe einteilt.

β-Strang: Arg, Ile, Cys, Trp, His, Ile, Ser, Leu, Ser, Val

Aufgabe 1.3:

Gegeben seien die folgenden beiden Fragmente *A* und *B* aus Gensequenzen, die für Proteine codieren. Wie groß ist *Cdn_contr*(*A*, *B*)?

```
A   TTTTTATTCTTTTTGTTATTT
B   TTATTATTTTTCTTGTTATTG
```

Aufgabe 1.4:

In Tabelle 3.1 des Basistextes ist ein Ausschnitt aus einem pdb-File gelistet. Wie groß ist der räumliche Abstand zwischen den Atomen 41 und 42?

2 Grundbegriffe der Stochastik, paarweiser Sequenzvergleich

In dieser Kurseinheit beschäftigen wir uns zunächst mit der Wiederholung wichtiger stochastischer Begriffe. Im Zentrum der Einheit steht jedoch der paarweise Sequenzvergleich. Eine bedeutende Aufgabe der Bioinformatik ist es, zu berechnen, welche Teilzeichenketten in zwei zu vergleichenden Strings (Sequenzen) wo vorkommen. Das einfachste, hierfür geeignete Verfahren ist der Dotplot, mit dem leicht erkannt werden kann, wo *identische* Teilzeichenketten liegen. Meist kommt es jedoch darauf an, *zueinander ähnliche* Teilzeichenketten zu finden. Daher müssen aufwendigere Methoden genutzt werden, die in den Verfahren zum Berechnen globaler und lokaler Alignments implementiert wurden. Diese Algorithmen gehören zu den wichtigsten bioinformatischen Ansätzen. Aufgrund der zentralen Bedeutung des Vergleichs von Sequenzen werden uns diese Verfahren im weiteren Kursverlauf in abgewandelter Form häufiger begegnen.

2.1 Arbeitspensum

Bitte bearbeiten Sie im Basistext die unten angegebenen Abschnitte der Kapitel

4 Grundbegriffe der Stochastik

9 Paarweiser Sequenzvergleich

2.2 Lernziele

Nach dem Bearbeiten der Kurseinheit zwei sollten Sie

- mit Grundbegriffen der Stochastik, sowie Maximum-Likelihood-Schätzern und der Neyman-Pearson-Methode vertraut sein,
- den Algorithmus Dotplot auch auf andere Probleme übertragen können,
- den Begriff der Editierdistanz verstanden haben,
- das Konzept des dynamischen Programmierens erläutern können,
- Eigenschaften und Unterschiede lokaler und globaler Alignments kennen,
- die Begriffe Distanz und Ähnlichkeit zueinander in Beziehung setzen und gegeneinander abgrenzen können.

2.3 Grundbegriffe der Stochastik

Die meisten der eingeführten Begriffe sind Ihnen sicherlich bekannt; der Text dient im Wesentlichen der Wiederholung und als Formelsammlung. Wichtig ist der Begriff der Zufallsvariablen, dem wir im folgenden Text häufig begegnen werden, da viele bioinformatische Algorithmen eine stochastische Komponente besitzen. Weniger geläufig sind Markov-Ketten höherer Ordnung, die in der Bioinformatik bei der Modellbildung z. B. zur Genvorhersage eine wichtige Rolle spielen. Von besonderer Bedeutung sind die Abschnitte zu Maximum-Likelihood-Schätzern (Abschnitt 4.9) und zur Neyman-Pearson-Methode (Gleichung (4.54)). Diese Konzepte sind z. B. die Basis für die Entwicklung von Klassifikatoren und von Scoring-Schemata. Da es kein besseres Verfahren gibt als die vorgestellte Quotientenbildung, wird es in der Bioinformatik sehr häufig eingesetzt.

2.4 Paarweiser Sequenzvergleich

In der Einführung zu Kapitel neun des Basistextes wird eingehend begründet, weshalb der Vergleich von Zeichenketten (nichts anderes tun wir, wenn wir Sequenzen paarweise bewerten) in der Bioinformatik eine derartige Bedeutung hat. Bitte machen Sie sich klar, dass in der Regel eine Übereinstimmung von ca. 25 % der Residuen zweier Proteinsequenzen ausreicht, um auf ähnliche Raumstruktur der Proteine schließen zu können; vergleiche Abbildung 9.1.

2.5 Dotplots

Die Grundidee des Dotplots (Algorithmus 9.1) ist eine ganz einfache; auch die Abschätzung der Laufzeit zu $O(n^2)$ ist leicht nachzuvollziehen. Trotz seiner Einfachheit wird dieser Algorithmus in der Bioinformatik eingesetzt, z. B. um Genome miteinander zu vergleichen. Abbildungen 9.4 und 9.5 zeigen, wie Unterschiede in den Genomen nahe verwandter Arten mithilfe eines Dotplots erkannt werden können. Wie wird hierbei die Ähnlichkeit von Genpaaren bestimmt? Mit exakt den Verfahren, die uns im Folgenden länger beschäftigen. Uns dient der Dotplot hauptsächlich aus didaktischen Gründen als Einstieg in die Algorithmen zur Berechnung von Editierdistanzen mittels dynamischer Programmierung.

