Till Biskup

Bioinformatik

Eine Einführung für Biologen

STB Skripte

1. Auflage

Till Biskup

Bioinformatik

Eine Einführung für Biologen

1. Auflage

Bioinformatik

1. Auflage

Version 0.1.7
19. Oktober 2003

gesetzt mit LaTeX 2ε
unter Verwendung von MakeIndex, GlossTeX und BibTeX
Formeln mit $\mathcal{AMS}$–LaTeX

Kontakt zum Autor:

email: till@till-biskup.de
Homepage: http://www.till-biskup.de/

Vorwort

blabla

Till Biskup
Berlin, im Oktober 2003

URL Aktuelle Informationen und Links zu diesem Skript im Internet unter
http://www.till-biskup.de/bioinformatik/

Hinweise zur Benutzung

Der besseren Übersichtlichkeit und des schnellen Zugriffs auf die Informationen halber wurden drei Symbole im ganzen Skript immer wieder verwendet:

▲ **Begriffe, Regeln, Sätze** erscheinen mit einem kleinen vorangestellten Dreieck. Die weitere Beschreibung erscheint eingerückt und setzt sich damit deutlich vom umgebenden Text ab.

■ **Beispiele** und praktische Anwendungen sind durch ein vorangestelltes Quadrat gekennzeichnet. Sie zeigen den Bezug zur Praxis auf und stammen häufig direkt aus der zur Vorlesung gehörenden Computerübung.

➤ Hinweise auf wichtige **Fehlerquellen**, Tips und Querverweise werden durch einen Pfeil kenntlich gemacht.

Literaturempfehlungen

Drei Bücher sind sehr zu empfehlen (und wurden sowohl von Prof. Herzel als auch von Johannes Schuchhardt empfohlen):

Baldi P, Brunak S (2001) Bioinformatics, Adaptive Computation and Machine Learning, MIT Press, Massachusetts, 2. Aufl

Durbin R, Eddy SR, Krogh A, Mitchison G (1998) Biological sequence analysis, Cambridge University Press, Cambridge

Li WH (1997) Molecular Evolution, Sinauer Associates, Sunderland

Darüber hinaus existiert im Internet eine Menge guten Materials, vornehmlich in englischer Sprache. Für weitere Informationen dazu siehe den Link zum Skript.

Inhaltsverzeichnis

Kapitel 0

Einführung, Genome, Datenbanken

Our descendants will certainly not say that biology began with today's genome projects, but they may well recognize that a great acceleration in the accumulation of biological knowledge began in our era.

Durbin et al. [21]

0.1 Was ist Bioinformatik?

Layne Watson, professor in mathematics and computer science, defines bioinformatics as a tool for three levels of problem solving – managing data, sequence analysis, and inferring function.

1

The first level, he says, is "just getting the data and having confidence in it. Presently, the experiments which decode or sequence segments of a plant or animal genome are not entirely reproducible. That is, scientists rarely get exact sequences when they process DNA-sequence gel images. Having better algorithms for evaluating gel images will provide more precise results."

Clark Tibbetts, associate director of the new Virginia Bioinformatics Institute (VBI) agrees. "Bioinformatics is most effective when the computational experts are involved in how the data are collected, as well as the design of the analysis," he says. "The VBI will operate as a partner in data creation and use."

2

"The second level of problem solving for bioinformatics is where much of the work is being done now," Watson says. "You have fragmentary and conflicting data and you must construct and extract the important parts of the genetic sequence. This sequence analysis requires sophisticated discrete algorithms to search for patterns."

Heath explains, "When biologists find a particular genetic sequence, they can look in the database for a match or for something similar, which might have an

evolutionary relationship or a functional relationship. That gives the biologists clues that certain functions are performed by certain genes."

Plant scientist M.A. Saghai Maroof used DNA similarity between two disease resistance genes from tobacco and Arabidopsis to develop a general technique for the identification and isolation of new disease resistance genes from soybean and other crops. His discoveries in molecular marker technology have advanced plant science, and he and colleagues at Virginia Tech have developed improved varieties of corn and soybeans.

"We want to be able to explore questions regarding the specific functions of certain genes," says Heath. For example, Malcolm Potts, professor of biochemistry, and Richard Helm, professor of wood science, are working on the genomes of several cyanobacteria. These organisms are capable of surviving under extreme environmental stress. The team is identifying the genes central to stabilization in order to engineer this trait into other organisms' tissue and cell types.

Another second-level bioinformatics task is piecing information together. "Chromosomes are so large that experimental work is done by chopping them into pieces small enough to determine DNA sequences," Heath says. "Computational work is required to put the pieces back together – to join pieces of a few hundred genetic sequences into DNA code made up of millions of sequences."

3

"The third level of bioinformatics problem solving is to try to infer function," says Watson. "At this stage the algorithms and mathematics are the most sophisticated, requiring both discrete and continuous algorithms to make the connection between the genetic sequence and the biological function, and to create models that describe and predict interactions at the level of the cell cycle.

"The difference between the second and third level is that at the second level a scientist may observe that disease occurs when a specific gene is missing or flawed. At the third level, we have an explanation of how the missing or flawed gene results in a physiological phenomenon.

"Treatment is still possible at the second level. We may know that a certain substance prevents or eases the symptoms of Parkinson's disease without knowing why, for instance. But a lot more can be done once we know how substances work – how and why genes are responding," Watson says. [20]

0.2 Genome

Unlike the word "genome" which was coined just after the First World War by the German botanist Hans Winkler [93, 6], the word "proteome" entered the scientific literature recently, in 1994 papers by Mark Wilkins and Keith Williams [92].

Organismus	Genomgröße	Proteine	codierend (%)	GC (%)	Knockouts (lethal)
Retroviren	≈ 10 kb	470	≈ 90		
Mycoplasma genitalium	0.58 Mb	470 ORFs	88	32	
Bac. subtilis	4.2 Mb	4100 ORFs	87	43.5	
E. coli	4.6 Mb				1800
Sacc. cerevisiae	12 Mb	5800...6300	70		3600
C. elegans	97 Mb	16...19.000			
D. melanogaster	165 Mb	12.000			3100
A. thaliana	70-145 Mb	25.000			
H. Sapiens	3300 Mb	35.000	1–2	42	

Die Sequenzierung kompletter Genome ist immer mit Zerstückelung verbunden. Die Zusammensetzung erfolgt im Rechner, bewegt sich allerdings an der Grenze der Zuverlässigkeit

0.3 Proteome

Der Counterpart der Genomanalyse — die Analyse kompletter Genome — auf dem Proteinlevel ist die *Proteom*–Analyse [299,413]. Proteome enthalten die komplette Protein–Expression eines Sets von Chromosomen. In einem mehrzelligen Organismus unterscheidet sich dieses Set von Proteinen von Zelltyp zu Zelltyp und verändert sich außerdem über die Zeit als Folge der Kontrolle der Entwicklung aus dem Embryo zum ausgewachsenen Organismus über die Genregulation. Proteom–Forschung beschäftigt sich mit den Proteinen, die von den Genen eines gegebenen Genoms produziert werden.

Im Gegensatz zu dem Begriff "*Genom*", der kurz nach dem ersten Weltkrieg vom deutschen Botaniker Hans Winkler [93, 6] geprägt wurde, ist der Begriff "*Proteom*" vergleichsweise jung: Er taucht zum ersten mal in Papers von Mark Wilkins and Keith Williams [92] in der wissenschaftlichen Literatur auf.

Proteom–Analyse beschäftigt sich nicht ausschließlich mit der Bestimmung von Sequenz, Lokation und Funktion von Protein–codierenden Genen, sondern ist genauso eng befaßt mit der Erforschung des präzisen biochemischen Zustandes jedes Proteins in seiner posttranslationalen Form. Diese aktiven und funktionalen Formen der Proteine wurden in einigen Fällen erfolgreich durch die Nutzung von *machine–learning techniques* vorhergesagt. [1, p. 16]

0.4 Datenbanken

Vollständige Genome

- NCBI
- TIGR
- MagPie

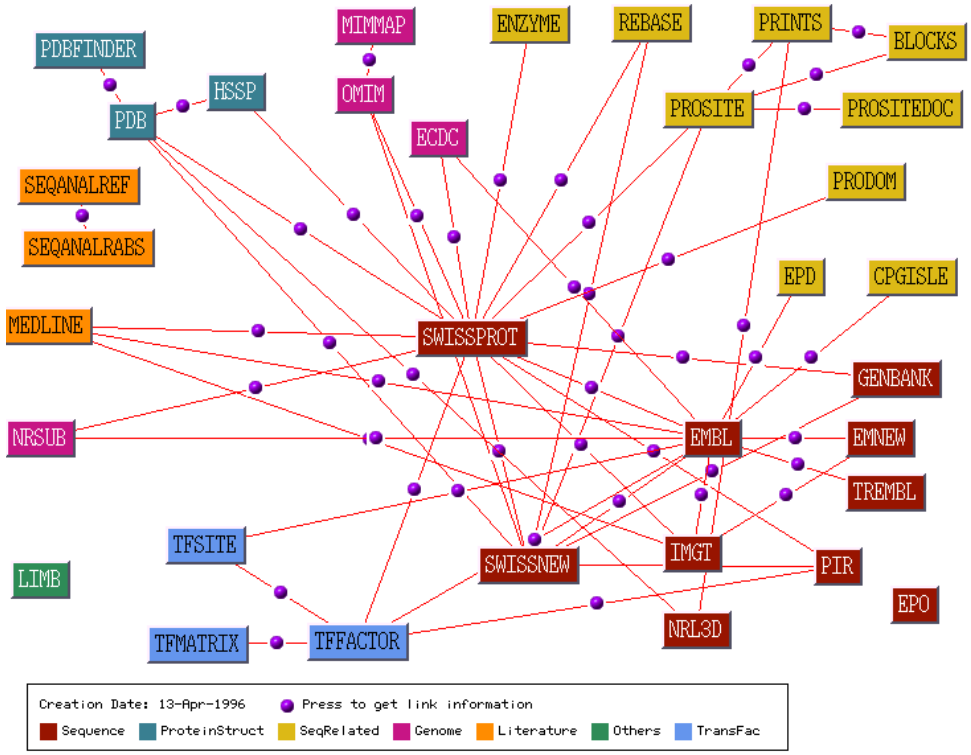

Abb. 0.1: Übersicht über einen Teil der im WWW verfügbaren Datenbanken für verschiedene Daten der Molekularbiologie.