2.6 Dynamisches Programmieren

Möglicherweise haben Sie diese Programmiertechnik bereits im Zusammenhang mit effizienter Matrizenmultiplikation kennengelernt. Hier werden Sie eine Anwendung studieren, die auf höchst elegante Weise eine Distanz zwischen Zeichenketten berechnet. Abbildung 9.6 macht das Prinzip deutlich.

2.7 Distanzen und Metriken

Diese beiden Begriffe gehören sicherlich zu Ihrem mathematischen Grundwissen. Zunächst betrachten wir einige Metriken, die in der Bioinformatik eine Rolle spielen. Anschließend wenden wir uns der Levenshtein-Distanz zu, mit der eine

Metrik auf Zeichenketten motiviert werden kann. Im Gegensatz zur Hamming-Distanz setzt dieses Konzept nicht voraus, dass die zu vergleichenden Zeichenketten dieselbe Länge besitzen.

2.8 Berechnung der Levenshtein-Distanz

Aus der Definition der Levenshtein-Distanz lässt sich zunächst nicht unmittelbar ein Berechnungsverfahren ableiten. Der Algorithmus 9.2 erschließt sich beim Studium der Beweisskizze. Beachten Sie bitte, dass einer der Eingabeparameter diejenige Funktion ist, mit der **Distanzen zwischen Symbolen** bewertet werden. Mithilfe dieses Bewertungsschemas wird Wissen aus der Anwendungsdomäne in den Algorithmus importiert. Hierbei sind die Objekte geeignet zu modellieren; daher müssen beim Vergleich von Proteinsequenzen die physikalisch-chemischen Eigenschaften der Aminosäuren bewertet werden.

In der Bioinformatik sind *Alignments* von besonderer Bedeutung. Darunter werden Sequenzanordnungen verstanden, die durch Verschieben von Symbolen und möglicherweise durch Einführen von Lücken entstehen und hinsichtlich einer Zielfunktion optimal sind. Bei der vorgestellten Art der Distanzberechnung fällt die Ableitung eines Alignments praktisch automatisch mit an. Es muss nur, ausgehend vom Eintrag *n*, *m* der Matrix, der optimale Pfad zu einem Rand der Matrix hin zurückverfolgt werden. Dieses Zurückverfolgen wird *traceback* genannt; vergleiche Abbildung 9.12. Die Vorgängerposition ist jeweils durch diejenige Zelle definiert, deren Wert in Gleichung (9.10) zum minimalen Term beitrug.

Bitte machen Sie sich klar, dass dieses Verfahren (beweisbar!) stets die minimale Editierdistanz berechnet. In jedem Teilschritt wird jeweils das optimale Teilergebnis ermittelt. Voraussetzung hierfür ist, dass die betrachteten Distanzen auf Zeichenketten durch **Addition von Distanzwerten** auf Symbolen berechnet werden können. Wie das Beispiel der Routenplanung illustriert, kann dynamische Programmierung auf unterschiedlichste Probleme angewandt werden, sofern die genannten Voraussetzungen erfüllt sind.

2.9 Die Ähnlichkeit von Sequenzen

Die Begriffe Distanz und Ähnlichkeit sind zueinander dual, die Verwendung eines Ähnlichkeitsmaßes hat in der Bioinformatik gewisse Vorteile: Dieses Schema kann durch eine geeignete Wahl von Scores, die anstelle der Distanzen treten, den jeweiligen Gegebenheiten leichter angepasst werden. Die Umstellung der Algorithmen von Distanz- auf Ähnlichkeitsberechnungen macht erstaunlicherweise keinerlei Schwierigkeiten, es muss nur die *min*-Funktion durch *max* ersetzt werden.

Im Kapitel eins haben wir gelernt, dass Proteine aus Domänen zusammengesetzt sind, die in unterschiedlicher Anzahl und Folge vorkommen können. Es ist daher sinnvoll, die Algorithmen zum Vergleich von Proteinsequenzen auf diese Eigen-

schaft hin anzupassen. Leiten Sie bitte aus Gleichung (9.13) ab, wie die Berechnung lokaler Alignments erreicht wird. Mit der Änderung der Score-Berechnung geht eine veränderte Initialisierung der ersten Zeile und Spalte in der Matrix *S* einher. In diesem Fall werden die entsprechenden Zellen mit dem Wert null initialisiert. Hierfür gilt dieselbe Begründung wie für das Einführen der Null in Gleichung (9.13): Jede der genannten Positionen hat die gleiche Chance, der Anfang eines lokalen Alignments zu werden.

Was ändert sich für das Bestimmen der Scores und des Alignments? Der größte Score-Wert muss nun nicht mehr notwendigerweise in der Zelle mit den Indizes *n, m* vorkommen. Zur Identifizierung des maximalen lokalen Alignments wird diejenige Zelle gesucht, die den *höchsten* Eintrag aufweist. Ausgehend von dieser Zelle wird wiederum durch *traceback* das lokale Alignment mit dem bereits eingeführten Verfahren ermittelt. In der Matrix können gleichberechtigt mehrere lokale Alignments mit identischen oder vergleichbar signifikanten Score-Werten auftreten. Das Bestimmen der zugehörigen Alignments erfolgt wie beschrieben.