DNA-Datenanken (mit Annotation)

- Genbank (NCBI)
- EMBL
- PUMA (Stoffwechselwege)
- TransFac (Regulatorische Elemente)

Protein-Datenbanken (m. Annot.)

- SwissProt
- EMBL
- PIR
- Blocks/Prosite (Funktionelle Blöcke)
- PDB (Protein–3D–Struktur)
- CATH, SCOP (Protein–Familien)

Kapitel 1

Genidentifikation bei Prokaryoten, Sequenzstatistik

Mit der zunehmenden Automatisierung der Sequenzierung ganzer Genome hat die Menge von Rohsequenzen rapide zugenommen (vgl. auch Abb. 1.1). Eine Hauptaufgabe der Bioinformatik ist es hier, mittels geeigneter Algorithmen aus den Rohdaten Informationen zu gewinnen.

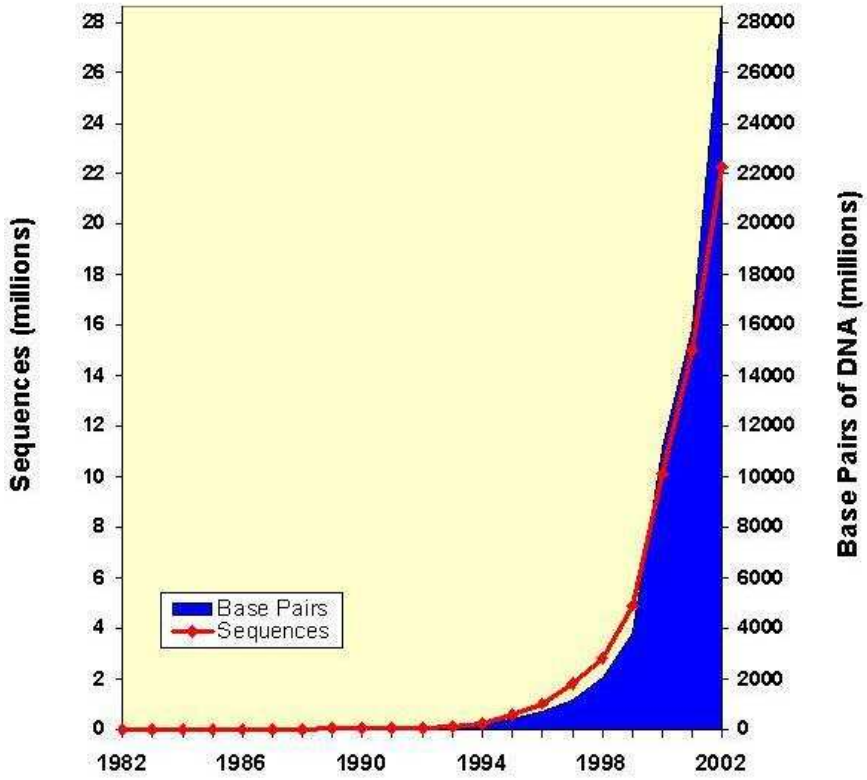

Abb.1.1: Wachstum der GenBank® Datenbank. GenBank® ist die Datenbank des NIH (National Institutes of Health) für Gensequenzen, eine annotierte (*annotated*) Sammlung aller öffentlich zugänglicher DNA–Sequenzen (Nucleic Acids Research 2003 Jan 1;31(1):23-7). Mit dem Stand Januar 2003 umfaßt sie ca. 28,507,990,166 Basen in 22,318,883 Sequenz–Records.

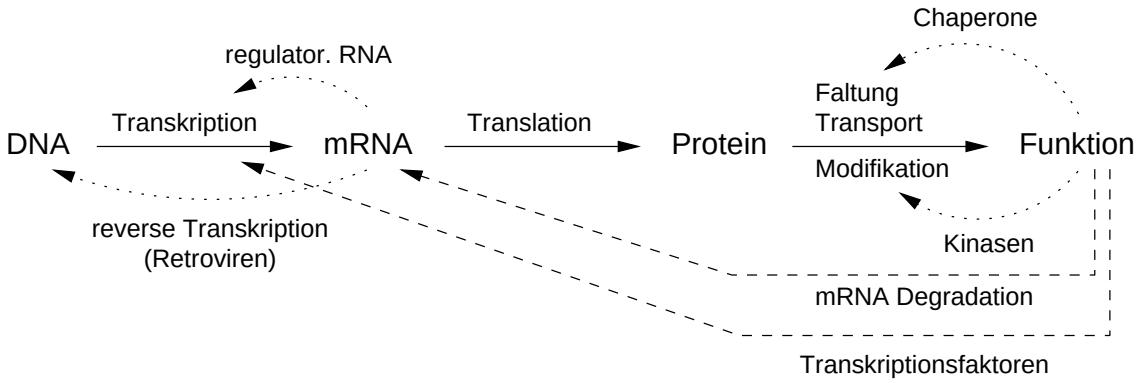

Abb. 1.2: Schema der Realisierung der auf den Genen codierten Information. Dem Schema zugrunde liegt das Zentrale Dogma der Molekularbiologie (*Central Dogma*, Crick [17]), daß der Informationsfluß immer von der DNA über die RNA zum Protein hin gerichtet ist. Erst mit der Entdeckung der Reversen Transkriptase 1970 [91, 2] wurde die Grundlage für heute in der Genomforschung wichtige Verfahren (z.B. cDNA) gelegt.

Für die Genidentifikation (*gene finding*) ist es notwendig, eine ganze Reihe verschiedener Signale einzubeziehen [1]: Promoter–Regionen, Sequenzen für den Translationsstart und Stop, Leseraster–Periodizitäten (*reading frame periodicities*), Polyadenylierungssignale und bei Eukaryoten zusätzlich einige Merkmale, auf die hier nicht eingegangen werden soll.

Prinzipiell gibt es drei verschiedene Ansätze für die Genidentifikation (vgl. Abb. 1.3):

- **search by signal** (Kap. 2)
  Auffinden kurzer Sequenzmotive, die mit Genen im Zusammenhang stehen, z.B. Promoter– und Transkriptionsfaktor–Bindungsstellen oder Splice Sites.

- **search by content** (Kap. 4)
  Auffinden von ORFs (*open reading frames*, offene Leseraster), Regionen mit GC–Gehalt und *codon usage characteristic* codierender Bereiche etc.

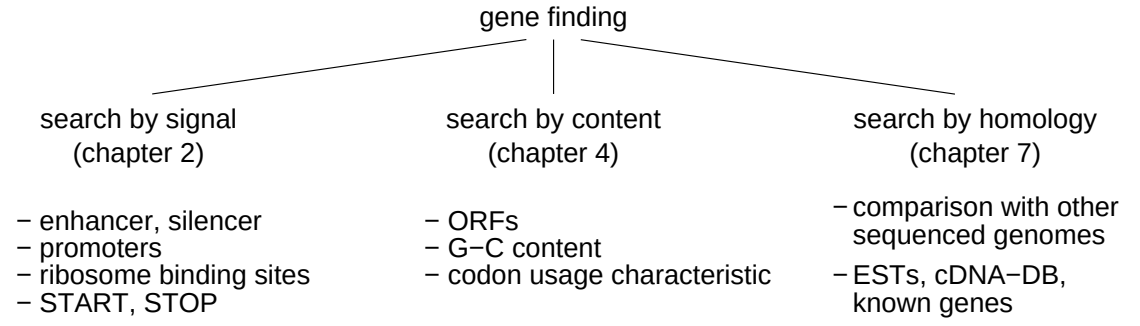

Abb. 1.3: Unterschiedliche Verfahren der Genidentifikation (*gene finding*), wie sie der Reihe nach in den folgenden Kapiteln behandelt werden. *search by signal* Auffinden kurzer Sequenzmotive, die mit Genen im Zusammenhang stehen, z.B. Promoter– und Transkriptionsfaktor–Bindungsstellen oder Splice Sites. *search by content* Auffinden von ORFs (*open reading frames*, offene Leseraster), Regionen mit GC–Gehalt und *codon usage characteristic* codierender Bereiche etc. *search by homology* Nutzung der Sequenz eines bekannten Gens eines anderen Organismus zur Identifizierung von Homologen; Aligning von EST–Sequenzen an genomische Sequenzen.

- **search by homology** (Kap. 7)
  Nutzung der Sequenz eines bekannten Gens eines anderen Organismus zur Identifizierung von Homologen; Aligning von EST–Sequenzen an genomische Sequenzen.

Heute nutzen die erfolgreichsten Genidentifikationsmethoden *machine learning*–Techniken wie Neuronale Netze (*neural nets*), Entscheidungsbäume (*decision trees*) und Hidden Markov Modelle (*hidden Markov models*, HMM), um die Information all dieser "traditionellen" Ansätze für eine Vorhersage auszuwerten. Diese Programme werden über Datenbanken mit den Daten bekannter Gene trainiert. Deshalb können die Vorhersagen für Organismen, die nicht in dieser Datenbank repräsentiert sind, ungenauer ausfallen.[1]

- Unterschied von EST[2] zu cDNA
  - meist kürzer
  - schlampig sequenziert (nur 1x, cDNA ca. 5x)
  - Vorteil: Menge
    → effektive Nutzung bei Homologie–Vergleichen

1.1 Genidentifikation bei Prokaryoten

1.1.1 Besonderheiten prokaryotischer Genome

1.1.2 Methoden prokaryotischer Genidentifikation

Bei Prokaryoten ist die Genidentifikation wesentlich einfacher als bei Eukaryoten, da im prokaryotischen Genom codierende Abschnitte nicht durch Introns unterbrochen werden. Trotzdem ist es insbesondere für relativ kurze ORFs (*open reading frames*, offene Leseraster) nicht einfach zu entscheiden, ob es sich um ein Gen handelt oder eine zufällige Sequenz.

Ein sehr erfolgreiches Genidenitifikationsprogramm (*gene finder*) ist GeneMark™ vom Georgia Institute of Technology[3] [12, 14, 13, 58]. Seine Genauigkeit erreicht ca. 99%[4]. Ein Schlüsselmerkmal dieses Programms ist, den "Schatten" einer wahren codierenden Region (*true coding region*) auf dem nichtcodierenden Strang (*noncoding strand*) auf intelligente Weise zu bestimmen. Dieser Mechanismus trägt wesentlich zur hohen Leistung und Geschwindigkeit des Programms bei. [1] Genemark™ verwendet dazu ein Hidden Markov–Modell (*hidden Markov model*, HMM).

Die Erkennung regulatorischer Sequenzen ist nach wie vor sehr schwierig. So wurden z.B. nur für einige 100 der ca. 4000 Gene von *Escherichia coli* TATA–Boxen gefunden.

[1] vgl. `http://bioweb.cgb.indiana.edu/seqanal/genes/intro.html`

[2] Expressed sequence tags (ESTs) are short (200-500bp) DNA sequences generated from the 3' and 5' ends of randomly selected cDNA clones. The purpose of EST sequencing is to rapidly scan for all the protein coding genes and to provide a tag for each gene on the genome.

[3] GeneMark™ A family of gene prediction programs provided by Mark Borodovsky's Bioinformatics Group at the Georgia Institute of Technology, Atlanta, Georgia.

[4] Zum Vergleich: Die Trefferquote für korrekte Genvorhersagen *in silico* liegt bei eukaryotischen Genen bei $\leq 50\%$, vgl. Kap. 4, S. 27

➤ Die Aufklärung der Genregulation ist nach wie vor eines der größten ungelösten Probleme.

Frishman et al. [34]

"The principial goal of large–scale genome sequencing is to obtain new insights into physiological and biochemical processes in living organisms. An essential step in this process is gene identification with subsequent computer–based annotaion of the corresponding gene products. Although bacterial genomic sequences are devoid of introns, gene recognition in bacteria is far from being simple. It is easy to extract all possible open reading frames (ORFs) from a given DNA sequence; it is much less trivial to decide which of them correspond tho genes that are actually expressed and code for proteins. The following features are important indicators of protein coding regions in DNA: (i) sufficient ORF length. Long ORFs rarely occur by chance; (ii) specific patterns of codon usage that are different from triplet frequencies in non–coding regions ('coding potential'); (iii) the presence of ribosome binding sites (RBS) in the $(-20)\ldots(-1)$ region upstream of the start codon that help to direct ribosomes to the correct translation start positions [1]. A part of the RBS is formed by the purine–rich Shine–Dalgarno (SD) sequence which is complementary to the 3' end of the 16S mRNA [2]; (iv) similarity to known, especially experimentally characterized, gene products."

1.2 Sequenzstatistik I (Häufigkeiten)

Ein großer Teil der Bioinformatik widmet sich der Analyse von Genom– und Protein–Sequenzen. Folgende Fragen werden dabei unter anderem bearbeitet:

- Was ist der mittlere Abstand von Restriktionsenzym–Bindungsstellen?
  (z.B. GTT'AAC von Hinc II)
- Wie groß ist der mittlere Abstand von Konsensus–Sequenzen, z.B. Stop–Codons, TATA–Boxen?
- Wie viele solcher "Wörter" gibt es in einer Sequenz der Länge N? Wie groß ist die Wahrscheinlichkeit, daß ein Motiv (z.B. ATG, AGGAGG, ...) zufällig auftritt?

Entscheidend ist vor allem, in einer gegebenen Sequenz zufälliges Auftreten bestimmter Teilsequenzen von relevanter Information zu unterscheiden. Daraus ergibt sich eine **Definition der Bioinformatik**:

➤ Die Bioinformatik versucht, biologisch relevante Signale von "Falschpositiven" zu unterscheiden.

Ein wichtiger Grund für den Erfolg der Bioinformatik insbesondere auf dem Gebiet der Sequenzanalyse beruht auf der Anwendung statistischer Methoden, deren Grundlagen vorher in anderen Fachgebieten (insbesondere der Spracherkennung, *voice recognition*) entwickelt wurden. Daher sollen zu Anfang einige wichtige statistische Konzepte eingeführt werden, ohne konkrete Anwendungen in Form von Beispielen aus dem Auge zu verlieren.

1.2.1 Bernoulli–Sequenzen

Bernoulli–Sequenzen sind Folgen von λ statistisch unabhängigen Symbolen (z.B. Lottozahlen, DNA ohne Selektionsdruck, zufällige Peptide, Oligos[5]) mit den Wahrscheinlichkeiten p_i für jedes Symbol $i = 1, 2, \ldots, \lambda$. Im Modell sind die p_i *a priori* gegeben, in realen Systemen müssen sie dagegen geschätzt werden:

$$\hat{p}_i = \frac{N_i}{N} \tag{1.1}$$

mit der Zahl N_i des Auftretens des Symbols i und der Gesamtzahl N aller Symbole.

▲ **statistische Unabhängigkeit** Zwei Ereignisse i und j sind statistisch unabhängig, wenn sich ihre Verbundwahrscheinlichkeit $p(i, j)$ aus dem Produkt der Einzelwahrscheinlichkeiten ergibt:

$$p(i, j) = p(i) \cdot p(j) \tag{1.2}$$

Entsprechend gilt für ein Ereignis–Tiplett i, j, k

$$p_{ijk} = p_i \cdot p_j \cdot p_k$$

Auch wenn es auf den ersten Blick widersinnig erscheint und das Genom sehrwohl Information enthält, gilt als erste Näherung in der Bioinformatik:

➤ Das (humane) Genom wird als gewürfelte Sequenz (Bernoulli–Sequenz) approximiert.

■ **Häufigkeit einer TATA–Box** Das humane Genom wird als gewürfelte Sequenz approximiert. Damit können die einzelnen Basen einer Sequenz als statistisch unabhängig angesehen werden und es gilt:

$$p(\texttt{TATA}) = p(\texttt{A}) \cdot p(\texttt{C}) \cdot p(\texttt{G}) \cdot p(\texttt{T}) = \frac{1}{256}$$

Die typische Genlänge beträgt 1000 bp. Das bedeutet, daß im Schnitt nur jeder 4. Treffer für eine TATA–Box richtig ist. Weiter kommt hinzu, daß nur jedes zweite Gen mit einem Promoter (und damit einer TATA–Box) ausgestattet ist: Damit ist nur noch jeder 8. Treffer richtig.[6]

Der GC–Gehalt des Genoms liegt selten bei 50%. In einem **AT–reichen** Genom verschiebt sich die Wahrscheinlichkeit für die Sequenzfolge TATA also zusätzlich:

$$p(\texttt{A}) = p(\texttt{T}) = 0.35$$
$$p(\texttt{G}) = p(\texttt{C}) = 0.15$$

$$p(\texttt{TATA}) = 0.35^4 = 0.15 \approx \frac{1}{67}$$

[5] Oligos: Kurzform für Oligonucleotide *oligonucleotides*, kurze Nucleotid–Sequenzen von DNA oder RNA, typischerweise mit 20 oder weniger Basenpaaren.

[6] Nicht eingerechet ist hier, daß es sehrwohl noch Unterschiede in der genauen Sequenzfolge der TATA–Boxen gibt, die zusätzliche Variabilität bedeuten.

Das bedeutet, daß in einem solchen AT–reichen Genomabschnitt im Schnitt alle 67 bp eine TATA–Box zufällig auftritt.

➤ Die Gleichsetzungen $A = T$ und $G = C$ sind nur eine Approximation, bei Eubakterien gibt es Unterschiede zwischen leading– und lagging strand (sogenannter *GC–skew*).

Promoterbereiche sind meist AT–reich, da sie sich so leichter aufschmelzen lassen (die AT–Bindung wird durch zwei Wasserstoffbrücken realisiert, die GC–Bindung dagegen durch drei).

Das Startcodon bei Prokaryoten ist nicht perfekt: Neben ATG kommen auch GTG und TTG vor.

■ **Übungsaufgabe** Gegeben sei die Promotersequenz TATAAT und eine Häufigkeit aller vier Basen von $p_i = \frac{1}{4}$. Wie viele TATAAT–Sequenzen kommen in 2^{10} bp vor?

$$p(\texttt{TATAAT}) = \left(\frac{1}{4}\right)^6 = \frac{1}{4096}$$

1.2.2 Binomial–Verteilung

Wie schon erwähnt ist eine wesentliche Aufgabe der Bioinformatik bei der Sequenzanalyse die Unterscheidung biologisch relevanter Signale von Falschpositiven. Dafür ist es wichtig, zu wissen, wie häufig ein Signal rein statistisch ist. Das führt zu der Frage:

- Wie häufig taucht eine Symbolsequenz (Schnittstelle, Motiv, STOP) in einer Sequenz der Länge N auf?

Annahme: In der Sequenz der Länge N existieren N unabhängige Wörter der Länge L und der Wortwahrscheinlichkeit p. In der Praxis hat diese Annahme einige Probleme:

- In einer Sequenz der Länge N gibt es nur $N - (L + 1)$ unabhängige Wörter.
- Die Wörter sind nicht unabhängig.

Lassen wir diese Probleme unbeachtet, führt das zur Binomialverteilung.