Interessanterweise hat es mehr als ein Jahrzehnt gedauert, bis die Modifikation des Needleman-Wunsch-Algorithmus hin zum Smith-Waterman-Verfahren entwickelt und publiziert wurde. Es dauerte eben seine Zeit, bis das statistische Fundament entwickelt und belastbar war. Der Needleman-Wunsch-Algorithmus wurde mehrere Male von verschiedenen Gruppen "entdeckt"; dynamische Programmierung wird auch in anderen informatischen Fachdisziplinen verwendet.

2.10 Die adäquate Bewertung von Lücken

Betrachten wir die Domänenstruktur von Proteinen speziell im Hinblick auf die Länge von Lücken, so ist leicht einzusehen, dass das bisher verwendete Scoring-Verfahren zur Lückenbewertung nicht optimal ist. Fehlt beispielsweise einer von zwei Proteinsequenzen eine Domäne, so macht es keinen Sinn, die Vergleichssequenz unterbrochen durch eine Vielzahl kleiner Lücken breit über die andere Sequenz zu "verschmieren". In solchen Fällen ist es korrekter, genau eine, allerdings längere Lücke einzuführen. Auch für das Alignment von DNA-Sequenzen lassen sich Argumente ins Feld führen, die für eine modifizierte Bewertung von Lücken plädieren. Als gut geeignet für die Modellierung biologischer Phänomene haben sich sogenannte *affine Kostenfunktionen* erwiesen, die aus zwei Anteilen bestehen; siehe Gleichung (9.14). Damit wird erreicht, dass ein Öffnen (Einführen) einer Lücke zunächst "teuer" ist. Ein Verlängern einer bestehenden Lücke wird dann allerdings wesentlich "günstiger".

Aufgrund der Verwendung einer affinen Kostenfunktion wächst zunächst die Komplexität des Algorithmus um eine Potenz. Mit geeigneter Programmiertechnik lässt sich jedoch die Komplexität wieder auf $O(n^2)$ drücken.

2.11 Selbsttestaufgaben

Aufgabe 2.1:

Das Genom von *Escherichia coli* hat einen GC-Gehalt von 52 %. Wir nehmen im Folgenden an, dass p(A) = p(T) = 24 % und p(C) = p(G) = 26 % sei. Berechnen Sie die Wahrscheinlichkeit für das Vorkommen von CTA, GCG und ATA. Unterstellen Sie Unabhängigkeit für das Auftreten der Nucleotide. Vergleichen Sie Ihre Ergebnisse mit den Werten aus Tabelle 1.2 des Basistextes. Was schließen Sie aus Ihren Ergebnissen?

Aufgabe 2.2:

Dotplots eignen sich sehr gut dazu, eine wichtige Aufgabe des paarweisen Sequenzvergleichs zu illustrieren: Es gilt, die Übereinstimmung von Zeichenketten zu untersuchen und gemeinsame Teilzeichenketten und Lücken zu finden. Vergleichen Sie mithilfe eines Dotplots die Zeichenketten:

A = GARFIELDTHECAT
B = GARFIELDTHEVERYFATCAT

Interpretieren Sie die Zeichenketten als Proteinsequenzen und einmal vorkommende Infixe als "Proteindomänen". Wie machen sich im Dotplot Unterschiede in der Zusammensetzung bemerkbar? Was bewirkt das Vertauschen von *A* und *B*?

Aufgabe 2.3:

Gegeben seinen drei Domänen D1, D2, D3, die keine Sequenzähnlichkeit zueinander aufweisen. Das Protein *A* bestehe aus der Abfolge D2, D3, D2, das Protein *B* bestehe aus D1 gefolgt von D3. Wie sieht das Alignment aus, wenn *i*) der Smith-Waterman-Algorithmus und *ii*) der Needleman-Wunsch-Algorithmus mit affiner Kostenfunktion verwendet wird? Skizzieren Sie Ihre Lösung mithilfe von Rechtecken, die für die Domänen stehen.

Aufgabe 2.4:

Die Abbildung 9.5 des Basistextes zeigt den Vergleich der Genome einer pathogenen *Escherichia coli* Art und einer nicht-pathogenen Art. In welchen Regionen des Genoms würden Sie nach Genen suchen, die für die Pathogenität verantwortlich sind?

Aufgabe 2.5:

Die Berechnung eines globalen Alignments könnte auch rekursiv bestimmt werden. Dann würde z. B. initial das Programm mit NW_REC(*A*, *B*, *n*, *m*) aufgerufen. Formulieren Sie eine Lösung (ohne affine Kostenfunktion) und erläutern Sie, weshalb diese Strategie nicht in Betracht gezogen wird.