▲ **Binomialverteilung** Gegeben sei eine Sequenz der Länge N, in der N unabhängige Wörter der Länge L und der Wortwahrscheinlichkeit p existieren. Dann ist die Wahrscheinlichkeit $W(k)$, daß k aus N Wörtern gefunden werden:

$$W(k) = \binom{N}{k} p^k (1-p)^{N-k} \tag{1.3}$$

mit dem **Binomialkoeffizienten**

$$\binom{N}{k} = \frac{N!}{(N-k)! \cdot k!} \qquad \text{für } N \geq k \tag{1.4}$$

Die Binomialverteilung hat den Mittelwert

$$\langle k \rangle \equiv \bar{k} \equiv E(k) = p \cdot N \tag{1.5}$$

und die Varianz (Streuung)

$$\begin{aligned} \mathrm{Var}(k) \equiv \sigma^2 &= \langle (k - \langle k \rangle)^2 \rangle = \langle k^2 \rangle - \langle k \rangle^2 \\ &= p(1-p)N \end{aligned} \tag{1.6}$$

Für $p \ll 1$ kann die Varianz vereinfacht geschrieben werden als

$$\sigma^2 \approx pN \tag{1.7}$$

und ist damit für diesen Fall identisch mit dem Mittelwert. Die Standardabweichung ist die Wurzel der Varianz:

$$\sigma = \sqrt{\sigma^2} \tag{1.8}$$

Die Binomialverteilung $W(k)$ ist normiert, d.h.

$$\sum_{k=0}^{N} W(k) = 1 \tag{1.9}$$

Daraus ergibt sich unter Ausnutzung der Beziehung $1^N = (1 - p + p)^N$ mit $q = (1 - p)$ der **Binomische Satz**

$$(p+q)^N = \sum_{k=0}^{N} \binom{N}{k} p^k q^{N-k} \tag{1.10}$$

Bemerkungen

- Für seltene Ereignisse ($p \to 0$) und großen Umfang der Stichprobe ($N \to \infty$) geht die Binomialverteilung in eine Poisson–Verteilung über. Für $p \leq 0.08, n \geq 1500p$ kann die Binomialverteilung mit im allgemeinen ausreichender Genauigkeitdurch die Poisson–Verteilung ersetzt werden, deren Auswertung einfacher ist. [15]
- Für ($N \to \infty, pN$ endlich) geht die (stetige) Binomialverteilung in die (kontinuierliche) Gauß–Verteilung über.

■ **Häufigkeit einer TATA–Box** Gegeben sei eine Sequenzlänge $N = 1000$ und eine Wahrscheinlichkeit $p = \frac{1}{256}$ (z.B. $p(\mathrm{TATA})$) eines Wortes der Länge $L = 4$. Unter der Berücksichtigung, daß es in einer Sequenz der Länge N nur $N - (L + 1)$ unabhängige Wörter gibt, ergibt die Binomialverteilung

$$\langle k \rangle = 3.9 \qquad\qquad \sigma^2 \approx 3.9$$

Man findet also (rein statistisch!) etwa 4 ± 2 ($\langle k \rangle \pm \sigma$) Sequenzen TATA in einer Sequenz der Länge $N = 1000$ bp.

Standardfehler Der Standardfehler (engl.: *standard error*) ist ein Maß für die Streuung einer Stichprobenstatistik über alle möglichen Zufallsstichproben vom Umfang N aus der Grundgesamtheit. Vereinfachend gesagt: Er ist ein Maß für die "durchschnittliche" Größe des Stichprobenfehlers der Stichprobenstatistik (z.B. des arithmetischen Mittels oder des Anteilswertes). Der Standardfehler einer Stichprobenstatistik hängt von verschiedenen Faktoren ab, je nachdem, um welche Statistik es sich handelt. Ganz allgemein kann man jedoch sagen, daß ein Standardfehler um so kleiner wird, je größer der Stichprobenumfang ist. Größere Zufallsstichproben erlauben präzisere Schätzungen, weil der Stichprobenfehler kleiner wird.[7]

$$\frac{\sigma}{\langle k \rangle} \approx \frac{\sqrt{pN}}{pN} = \frac{1}{\sqrt{pN}} \propto \frac{1}{\sqrt{N}}$$

➤ **Faustregel:** Die Schwankungsbreite ist oft die Wurzel aus den gezählten Ereignissen (bei unabhängigen Ereignissen!)

■ **Genidentifikation in einer DNA–Sequenz (Modell)** Die Wahrscheinlichkeit für jedes Nukleotid i sei gleich, $p_i = \frac{1}{4}$ und die Sequenz bestehe aus 100 unabhängigen Tripletts (300 bp). Wie viele STOP–Codons kommen zufällig in dieser Sequenz vor?

Jedes Triplett besteht aus drei Nukleotiden, die Wahrscheinlichkeit für ein bestimmtes Triplett ist also

$$p_{\text{Triplett}} = \left(\frac{1}{4}\right)^3 = \frac{1}{64}$$

Da es drei STOP–Codons (TGA, TAA, TAG) gibt, ist die Wahrscheinlichkeit für ein STOP–Codon

$$p_{\text{STOP}} = \frac{3}{64}$$

Für eine Sequenzlänge von $N = 100$ Tripletts (entspricht 300 bp) ist damit die Wahrscheinlichkeit $W(0)$, kein STOP–Codon zu enthalten (und damit ein ORF, *open reading frame*, zu sein)

$$\begin{aligned} W(0) &= \binom{N}{0} p^0 (1-p)^{100} \\ &= 1 \cdot 1 \left(\frac{61}{64}\right)^{100} = 0.008 = 0.8\% \end{aligned}$$

Für das humane Genom mit seinen $3.5 \cdot 10^9$ bp ergäbe sich nach diesem Schema eine Zahl von ORFs der Länge 300 bp (entsprechend 100 Tripletts) von

$$\frac{3.5 \cdot 10^9}{300} \cdot 0.008 \approx 93.000$$

[7] Quelle: `http://wwwhomes.uni-bielefeld.de/hjawww/glossar/node142.html`

Aber mit wie vielen STOP–Codons müssen wir durchschnittlich auf 100 Tripletts rechnen? Entsprechend der obigen Rechnung können die Wahrscheinlichkeiten $W(k)$ für k STOP–Codons in der Sequenz berechnet werden:

$$\begin{aligned} W(1) &= \binom{100}{1} \cdot \frac{3}{64} \cdot \left(\frac{61}{64}\right)^{99} = 4\% \\ W(2) &= \frac{100 \cdot 99}{1 \cdot 2} \cdot \left(\frac{3}{64}\right)^{2} \cdot \left(\frac{61}{64}\right)^{98} = 9.8\% \\ W(3) &= 15.8\% \quad W(4) = 18.9\% \\ W(5) &= 17.8\% \quad W(6) = 13.9\% \\ W(10) &= 1.2\% \end{aligned}$$

Einfacher ist allerdings, den Mittelwert und die Varianz (Standardabweichung) der Binomialverteilung zu verwenden:

$$\begin{aligned} \langle k \rangle &= pN \approx 4.7 \\ \sigma^2 &= p(1-p) \cdot N \approx 4.5 \Rightarrow \sigma \approx 2.1 \end{aligned}$$

Es sind ca. 5 ± 2 STOP–Codons in einer Sequenz von $N = 100$ unabhängigen Tripletts zu erwarten.

Kein STOP–Codon in einer Sequenz von 100 Tripletts zu finden, ist dagegen statistisch signifikant ($p < 0.01$). D.h. es handelt sich mit großer Wahrscheinlichkeit um einen ORF.

■ **TATA–Boxen in *Bacillus subtilis*** Die Genomgröße von *Bacillus subtilis* beträgt $N = 4.2$ Mb $= 4.2 \cdot 10^6$ bp, der GC–Gehalt 43%. Daraus folgt für die Wahrscheinlichkeit für eine TATA–Sequenz

$$p(\text{TATA}) = p(0.285)^4 \approx 0.0066$$

Der Erwartungswert der Zahl an TATA–Sequenzen im Genom von *Bac. subtilis* ist damit

$$\langle k \rangle = p \cdot N \approx 27\,700$$

mit einer Standardabweichung

$$\sigma = \sqrt{\sigma^2} \approx 170$$

Die tatsächliche Zahl der ORFs im Genom von *Bac. subtilis* beträgt dagegen nur 4 100, die in 1 400 Operons lokalisiert sind.

D.h., daß ca. zwanzigmal mehr TATA–Sequenzen zufällig auftreten als biologisch relevant sind. Daraus folgt eindeutig, daß die TATA–Sequenz (Konsensus–Sequenz) alleine nicht ausreicht, um ein Gen zu identifizieren. Aus diesem Grund verwenden moderne Genidentifikations–Algorithmen meist eine Kombination aus verschiedenen Ansätzen zur Genidentifikation.

Kapitel 2

Konsensussequenzen, Gewichtsmatrizen

Es gibt drei Möglichkeiten zur Identifizierung (prokaryotischer) Gene via Erkennung charakteristischer Signale (*search by signal*): Die *Konsensussequenzen* sind das bekannteste Beispiel; sie bringen aber auch eine Reihe Probleme mit sich, die zur Entwicklung verfeinerter Methoden geführt haben: Die *Häufigkeitstabellen* erlauben eine bessere Suche nach neuen Sequenzen und ermöglichen den Vergleich, die *Gewichte* (Logarithmen der Häufigkeiten) führen schließlich zu einer Quantifizierung des Informationsgehaltes einer Sequenz.

2.1 Konsensussequenzen

Beim Studium molekularer Bindungsstellen auf DNA und RNA ist es eine gängige Praxis, ein Alignment von Sequenzen mehrerer Bindungsstellen ein und desselben makromolekularen Bindungsfaktors zu erstellen und dann die verbreitetsten Basen an jeder Position auszuwählen, um daraus eine Konsensussequenz zu machen. [75]

- **Bekannte Konsensussequenzen** sind z.B. die σ^{70}–Promotoren[1] von *E. coli*, die -35 Sequenz und die -10 Box (TATA–Box, Pribnow box [59]), die Shine–Dalgarno–Sequenz des prokaryotischen Promoters, der Leucine Zipper [54] und Zinc Finger [4, 71, 66].