3 Bayessche Entscheidungstheorie, Sequenzmotive, Scoring-Schemata

Im Zentrum der Kurseinheit drei stehen Scoring-Schemata, die für viele bioinformatische Fragestellungen benötigt werden. Bei der Beschäftigung mit Algorithmen zum paarweisen Sequenzvergleich in Kurseinheit zwei ist klar geworden, dass alleine über das Scoring-Schema Wissen aus der Anwendungsdomäne in die Berechnung einfließt. Ein wichtiger Teil beim Entwickeln von Scoring-Schemata ist das Ablegen von Scores (Gewichte) für den paarweisen Vergleich von **Symbolen** in Matrizen. Häufig unterscheiden sich Sequenzfragmente in ihrer Zusammensetzung, obwohl sie alle dieselbe Funktion z. B. als Bindestelle besitzen. Für die Beschreibung solcher Sequenzmengen oder Sequenzmotive wurden mehrere Ansätze vorgeschlagen, die wir in dieser Kurseinheit ebenfalls kennenlernen werden. Grundlage für die Berechnung von Scores sind häufig Chancenquotienten, die wiederum auf der Bayesschen Entscheidungstheorie basieren. Mit den wichtigsten Konzepten dieser Theorie beschäftigen wir uns zu Beginn der Einheit.

3.1 Arbeitspensum

Bitte bearbeiten Sie im Basistext die unten angegebenen Abschnitte der Kapitel

5 Bayessche Entscheidungstheorie und Klassifikatoren

10 Sequenzmotive

11 Scoring-Schemata

3.2 Lernziele

Nach dem Bearbeiten der Kurseinheit drei sollten Sie

- die Grundzüge Bayesscher Entscheidungstheorie verstanden haben,
- ROC-Kurven aufnehmen und interpretieren können,
- verschiedene Verfahren zum Beschreiben von Sequenzmotiven nennen und anwenden können,
- die wichtigsten Substitutionsmatrizen charakterisieren und ihre Herleitung erklären können.

3.3 Bayessche Entscheidungstheorie

Nach der Beschäftigung mit dem Neyman-Pearson-Lemma in Kurseinheit zwei ist die Bayessche Entscheidungsregel sofort einsichtig: Wir entscheiden uns stets für die Klasse bzw. das Modell, dessen Likelihood am größten ist. In der Bioinformatik genügt es jedoch häufig nicht, sich bei Entscheidungen auf eine Zufallsvariable zu stützen. Sind mehrere Parameter zu bewerten, so führt der Fall statistisch unabhängiger Variabler zu einfachen Tests. Beim naiven Bayesschen Klassifikator, der statistische Unabhängigkeit der betrachteten Eigenschaften unterstellt, werden die bedingten Wahrscheinlichkeiten einfach multipliziert. Die Quotientenregel und das Bewerten der Terme für jeweils einen Parameter führen zu den Chancenquotienten der Likelihood-Verhältnisse. Falls es Sie interessiert, wie die Unabhängigkeit von Parametern überprüft werden kann, können Sie dies im Kapitel zur Vorhersage von Protein-Protein-Interaktionen des Basistextes nachlesen. Die Abschnitte zum Marginalisieren und Boosting im Kapitel fünf des Basistextes können Sie überschlagen.

3.4 ROC-Kurven

Die Schwelle, die bei der Bayesschen Regel über die Entscheidung zugunsten einer Alternative entscheidet, muss in Abhängigkeit von den spezifischen Kosten, d. h. den Konsequenzen, gewählt werden. Beim Festlegen einer spezifischen Schwelle hilft die Analyse einer ROC-Kurve. Solche Kennlinien geben zunächst Auskunft über die Qualität des Klassifikators. Zusätzlich erlauben sie die jeweils optimale Wahl der Schwelle. Je nach Anwendung wird die Anzahl falsch positiver oder falsch negativer Vorhersagen festgelegt. Sie sollten sich an der Abbildung 5.3 klarmachen, dass in den Fällen, die wir gewöhnlich betrachten, stets mit einer gewissen Anzahl von Fehlklassifikationen zu rechnen ist. Wir können zwar deren Verhältnis ändern, aber ein Erniedrigen des Anteils Fehler erster Art erhöht automatisch den Anteil von Fehlern zweiter Art und umgekehrt. Diesem Dilemma können wir nicht entkommen. Es ist nur möglich, die für die betrachtete Entscheidungssituation optimale Schwelle zu wählen; hierbei hilft die ROC-Kurve.

3.5 Testen kleiner Trainingsmengen

Häufig stehen bei der bioinformatischen Methodenentwicklung nur kleine Testmengen zur Verfügung. Kreuzvalidierung und *leave-one-out*-Verfahren sind gängige Methoden, mit denen verhindert wird, dass sich Trainings- und Testmengen überlappen. Diese strikte Trennung der Daten muss gesichert sein, wenn die Qualität des Klassifikators in realistischer Weise bestimmt werden soll. In bioinformatischen Anwendungen ist das Verhältnis der positiven zu negativen Testfällen häufig sehr extrem. Beispielsweise sind nur wenige der meist mehreren hundert Residuen eines Proteins an der Katalyse beteiligt. Für die Bewertung der Klassifikationsleistung hat sich in solchen Fällen der Matthews Korrelationskoeffizient (Gleichung (5.17)) bewährt.

3.6 Sequenzmotive

Häufig unterscheiden sich Bindestellen auf der DNA, die den gleichen Faktor betreffen oder auch Proteinfragmente mit *übereinstimmender Funktion* in ihrer Sequenz. Die einfachste Methode, um diese, im Hinblick auf die Funktion zusammengehörenden Sequenzen zu gruppieren, ist ihre Auflistung. Natürlich gibt es intelligentere Methoden, die eine Analyse ihrer Gemeinsamkeiten unterstützen. Naheliegend ist das Berechnen von regulären Ausdrücken, die in diesem Kontext *Signaturen* genannt werden. Eine ganz wichtige Funktion haben in der Bioinformatik multiple Sequenzalignments (MSAs), deren Berechnung wir in der nächsten Kurseinheit studieren werden. Im Moment unterstellen wir deren Existenz. Wie lassen sich MSAs informatisch und statistisch bewerten? Es bietet sich an, Profile zu verwenden und zunächst positionsweise das Vorkommen der einzelnen Symbole zu bestimmen. Diese Häufigkeiten liefern uns einen Überblick zur Variationsbreite an den einzelnen Positionen im Motiv, lassen aber nicht abschätzen, ob diese Variationen statistisch auffällig sind. Wie kommen wir weiter? Wir vergleichen die positionsweise bestimmten Häufigkeiten mit *erwarteten Werten*. Unterstellen wir eine zufällige Verteilung, sollten die Symbole an allen Positionen mit derselben mittleren Häufigkeit vorkommen. Diese Vorgehensweise wird beim Berechnen der Promotor-Scores eingeführt. Die Anwendung Shannonscher Ideen führt auf Sequenz-Logos, die häufig benutzt werden, wenn die Variabilität in einem größeren MSA im Überblick dargestellt werden soll. Eine andere Art, um Gemeinsamkeiten eines MSAs darzustellen, sind Konsensus-Sequenzen. Allerdings geht beim Bilden des Konsensus viel Information verloren, sodass diese Vorgehensweise an Bedeutung verliert.

3.7 Sequenz-Logos

Bitte machen Sie sich anhand der Gleichung (10.1) klar, wie hier Shannonsche Konzepte der Informationstheorie eingesetzt werden. Völlig analog werden Proteinsequenzen mit Logos charakterisiert, beispielsweise in der Pfam-Datenbank.

3.8 Sequenzen niedriger Komplexität

In vielen biologischen Sequenzen kommen repetitive Teilsequenzen vor, die eine statistische Bewertung erschweren. Der SEG-Algorithmus erlaubt es, solche Bereiche auszufiltern. Die genaue Funktionsweise dieses Verfahrens wollen wir hier im Kurs jedoch nicht genauer untersuchen.

3.9 Scoring-Matrizen

Es klang bereits an, dass alleine durch das Bewerten der Lücken und die Scoring-Funktion Wissen aus den Anwendungsdomänen in die Vergleichsverfahren einfließt. Im Kapitel elf des Basistextes lernen Sie zwei Familien von Scoring-Schemata kennen, die für die beiden wichtigsten Anwendungen von Alignment-

verfahren entwickelt wurden: Dies sind das Studium evolutionärer Verwandtschaftsbeziehungen und das Identifizieren von Proteindomänen.

Das zur Theorie von Scoring-Matrizen Gesagte ergibt sich quasi automatisch aus den Ausführungen zum Neyman-Pearson-Lemma und zur Bayesschen Entscheidungstheorie. Mit jedem Score-Wert wird das Vorkommen der betrachteten Symbole in zwei Stichproben verglichen, die für zwei Modelle stehen (Gleichungen (11.1) und (11.2)). Beachten Sie bitte die Bedeutung der Logarithmierung: Damit wird aus dem Produkt von Wahrscheinlichkeiten eine Summe. Genau dies, nämlich eine Scorebildung durch Addition, haben wir bei der Einführung und Definition der Levenshtein-Distanz gefordert. Der Übergang zu Logarithmen hat in der Praxis einen nützlichen Nebeneffekt: Die Gefahr, dass bei der Multiplikation die kleinste, im Rechner darstellbare Zahl unterschritten wird, ist damit erheblich reduziert.

3.10 PAM-Matrizen

Die Bedeutung der PAM-Matrizen hat seit Einführung der BLOSUM-Familie stark abgenommen. Sie spielen für die Charakterisierung von Proteindomänen praktisch keine Rolle mehr. Auf einer größeren Datenbasis beruht die JTT-Matrix. Für phylogenetische Fragestellungen, die wir später genauer studieren, werden neben der JTT-Matrix jedoch häufig wesentlich komplexere Substitutionsmodelle verwendet.

3.11 BLOSUM-Matrizen

Die wichtigste Familie von Scoring-Matrizen ist die der BLOSUM-Matrizen. Machen Sie sich bitte klar, wie sie entwickelt wurden und wie sich die einzelnen Mitglieder unterscheiden. Sie sollten erkennen, dass die Bayessche Entscheidungstheorie und das Neyman-Pearson-Lemma die theoretischen Grundlagen für das Berechnen der Matrizen sind. Überlegen Sie sich bitte, wofür die Zahl 62 im Namen der BLOSUM 62-Matrix steht, deren Inhalt in Tabelle 11.2 gelistet ist.