- **Shine–Dalgarno–Sequenz** Ribosomenbindungsstelle (RBS) bei Prokaryoten. Purinreiche Sequenz, die 8 bis 12 Nukleotide vor dem Startcodon ATG liegt. 1974 zunächst bei *E. coli* beschrieben [78]. Durch eine Basenkomplementarität der Shine–Dalgarno–Sequenz mit einem Bereich nahe des 3'–Endes der 16S–rRHA wird eine kurze Doppelstrangregion kurz vor dem Startcodon in der mRNA geformt. Diese RBS wird für eine korrekte Positionierung des Startcodons am Ribosom benötigt. Bei Eukaryoten ist die Shine–Dalgarno–Sequenz nicht vorhanden. Hier übernehmen Initiationsfaktoren funktionell deren Aufgabe. [43]

[1] Der hochgestellte Index (70) gibt das Molekulargewicht des σ–Faktors in kD an.

Tab. 2.1: Konsensussequenz für die TATA–Box (Promoter) von *E. coli*. Die Daten entsprechen real vorkommenden Sequenzen für TATA–Boxen. Charakteristisch ist ebenfalls, daß die eigentliche Konsensus–Sequenz gar nicht vorkommt.

	...	T	A	G	A	A	T	...
	...	T	A	T	C	A	T	...
	...	A	A	T	A	A	T	...
	...	T	A	T	A	G	T	...
	...	T	A	G	A	A	C	...
	...	T	T	T	A	A	T	...
Konsensussequenz:		T	A	T	A	A	T	

Historie Anhand der TATA–Box von *E. coli* läßt sich sehr gut die Entwicklung der Konsensussequenzen mit zunehmender Sequenzierung des Genoms zeigen: Je mehr Promotersequenzen bekannt wurden, desto mehr Sequenzen konnten beim *multiple alignment* für die Bestimmung der Konsensussequenz genutzt werden. Zu Beginn waren 6 Promoter bekannt, die Konsensussequenz lautete `TATRAT` (mit `R=G,A`).

Nachdem ≥ 100 Promoter bekannt waren, wurde eine erweiterte Konsensus–Sequenz mit Häufigkeiten erstellt: $\mathtt{T}_{80}\mathtt{A}_{95}\mathtt{T}_{45}\mathtt{A}_{60}\mathtt{A}_{50}\mathtt{T}_{95}$. Mittlerweile sind ≈ 3000 Promoter bekannt, z.T. handelt es sich dabei um mutmaßliche (*putative*) Promotersequenzen: Die gegenwärtige Konsensussequenz lautet `TAAAAT` und weicht damit vom bekannten `TATA` ab.

Probleme

- Konsensussequenz selbst eher selten
- nicht verläßlich bei der Suche nach neuen Bindungsstellen [75]
- abhängig von der Lernstichprobe
- *mismatches* schwer zu erfassen
- Willkür bei der Notation

  z.B.: 45% A, 20% C, 30% G, 5% T → A oder R oder non–T

Tab. 2.2: Relative Häufigkeiten (*relative frequencies*) der Basen für 242 Promotoren von *E. coli*. Angegeben sind die relativen Häufigkeiten des Vorkommens der vier Basen für jede Position in der Sequenz. Daten nach Staden [83].

`A`	0.04	0.88	0.26	0.59	0.49	0.03	
`C`	0.09	0.03	0.11	0.13	0.22	0.05	
`G`	0.07	0.01	0.12	0.16	0.12	0.02	
`T`	0.80	0.08	0.51	0.13	0.18	0.89	
Konsensus:	`T`	`A`	`T`	`A`	`A`	`T`	
bits:	1.0	1.3	0.3	0.3	0.2	1.5	$\Rightarrow 4.4$

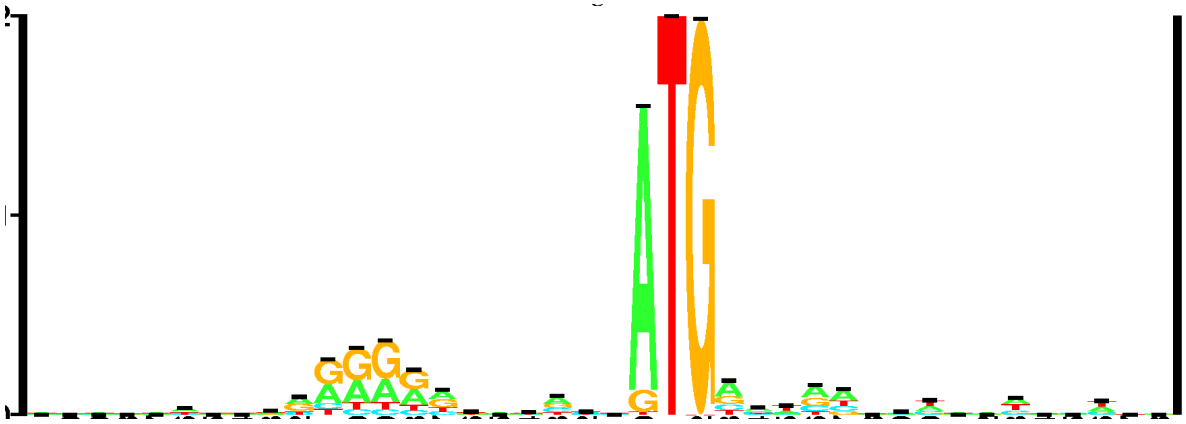

Abb. 2.1: Sequenzlogo eines Alignments von Translation–Startcodons von *E. coli*. Das übliche Startcodon ATG (codiert für Methionin) ist bei weitem das häufigste und dominiert das Logo. Deutlich sichtbar ist auch die upstream des Translationsstarts gelegene purinreiche Shine–Dalgarno–Sequenz. Daten aus [67].

- Korrelationen zwischen Nukleotiden vernachlässigt
  - Argument: Die räumliche Anordnung der Nucleotide ist wichtig für die Bindung von Proteinen an die DNA.
  - Trotzdem: Die Konsensussequenz ist eine erstaunlich gute Approximation!
- *multiple alignment* nötig

Trotzdem: Die Konsensus–Sequenz ist bequem, anschaulich, erfolgreich.

Eine Methode, die Schwäche der Konsensussequenzen zu umgehen, daß zu viel Information über die einzelne Position in der Sequenz verlorengeht, ist die **Berechnung des Informationsgehaltes einer Bindungsstelle** [75]. Die dazu notwendigen Konzepte wie die Shannon–Entropie werden in Kap. 3 (ab S. 21) eingeführt.

Eine weitere Möglichkeit ist die **Darstellung von Konsensussequenzen als Sequenzlogos** Schneider und Stephens [74], vgl. auch Abb. 2.1. Die Idee dahinter ist, die Abweichung von der Gleichverteilung (alle Monomere treten mit der gleichen Wahrscheinlichkeit auf) hervorzuheben. Die meisten funktionalen Sequenzabschnitte weisen eine signifikante Abweichung ihrer Monomerverteilung von der Gleichverteilung auf.

Aus den beobachteten Häufigkeiten der Monomere an einer gegebenen Position i in der Sequenz kann die Abweichung von der zufälligen Verteilung berechnet werden als

$$D(i) = \log_2 |A| + \sum_{k=1}^{|A|} p_k(i) \log_2 p_k(i) \tag{2.1}$$

Dabei ist $|A|$ die Länge des Alphabets (normalerweise 4 für die DNA/RNA oder 20 für Proteine). Wird der Logarithmus zur Basis 2, $\log_2$ verwendet, dann ist die Einheit der Distanz $D(i)$ "bit". Die Distanz kann als Maximalwert $D(i)_{\max} = \log_2 |A|$ annehmen.

Bei der Sequenzlogo–Darstellung wird eine Spalte von Symbolen dazu genutzt, die Details einer Konsensussquenz an einer Position darzustellen. Die Gesamthöhe der Spalte ist identisch dem Wert von $D(i)$, die Höhe jedes Monomer–Symbols k ist proportional seiner Wahrscheinlichkeit an der Position $p_k(i)$. [1]

Tab. 2.3: Beispiel einer Gewichtsmatrix (*weight matrix*, *position specific score matrix* PSSM) für die Konsensus–Sequenz TATAAT für 242 Promotoren von *E. coli*. Die Gewichte wurden jeweils aus den relativen Häufigkeiten für die Base an der gegebenen Position (Tab. 2.2) als Logarithmus aus dem Quotienten von beobachteter und zufälliger Wahrscheinlickeit ($\log_2[p_i/p_0]$) mit $p_0 = 0.25$ berechnet. Daten nach Staden [83].

A	-2.76	1.81	0.06	1.24	0.97	-3.06
C	-1.46	-3.11	-1.18	-0.94	-0.18	-2.32
G	-1.76	-5.00	-1.06	-0.64	-1.06	-3.64
T	1.68	1.04	1.04	-0.94	-0.47	1.83
Konsensus:	T	A	T	A	A	T

2.2 Gewichtsmatrizen (*profiles*)

Profiles "A very productive way of exploiting database redundancy—both in relation to sequence retrieval by alignment and when designing input representations for machine learning algorithms—is the *sequence profile* [226]. A profile describes position by position the amino acid variation in a family of sequences organized into a multiple alignment. While the profile no longer contains information about the sequential pattern in individual sequences, the degree of sequence variation is extremely powerful in database search, in programs such as PSI–BLAST, where the profile is iteratively updated by the sequences picked up by the current version of the profile [12]." [1]

Gewichtsmatrix, *weight matrix*

Wahrscheinlichkeit verschiedener Hypothesen:

1. Beobachtung TACAAT ist zufällig (Annahme: $p_i = \frac{1}{4}$, Bernoulli–Sequenz [statistisch unabhängig])

$$W_{\text{random}}(\texttt{TACAAT}) = \left(\frac{1}{4}\right)^6 \approx 0.0002$$

2. Der Promoter ist nach der obigen Matrix (Tab. 2.2) "gezogen"

$$W_{\text{Promoter}}(\texttt{TACAAT}) = 0.80 \cdot 0.88 \cdot 0.11 \cdot 0.59 \cdot 0.49 \cdot 0.89 = 0.0140$$

▲ **"likelihood–ratio"** (engl. *likelihood*, Wahrscheinlichkeit) Verhältnis von Wahrscheinlichkeiten, hier Quotient aus Wahrscheinlichkeit für eine bestimmte Sequenz und der Zufallswahrscheinlichkeit.

$$\frac{W_{\text{Promoter}}}{W_{\text{random}}} = \frac{0.80}{0.25} \cdot \frac{0.88}{0.25} \cdot \frac{0.11}{0.25} \cdot \ \cdots \approx 57 \tag{2.2}$$

▲ **log–likelihood** Logarithmus des *likelihood ratio* (dem Quotienten aus beobachteter und zufällig zu erwartender Häufigkeit), auch als Gewicht bezeichnet und als solches

zur Bestimmung von Gewichtsmatrizen verwendet.

$$\log_2 \frac{W_{\text{Promoter}}}{W_{\text{random}}} = \log_2 \frac{0.80}{0.25} \cdot \log_2 \frac{0.88}{0.25} \cdot \log_2 \frac{0.11}{0.25} \cdot \ \cdots \approx 5.8 \tag{2.3}$$

$$\text{Gewicht} = \log_2 \frac{\text{beobachtete Häufigkeit}}{\text{zufällig zu erwartende Häufigkeit}} \tag{2.4}$$

▲ **Score** Als *score* wird die Summe der Gewichte für eine spezifische Sequenz bezeichnet.

Bemerkungen

- Beobachtete Häufigkeiten besitzen stochastische ($\approx \sqrt{N}$) und systematische Fehler.
  systematische Fehler
  - Auswahl der Gene nach Wichtigkeit ($\rightarrow$ Promotoren stark)
  - Struktur der Promotoren vorher festgelegt
    $\rightarrow$ neue, abgewandelte Sequenzen werden evtl übersehen
  - *multiple alignment*
- Korrelationen zwischen Positionen bleiben unberücksichtigt
- Zufallsmodelle variabel
  - Bernoulli: $p_i = \frac{1}{4}$ oder p_i realistisch
  - Markov–Modelle
- Warnung: $\frac{W(\text{Promoter})}{W(\text{random})} = 57$ garantiert nicht wirklichen Promoter
  z.B. *E. coli*: mehr als 2000 zufällige TACAAT im gesamten Genom

Bayes–Theorem Letztlich führt die Beantwortung der Frage, ob die Sequenz `TACAAT` ein Promoter sei, direkt zum *Bayes–Theorem*.

$$\begin{aligned} p(\text{TACAAT Prom.}) &= \frac{\text{Zahl der wahren Promotoren mit TACAAT}}{\text{Zahl der wahren Promotoren} + \text{Zahl “falscher” TACAAT}} \\ &= \frac{p(\text{TACAAT}|\text{Promoter}) \cdot \#\text{Promotoren}}{p(\text{TACAAT}|\text{Prom.}) \cdot \#\text{Prom.} + p(\text{TACAAT}|\text{rand.}) \cdot \#\text{Wörter}} \end{aligned}$$

$$\textit{E. coli} \approx \frac{0.0140 \cdot 2000}{0.0140 \cdot 2000 + \frac{1}{4096} \cdot 4600000 \cdot 2} \approx \frac{28}{28 + 2246} \approx 0.012 = 1.2\%$$

Kapitel 3

Einführung in die Informationstheorie (Informationstheorie I)

Eine zentrale Aufgabe der Bioinformatik ist die quantitative Bestimmung des in einer gegebenen Sequenz steckenden Informationsgehaltes. Die zugrundeliegende Informationstheorie geht auf die Arbeiten Shannons zurück [76], der in den 1940er Jahren eine mathematische Theorie der Kommunikation entwickelte.

Da die Informationstheorie Shannons eng mit der Wahrscheinlichkeitstheorie und Statistik verknüpft ist, werden wir zunächst einige zentrale Konzepte der Wahrscheinlichkeitstheorie einführen, um uns dann der Informationstheorie, speziell dem Begriff der **Entropie**, zuzuwenden.

3.1 Mengen und Wahrscheinlichkeiten

▲ Zwei beliebige Mengen A und B heißen **disjunkt** oder elementfremt, wenn sie kein Element gemeinsam haben. Für sie gilt

$$A \cap B = \emptyset = \{\} \tag{3.1}$$

d.h. ihr Durchschnitt (die Schnittmenge) ist eine leere Menge.

▲ **Ereignismenge** Eine Ereignismenge A ist die Vereinigung sich ausschließender (disjunkter) Untermengen A_i mit $i = 1 \ldots \lambda$. (Aus dieser Definition folgt die statistische Unabhängigkeit der Ereignisse)

▲ **Wahrscheinlichkeit** Die Wahrscheinlichkeit $p_i \equiv p_i(A_i)$ einer Untermenge A_i der Ereignismenge A gibt an, wie häufig diese Untermenge (das Ereignis) im Mittel auftritt.

3.1.1 Kolmogorov–Axiome

Die drei Kolmogorov–Axiome[1] (nonnegativity, total probability one rule, addition formula) sind das theoretische Fundament der modernen Wahrscheinlichkeitstheorie (*probability theory*).

▲ **Erstes Kolmogorov–Axiom** Für jedes Ereignis i gilt

$$0 \leq p_i \leq 1 \tag{3.2}$$

d.h. die Wahrscheinlichkeit dieses Ereignisses wird durch eine reelle Zahl zwischen 0 und 1 dargestellt.

▲ **Zweites Kolmogorov–Axiom** Für eine Ereignismenge $E = \{1, \ldots, \lambda\}$ gilt

$$\sum_{i=1}^{\lambda} p_i = 1 \tag{3.3}$$

d.h. die Summe der Wahrscheinlichkeiten aller möglichen Ereignisse ist immer 1.

▲ **Drittes Kolmogorov–Axiom** Eine Folge sich gegenseitig ausschließender Ereignisse $E_1, E_2, \ldots$ genügt der Bedingung

$$p(E_1 \cup E_2 \cup \cdots) = \sum_i p(E_i) \tag{3.4}$$

d.h. die Wahrscheinlichkeit einer Ereignismenge, die die Vereinigung anderer sich ausschließender Untermengen ist, ist die Summe der Wahrscheinlichkeiten dieser Untermengen. Dies wird σ–Additivität genannt. Gibt es irgendeine Überschneidung zwischen den Untermengen (d.h. die Untermengen sind nicht disjunkt), gilt diese Beziehung nicht.

3.1.2 Bedingte Wahrscheinlichkeit und Bayes–Theorem

▲ **bedingte Wahrscheinlichkeit** Die Wahrscheinlichkeit, daß Ereignis A eintritt unter der Voraussetzung von Ereignis B,

$$p(A|B) = \frac{p(A,B)}{p(B)}, \qquad p(B) \neq 0 \tag{3.5}$$

wird bedingte Wahrscheinlichkeit genannt. Dabei ist $p(A,B)$ die Verbundwahrscheinlichkeit beider Ereignisse und $p(B)$ die Wahrscheinlichkeit für das Ereignis B. Analog läßt sich die bedingte Wahrscheinlichkeit für das Ereignis B unter der Voraussetzung von A schreiben als:

$$p(B|A) = \frac{p(B,A)}{p(A)}, \qquad p(A) \neq 0 \tag{3.6}$$

[1] benannt nach dem russischen Mathematiker Andrey Kolmogorov (1903–1987)

Aus der Beziehung

$$p(A, B) = p(B, A)$$

ergibt sich durch Umstellen der Gleichungen (3.5) und (3.6) die Identität

$$p(A|B) \cdot p(B) = p(B|A) \cdot p(A)$$

und daraus das sogenannte **Bayes–Theorem**

$$p(A|B) = \frac{p(B|A) \cdot p(A)}{p(B)} \tag{3.7}$$

▲ **statistische Unabhängigkeit** Zwei Ereignisse A und B sind statistisch unabhängig, wenn sich ihre Verbundwahrscheinlichkeit $p(A, B)$ aus dem Produkt der Einzelwahrscheinlichkeiten ergibt:

$$p(A, B) = p(A) \cdot p(B) \tag{3.8}$$

Für sie gilt demzufolge mit den Gleichungen (3.5) und (3.6)

$$p(A|B) = p(A) \qquad\qquad p(B|A) = p(B) \tag{3.9}$$

d.h. die Verbundwahrscheinlichkeit eines Ereignisses A, gegeben ein anderes Ereignis B, ist identisch mit der Wahrscheinlichkeit des Ereignisses A.

3.2 Shannon–Entropie

Es sei p_i die Wahrscheinlichkeit für das i–te Ereignis (Symbol) und es gelte

$$i = 1, 2, \ldots, \lambda \qquad\qquad \sum_{i=1}^{\lambda} p_i = 1$$

dann heißt

$$H_1 = \sum_{i=1}^{\lambda} -p_i \log_2 p_i \tag{3.10}$$

Shannon–Entropie. Sie gibt die mittlere Zahl von Alternativfragen nach einem Ereignis (Symbol) bei optimaler Fragestrategie an und ist damit ein Maß für den Informationsgehalt des Symbols i. Alternativ kann die Shannon–Entropie geschrieben werden als

$$H_1 = \left\langle \log_2 \frac{1}{p_i} \right\rangle \tag{3.11}$$

➤ Wird der Logarithmus zur Basis 2 ($\log_2$) verwendet, dann lautet die Einheit der Shannon–Entropie "bit"[2].

[2]Diese Einheit mit der Verbreitung der auf dem Binär–System basierenden Computer weite Verbreitung und Eingang in den allgemeinen Sprachgebrauch gefunden.