3.12 Matrix-Entropie

Scoring-Matrizen können mithilfe der Matrix-Entropie charakterisiert werden. Auch hier werden wiederum Shannonsche Konzepte angewendet. Sie sollten den Zusammenhang zwischen den Entropiewerten und dem Datenmaterial, das für die Berechnung der Matrizen verwendet wurde, erklären können.

3.13 Selbsttestaufgaben

Aufgabe 3.1:

Für eine Klassifikationsaufgabe stehen uns zwei Klassifikatoren mit den Kennlinien zur Verfügung, die in Abbildung 5.3 des Basistextes wiedergegeben sind. In der betrachteten Anwendung können wir maximal eine Falsch-Positiv-Rate von 10 % tolerieren. Wie viele der echt positiven Fälle werden mit dieser Bedingung von den beiden Klassifikatoren erkannt?

Aufgabe 3.2:

Die folgenden Aufgaben sollen Ihnen veranschaulichen, wie schwer die Vorhersage von Proteininterfaces tatsächlich ist:

Wir haben für die Verteilung von Interfaceresiduen und Oberflächenresiduen die Wahrscheinlichkeiten mit $p(\omega_1) = 12\ \%$ und $p(\omega_2) = 88\ \%$ bestimmt. Wie groß müsste der Quotient $p(as \mid \omega_1) / p(as \mid \omega_2)$ sein, damit bei der Bayesschen Regel für ein Residuum zugunsten des Interfaces entschieden werden könnte?

Für die Aminosäure Tyrosin gilt: $p(\mathrm{Y} \mid \mathrm{Interf}) = 0.053$, $p(\mathrm{Y} \mid \mathrm{Oberfl}) = 0.034$. Wie groß müsste ein Cluster von benachbarten Tyrosin-Residuen sein, damit gemäß der Bayesschen Regel eine Zugehörigkeit zu einem Interface vorhergesagt würde? Wir gehen von der Unabhängigkeit der Positionen aus.

Aufgabe 3.3:

An zwei Positionen k, l eines MSAs wurden folgende Aminosäurenhäufigkeiten bestimmt:

$p(\mathrm{L} \mid k) = 0.5$, $p(\mathrm{I} \mid k) = 0.5$ sowie $p(\mathrm{D} \mid l) = 0.5$, $p(\mathrm{K} \mid l) = 0.5$.

Wie groß ist diesen Fällen $I(a_i \mid m)$? Benutzen Sie die Werte aus Tabelle 1.3 des Basistextes für die mittleren Häufigkeiten. Wie groß ist der Informationsgehalt an den Positionen k und l? Welcher Aspekt wird beim Shannonschen Konzept nicht berücksichtigt? Vergleichen Sie hierzu die Substitutionshäufigkeiten $S(\mathrm{L}, \mathrm{I})$ und $S(\mathrm{D}, \mathrm{K})$ aus der BLOSUM62-Matrix (Tabelle 11.2 des Basistextes) und die Lage der Aminosäuren in Abbildung 1.6 des Basistextes.

Aufgabe 3.4:

Erläutern Sie, weshalb beim Berechnen eines Alignments durch Addieren der Scorewerte $s_{asi,\ asj}$ gemäß der Bayesschen Entscheidungstheorie gehandelt wird.

4 Heuristischer und profilbasierter Sequenzvergleich, multiple Sequenzalignments

In der Kurseinheit zwei haben wir die klassischen Algorithmen des Sequenzvergleichs gründlich studiert. Wir haben dabei festgestellt, dass diese Algorithmen eine Laufzeit von mindestens $O(n^2)$ aufweisen, die für viele Anwendungen jedoch nicht ausreicht. Algorithmen des paarweisen Sequenzvergleichs dienen häufig dazu, eine Sequenz mit allen Einträgen einer Datenbank zu vergleichen, um der *Query* (der Eingabe) eine Funktion zuzuweisen. Nun enthalten Datenbanken wie die NCBI-Datenbank mehr als 10^8 Einträge, mit denen jede Query verglichen werden muss. Wir werden uns in dieser Kurseinheit zunächst mit zwei Heuristiken auseinandersetzen, die speziell für den oben geschilderten Einsatz entwickelt wurden. Anschließend werden wir Techniken kennenlernen, mit denen die Empfindlichkeit des paarweisen Sequenzvergleichs gesteigert werden kann. Schließlich beschäftigen wir uns mit einigen Methoden des multiplen Sequenzalignments.

4.1 Arbeitspensum

Bitte bearbeiten Sie im Basistext die Kapitel

12 FASTA und die BLAST-Suite

13 Multiple Sequenzalignments und Anwendungen

4.2 Lernziele

Nach dem Bearbeiten der Kurseinheit vier sollten Sie

- die Prinzipien angeben können, mit denen in den Heuristiken zum Sequenzvergleich eine Geschwindigkeitssteigerung erzielt wird,
- Konzepte entwickeln können, mit denen die Empfindlichkeit des paarweisen Sequenzvergleichs gesteigert werden kann,
- die Bedeutung und die speziellen Probleme des multiplen Sequenzvergleichs benennen können.