■ **Sprache** Für eine Sprache mit 27 Buchstaben ist die rechnerische Wahrscheinlichkeit für einen Buchstaben $p_i = \frac{1}{27}$. Die theoretische Shannon–Entropie eines Alphabetes mit 27 Buchstaben ist also $H_1 = \log_2 27 \approx 4.76$ bit.

Reale Werte für H_1 (errechnet aus realen p_i) liegen etwas darunter: 4.03 bit (Englisch), 4.10 bit (Deutsch), 3.96 bit (Französisch) und 4.35 bit (Russisch, *aber:* $\lambda = 32$!)

Auch die **DNA** kann als eine Sprache aufgefaßt werden. Für das *Rous sarcoma* Virus ($N = 9302$ bp) sind die Buchstaben–Wahrscheinlichkeiten und die Shannon–Entropie:

$$\left.\begin{array}{l} p(A) = 0.24 \\ p(C) = 0.25 \\ p(G) = 0.29 \\ p(T) = 0.22 \end{array}\right\} H_1 = 1.99 \text{ bit}$$

3.3 Gewichtsmatrix und Informationstheorie

Mit der Shannon–Entropie kann die in einer Konsensus–Sequenz steckende Information quantifiziert werden. Über die Gewichte einer Gewichtsmatrix (*weight matrix*, *position specific score matrix* PSSM) ergibt sich ein Zusammenhang zwischen den Gewichtsmatrizen und der Shannon–Entropie, die gleichzeitig eine quantitative Aussage über die Konserviertheit einer Position oder Sequenz ermöglicht.

▲ **Gewichte** Unter der Annahme statistischer Unabhängigkeit der vier Basen ergibt sich eine Wahrscheinlichkeit für jede Base von $p_i = \frac{1}{4}$. Das Gewicht einer Gewichtsmatrix errechnet sich aus der Differenz der Logarithmen von beobachtetem und zufälligem Auftreten eines Ereignisses.

$$-\log_2 \frac{1}{4} - \log_2 \frac{1}{p_i} = \log_2 \frac{p_i}{\frac{1}{4}} = \log_2 4p_i = 2 + \log_2 p_i \tag{3.12}$$

Dabei entspricht $\log_2 \frac{1}{4} = 2$ bit dem Informationsgehalt einer Position und $\log_2 \frac{1}{p_i}$ dem Beitrag zur Shannon–Entropie H_1.

Daraus ergibt sich unmittelbar die Größe für die **Konserviertheit einer Position**

$$2 - H_1 \tag{3.13}$$

Entsprechend ist die Konserviertheit einer Bindungsstelle die Summe über die Konserviertheit aller Positionen. Sie ist gleichzeitig ein Maß für die **Stärke einer Bindungsstelle** (je größer die Konserviertheit, desto stärker die Bindungsstelle).

▲ **Score** Der Score ist die Summe der Gewichte für eine spezifische Sequenz

$$S = \sum_{i=1}^{\lambda} \log_2 \frac{p_i}{p_0} \tag{3.14}$$

Tab. 3.1: Informationsgehalt verschiedener molekularer Bindungsstellen. Die Konserviertheit der -10 Box von *E. coli* reicht nicht aus für eine gute Beschreibbarkeit. Daten aus [75], für die *E. coli* -10 Box aus [83]

Bindungsstelle	n	$2 - H_1$
E. coli -10 Box	242	4.4 bit
E. coli Ribosomen–Bindungsstelle (RBS)	149	11.0 bit
T7 RNA Pol	17	35.4 bit
E. coli LacI	2	19.2 bit
E. coli LexA	14	21.1 bit
E. coli TrpR	6	23.4 bit

▲ **Konserviertheit einer Bindungsstelle** mittlerer Score aus dieser Position

$$\begin{aligned}\sum_{i=1}^{4} p_i \log_2 \frac{p_i}{p_0} &= \sum_{i=1}^{4} p_i \log_2 \frac{1}{p_0} + \sum_{i=1}^{4} p_i \log_2 p_i \\ &= \log_2 \frac{1}{p_0} \underbrace{\sum_{i=1}^{4} p_i}_{1} - \underbrace{\sum_{i=1}^{4} -p_i \log_2 p_i}_{H_1} \\ &= 2 \text{ bit} - H_1\end{aligned}$$

3.4 Statistische Thermodynamik ⇔ Affinität von Bindungsstellen

Berg und von Hippel [5]

Mulligan et al. [62]

$$\begin{aligned}p(\text{Bindung}) &\propto \exp\left(-\frac{E_A}{kT}\right) \\ \frac{p_1}{p_2} &= \frac{\exp\left(-\frac{E_{A_1}}{kT}\right)}{\exp\left(-\frac{E_{A_2}}{kT}\right)} = \exp\left(\frac{1}{kT}\left(E_{A_2} - E_{A_1}\right)\right) \\ \log \frac{p_1}{p_2} &= E_{A_2} - E_{A_1} - \text{const}\end{aligned}$$

Kapitel 4

Eukaryotische Genidentifikation

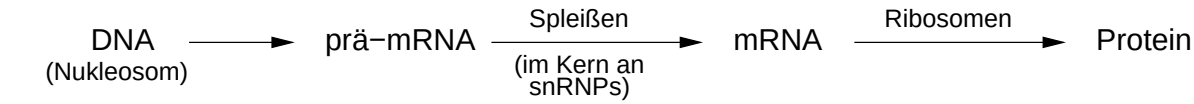

Abb. 4.1: Eukaryotische Genexpression (schematisch). Die Besonderheit der Eukaryoten ist das Spleißen (*splicing*) der prä–mRNA im Zellkern, d.h. das Herausschneiden der Introns.

4.1 Besonderheiten eukaryotischer Gene

Die DNA von Eukaryoten weist gegenüber der von Prokaryoten eine Reihe von Besonderheiten auf:

- **Promotoren**
  Die Promotoren sind nur schwach konserviert und sehr variabel.
- **regulatorische Proteine**
  Viele regulatorische Proteine binden an die DNA (Aktivatoren, Repressoren; Enhancer, Silencer), und das teilweise bis zu 85 kb vom regulierten Gen entfernt.
- **räumliche Struktur**
  Die räumliche Struktur der DNA spielt eine große Rolle.

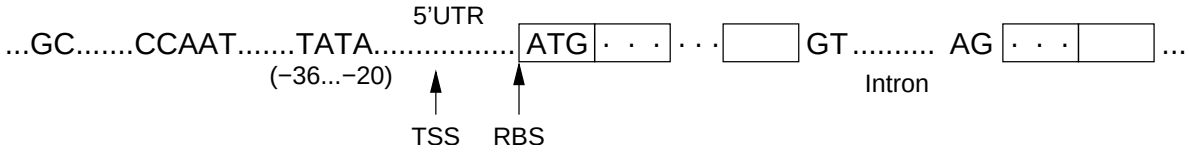

Abb. 4.2: Typischer Aufbau eines eukaryotischen Gens (schematisch). Die Besonderheit des eukaryotischen Gens sind die langen Introns, die die codierenden Abschnitte (Exons) eines Gens unterbrechen. TSS — *Transcription start sequence*, RBS — *ribosome binding site*. Die mittlere Länge eines Exons beträgt ca. 150 bp (z.T. nur 9 bp), die eines Introns 50 bp . . . 100 kb.

- **alternatives Splicen**
  Möglichkeit zur Aufklärung der Differenz zwischen der Zahl der Vorhergesagten Gene und der der cDNAs.

  cDNA: copy DNA bzw. complementary DNA, entsteht aus Rückschreiben der mRNA mittels reverser Transkriptase und enthält deshalb keine Introns mehr.

  - typische Genlänge: 30 kb
  - Zahl der Exons pro Gen: ≈ 10

- **repeats**
  Bis zu 50% des eukaryotischen Genoms bestehen aus repetitiven Sequenzen (*repeats*): Satelliten–DNA, dispersed repeats (SINEs, LINEs), low–complexity regions, microsatellites, triplett diseases

- **wenig codierende Sequenzen**
  Nur 1–2 % des gesamten Genoms sind Protein–codierend. Das macht die Genidentifikation extrem schwierig

- **Intronlänge**
  Die Introns sind typischerweise *wesentlich länger* als Exons.

■ **Schwierigkeiten eukaryotischer Genidentifikation** Getestet wurden verschiedene Programme zur Genidentifikation anhand von 24 ausgewählten Promotoren. Für die Ergebnisse vergleiche die Tabelle.

	Audic	**Autogene**	**GeneID**	**NNPP**	**P'Find**	**P'Scan**	**TATA**	**TSSG**
Sensitivity	5/24	7/24	10/24	13/24	7/24	3/24	6/24	7/24
Specifity (fp)	33	51	51	72	29	6	47	25

Gesamtgenauigkeit der getesten Programme. Für jedes Programm ist die Empfindlichkeit (sensitivity) als Zahl der korrekt detektierten Promotoren und die Spezifität (specifity) als Zahl der Falschpositiven (fp) angegeben. Daten aus [30]

4.2 Exon–Erkennung ("search by content")

Problem: Wie wahrscheinlich ist es, daß ein offenes Leseraster (ORF = open reading frame) flankiert von AG und GT[1] ein Exon ist? (→ "coding potential")

Längenverteilung von zufälligen ORFs:

$$p_i = \frac{1}{64}$$

da alle Codons gleich häufig sind.

[1] Die (auf dem Intron liegenden) Dinukleotide AG (Ende) bzw. GT (Anfang) sind stark konservierte Signalsequenzen für Introns.