4.3 FASTA und BLAST

Welche Möglichkeiten gibt es, die exakten Methoden des Sequenzvergleichs zu beschleunigen? Es sind zwei Ideen naheliegend: Aus der Betrachtung von Zelleninhalten $S[i, j]$ kann abgeleitet werden, dass es in vielen Regionen nicht sinnvoll ist, sozusagen rein schematisch weiterzurechnen. Sind die Scores hinreichend niedrig, kann kein signifikanter Treffer mehr vorkommen. Damit ist eine erste Idee zur Performanzsteigerung hinreichend plausibel: Es wird die Matrix nur soweit gefüllt, wie die Score-Werte auf interessante Treffer schließen lassen. Dies ist möglich, da ja die *hotspots* die Zentren von Treffern ausmachen. Für die Umsetzung einer zweiten Idee ist die Realisierung nicht sofort offensichtlich. Dies ist *Preprocessing*, d. h. ein vorbereitendes Rechnen, noch ehe der erste Sequenzvergleich ausgeführt wird. Sie werden beim Studium des BLAST-Algorithmus ein elegantes *Preprocessing*-Verfahren kennenlernen. Diese vorbereitenden Arbeiten resultieren in Indexdateien, die bei jeder Veränderung der Sequenzdatenbank aktualisiert werden müssen. Beachten Sie auch, dass die für das *Preprocessing* verwendeten Indizes von der Wahl der Scoring-Matrix abhängen. Damit ist klar, dass für jedes der zu BLAST angebotenen Scoring-Schemata ein separater Index gehalten werden muss.

Die beiden, im Basistext vorgestellten Heuristiken werden für die gleichen Zwecke eingesetzt; keiner der beiden Algorithmen zeichnet sich durch besondere Eigenschaften gegenüber dem anderen aus. Achten Sie in der FASTA-Ausgabe (Abbildung 12.2, Teil ①) auf den Verlauf des Histogramms mit Score-Werten. Diese Verteilung hat eine typische Form und kommt in der Bioinformatik öfters vor: Es ist eine Extremwertverteilung, die bei der Vorstellung des BLAST-Algorithmus genauer untersucht wird. Sie ergibt sich dann, wenn die Wahrscheinlichkeit für das Auftreten von Werten größer einer Schwelle betrachtet wird.

Es hat sich durchgesetzt, Treffer nicht mehr nach dem Score, sondern im Hinblick auf den Erwartungswert zu klassifizieren. Machen Sie sich die Bedeutung dieses Parameters klar: Er entscheidet beispielsweise in Sequenzierprojekten darüber, welche Funktion einem Genprodukt zugewiesen wird!

4.4 Profilbasierte Methoden des Sequenzvergleichs

Bei vielen bioinformatischen Methoden konnte eine Steigerung der Empfindlichkeit oder der Vorhersagequalität erreicht werden, indem eine Einzelsequenz durch ein Profil ersetzt wurde. Dies gilt sowohl für den Sequenzvergleich als auch z. B. für die Sekundärstrukturvorhersage, wie wir später noch sehen werden. In all diesen Fällen ist die Begründung dieselbe: Durch ein Profil werden die Ansprüche, die an den jeweiligen Positionen an die Aminosäurenreste gestellt werden, präziser beschrieben, als dies in einer einzelnen Sequenz möglich ist. Diese Annahme wird durch den Vergleich von BLAST, PSI-BLAST und DELTA-BLAST belegt: Das iterative Verfahren PSI-BLAST besitzt im Vergleich zu BLAST bereits eine höhere Empfindlichkeit, die von DELTA-BLAST noch

übertroffen wird. Beachtenswert ist das statistische Verfahren zur Berechnung der Profile in PSI-BLAST. Bitte machen Sie sich die Problematik deutlich, die mit kleinen Stichprobenumfängen einhergeht, und studieren Sie die Lösung per Verbundwahrscheinlichkeit.

4.5 Multiple Sequenzalignments

Der folgende Befund wurde schon bei der Vorstellung profilbasierter Sequenzvergleichsmethoden erwähnt: Ein multiples Sequenzalignment (MSA) charakterisiert eine Proteinfamilie oder eine Faltungstopologie wesentlich präziser als eine einzelne Sequenz. Führen Sie sich bitte das zusätzliche Potential von MSAs vor Augen: Ist die Anzahl alignierter Sequenzen hinreichend groß, kann Statistik zum Vorkommen der Residuen betrieben werden!

Allerdings sind die Probleme, die beim multiplen Sequenzalignment zu lösen sind, auch wesentlich komplexer als die von paarweisen Alignments. So ist bereits das Berechnen eines Gesamtscores für ein MSA extrem aufwendig. Alle Algorithmen, die zum Berechnen von MSAs in der Praxis eingesetzt werden, sind Heuristiken, beispielsweise *divide-and-conquer*-Verfahren. Wir betrachten hier zunächst ein einfaches Programm, das in den letzten Jahren erheblich verbessert wurde: ClustalW. Im Basistext wird auf die speziellen Eigenheiten dieses Verfahrens hingewiesen: Wie so häufig verderben lokale Minima manchmal die Lösung.

Die Nachteile des *greedy*-Ansatzes von ClustalW sind offensichtlich. Elegant haben die Entwickler die anstehenden Probleme in T-Coffee durch das Einführen der *erweiterten Bibliothek* gelöst. Machen Sie sich bitte anhand von Abbildung 13.5 klar, dass mit der erweiterten Bibliothek für jede Anwendung eine Matrix errechnet wird, die das Verwenden der üblichen Substitutionsmatrizen überflüssig macht. Konsequenterweise zählt T-Coffee zu den besten der eingeführten MSA-Algorithmen. Am Beispiel einer weiteren Variante, M-Coffee genannt, lernen Sie zum ersten Mal eine Technik kennen, die in der Bioinformatik häufiger genutzt wird: Zur Lösung eines Problems werden mehrere Alternativverfahren parallel angestoßen und die finale Lösung wird aus mehr oder weniger stark variierenden Ergebnissen per Jury- oder Mehrheitsentscheidung errechnet. Aufgrund der großen Bedeutung von MSAs werden ständig verbesserte Algorithmen vorgestellt. Da die Menge zu analysierender Sequenzen permanent wächst, ist neben der Qualität der MSAs vor allem die Laufzeit ein wichtiges Qualitätskriterium.

4.6 Verwendung von MSAs zur Vorhersage wichtiger Residuen

Wir werden in den folgenden Kurseinheiten Verfahren kennenlernen, bei denen MSAs als Eingabe dienen. Was interessiert bei der Analyse der multiplen Alignments? Üblicherweise werden solche Sequenzen in ein MSA aufgenommen, die *homolog* sind oder eine gemeinsame Funktion besitzen. Bei Enzymen wird diese Funktion häufig durch wenige Seitenketten im aktiven Zentrum umgesetzt. Welche Seitenketten (d. h. Residuen) sind wichtig? Kritische Residuen werden in

der Regel durch evolutionäre Vorgänge **nicht** verändert. Somit deutet strikte Konservierung auf eine bedeutende Rolle der Residuen hin. Welche Bereiche eines Proteins sind variabel und welche werden durch zusätzliche Schleifen ergänzt? Auch diese Fragen können durch die Analyse eines MSAs beantwortet werden: Variabilität einzelner Positionen lässt sich z. B. mit Sequenz-Logos untersuchen; in den MSAs eingeführte Lücken sind in solchen Bereichen häufig, die strukturell variabel sind. Präziser wird die Konserviertheit von Residuen durch Begriffe aus der Shannonschen Informationstheorie (Gleichungen (13.6) und (13.8) des Basistextes) gefasst, die dazu dienen, wichtige Residuen vorherzusagen. Modernste Methoden bewerten gleichzeitig auch die Konserviertheit benachbarter Positionen, wie *FRpred* belegt; siehe Gleichung (13.9). Am Beispiel von *SDPpred* können Sie studieren, wie mit solchen Konzepten diejenigen Residuen identifiziert werden, die ganz spezifisch die Funktion von Enzymen festlegen. Hierfür kommt in Form der Transinformation (Gleichung (13.11)) wiederum die Shannonsche Informationstheorie zum Zuge.

4.7 Selbsttestaufgaben

Aufgabe 4.1:

Gegeben seien das folgende MSA von fünf Erkennungssequenzen und die daraus abgeleitete unvollständige Scoring-Matrix. Die Scores wurden mithilfe der log10-Funktion errechnet.

```
MSA
1234
AATG
ACCG
GGAA
CTAC
TTGT
```

	1	2	3	4
A	0.125	-0.18	0.125	?
C	0	0	0	?
G	-0.10	-0.10	-0.10	?
T	-0.10	+0.20	-0.10	?

Welchen Score erreicht die Sequenz ATTG im Vergleich mit diesem Motiv?

Aufgabe 4.2:

Der erste Schritt im BLAST-Algorithmus ist das Identifizieren von signifikanten *w*-mers. Vollziehen Sie mit Papier und Bleistift diesen Schritt nach.

Geben sei die **Querysequenz** *A* = HFWK, die mit der **Datenbanksequenz** *B* = HYFCWR verglichen wird.

Wo kommen in *B* *w*-mers der Länge 3 vor, die einen Score $T > 10$ aufweisen? Liegen Sie auf der gleichen Diagonalen? Benutzen Sie für die Berechnung die BLOSUM62-Matrix.

Aufgabe 4.3:

Gegeben sei ein Index, der für jedes Teilwort *TW* die Positionen aller *w*-mers in allen Sequenzen der Datenbank mit $T > cut\_off$ angibt. Skizzieren Sie, wie analog zum Vorgehen von BLAST rein durch Bewertung der Indizes ein *HSP* identifiziert werden kann. Beschreiben Sie Ihr Vorgehen an zwei Teilworten TW_k und TW_l aus der Eingabesequenz *A*.