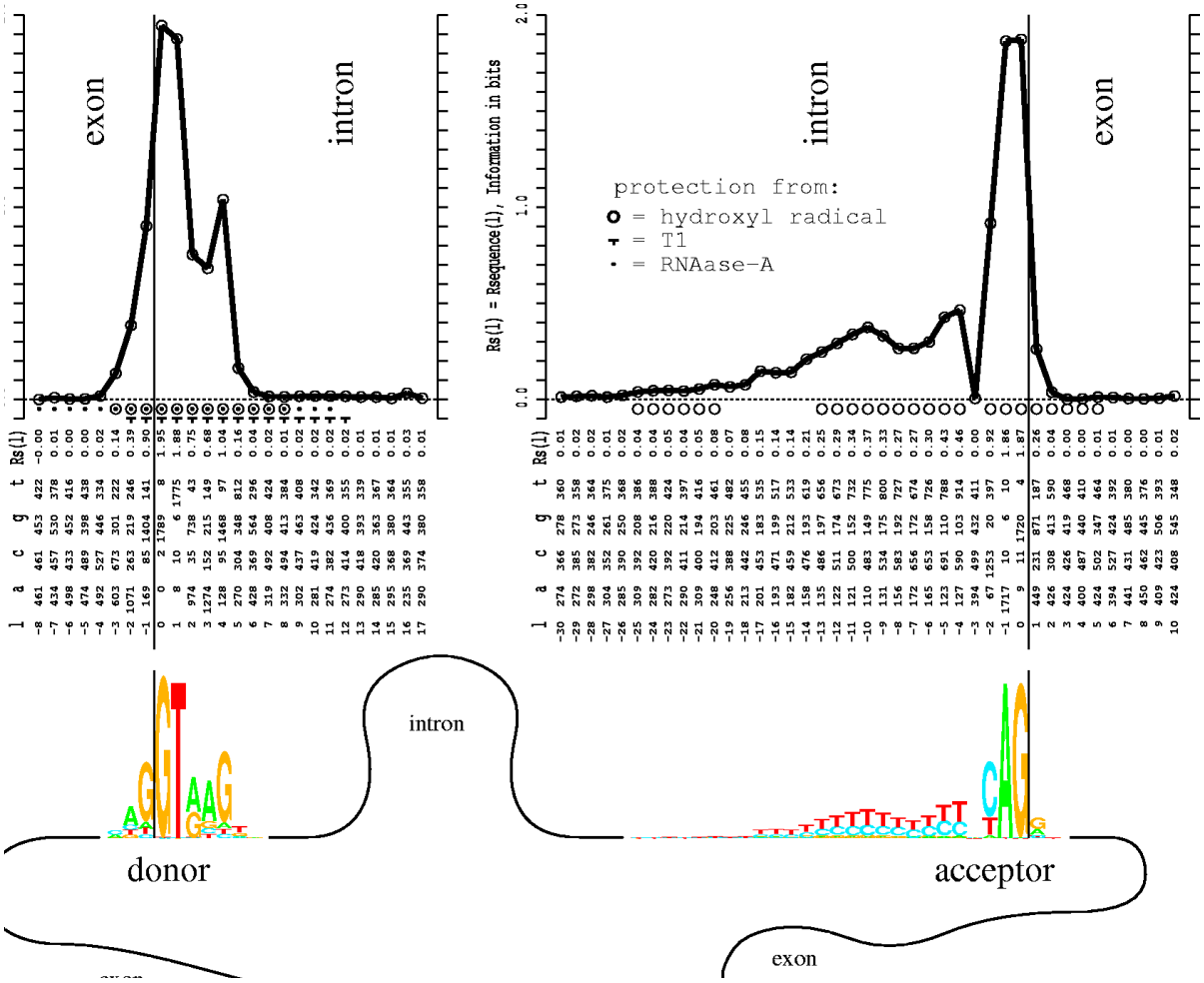

Abb. 4.3: Information curves and sequence logos for human spliceosome binding sites. The left half of the figure shows the donor splice sites from position -8 to position +17, while the right half shows the -30 to +10 region around the acceptor sites. Position zero on both curves is the point on the intron adjacent to the splice point, i.e. on the 50 side, the intron is cut immediately before position zero while on the 30 side it is cut immediately after position zero. (These are the coordinates provided by GenBank.) In the matrix corresponding to each graph, the bottom row, labeled l, contains the position on the sequences relative to the splice points. The next four rows are the numbers of a’s, c’s, g’s, and t’s (labeled as such) found at each position. These were used to create the frequency matrix for the analysis. For random sequences the frequencies at a position in the matrix should be about equal, and examination of the matrixes at the edges shows this to be the case. Examination of the matrixes around the zero points, however, shows a decided inequality in the numbers of the various bases. This means that the sequences around these zero positions are not random, and therefore there is information (conservation) at these points (the spikes on the graph). The top row of the matrixes, labeled Rs(l) (= Rsequence(l)), is the amount of information present at position l on the sequences. The symbols found between this row and the graph represent those positions apparently protected by the spliceosome in protection experiments (Mount et al., 1983; Wang & Padgett, 1989; R. A. Padgett, personal communication). The curve and the matrix are summarized by the sequence logos (Schneider & Stephens, 1990) at the bottom of the figure. In a logo, the total height of the stack of letters at each position is the amount of information present at that position. The heights of the individual letters are proportional to their frequencies at that position. The letters are ordered with the most frequent on top, so the most common base appears on the top of the logo and one may read the consensus sequence directly from the figure. The vertical bars are 2 bits high; the region between them is removed during splicing. Error bars for the heights of the information curves and sequence logos are not shown in the figure because they are below the resolution of the printer. From Stephens und Schneider [86]

Da es drei Stop–Codons gibt (TGA, TAA, TAG), ist die Wahrscheinlichkeit für ein Stop–Codon

$$p_{\text{STOP}} = \frac{3}{64}$$

und die Wahrscheinlichkeit dafür, kein Stop–Codon in einer Sequenz der Länge L zu finden

$$p(L) \propto \left(\frac{61}{64}\right)^L = (1 - p_{\text{STOP}})^L = \prod_L q_i$$

$$p(L) = \frac{3}{61}\left(\frac{61}{64}\right)^L$$

Um die durchschnittliche Länge eines zufälligen ORFs (Sequenz ohne Stop–Codon) zu erhalten, summieren wir über die Wahrscheinlichkeiten für eine Sequenz ohne Stop–Codon für alle Längen L, d.h. $L \to \infty$

$$\begin{aligned}\sum_{L=1}^{\infty}\left(\frac{61}{64}\right)^L &= \sum_{L=1}^{\infty} p^L = \frac{p}{1-p} \\ &= \frac{\frac{61}{64}}{1-\frac{61}{64}} = \frac{61}{3}\end{aligned}$$

Damit ergibt sich als typische (mittlere) Länge für zufällige ORFs unter der Annahme der Gleichverteilung der Nukleotide:

$$\langle L \rangle = \frac{1}{1-p} = \frac{61}{3} \approx 20$$

Dieser Wert entspricht durchaus der Realität: Insbesondere für GC–reiche DNA kommen viele zufällige ORFs der Länge 20–100 vor (sogenannte “Falschpositive”).

(schwache) statistische Eigenschaften von Introns:

- z.T. Periode 2, 6
- viele Poly(A)–, Poly(T)–Sequenzen
- minimale Länge $\approx$ 50 bp
- AT–Gehalt höher als bei Exons
- GT, AG Reduktion

Exons

- schwache Codon–Codon Korrelationen (z.B. von α–Helix)
- Proteindomänen, repeats, ...

Tab. 4.1: The nucleotide distribution in the data set given for translated exon (E), intron (I), untranslated exon (M), and non–transcribed DNA (N). Notice, in introns, the high presence of adenine and, especially, thymine. From Hebsgaard et al. [41]

Class	Count nt	%	A	C	G	T
E	186.585	39.46	27.60	21.06	24.86	26.48
I	113.369	22.73	26.45	15.31	17.37	40.86
M	20.905	6.44	29.32	18.44	16.51	35.73
N	149.462	31.36	32.58	17.53	16.77	33.12

- stärkstes Signal: Periode 3

  Grund für Periode 3

  - ungleichförmige AS–Verteilung
  - nonuniform codon usage
    - keine Gleichverteilung der Codons für dieselbe AS
    - z.T. tRNAs nicht alle vorhanden
  - Isochoren → 3. Position
  - Translationscode
  - GC–Skew (G häufiger als auf dem leading strand)

One well–known statistical pattern of exons is the existence of a reading frame and the unequal use of coding nucleotide triplets (codons). The reading frame induces a triplet periodicity in coding sequences, which is absent in non–coding sequences. The non–uniform codon usage gives rise to a different relative frequency $f(b|l)$ of each nucleotide $b = \mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}$, in a position $l \in (1, 2, 3)$ of the reading frame. Possible reasons for the non–uniformity of the codon usage are: (i) the non–uniform amino acid composition of proteins, (ii) the unequal number of codons encoding different amino acids, and (iii) the non–uniform distribution of synonymous codons encoding the same amino acid. [47]

4.3 Quantifizierung des Codierungspotentials

▲ **Codierungspotential** *coding potential*, spezifische Muster der Codon–Verwendung, die sich von den Triplett–Häufigkeiten nichtcodierender Regionen unterscheidet. [34]

Tab. 4.2: The nucleotide distribution at the three codon positions for the translatesd exon sequence in *A. thaliana*. The non–organism specific reading frame pattern G/non–G on the two first codon positions is clearly visible. From Hebsgaard et al. [41]

Nucleotide	Position 1	Position 2	Position 3
A	0.29	0.31	0.23
C	0.19	0.23	0.21
G	0.34	0.18	0.23
T	0.18	0.28	0.33

Es existiert eine Korrelation zwischen dem Codierungspotential und der Wahrscheinlichkeit, daß es sich bei einer bestimmten DNA–Region um eine Protein–codierende Sequenz handelt. Aus diesem Grund ist das Codierungspotential der Kern vieler Genidentifizierungs–Programme, um ORFs grob zu lokalisieren. [47]

Vorgehen [47]

Zur Quantifizierung des Codierungspotentials wird ein Fenster der Länge $3N$ Basenpaare (bp) einer DNA–Sequenz in N nicht–überlappende Tripletts zerlegt. Die Zahl der Vorkommen einer Base b, $b \in \{A, C, G, T\}$ an einer gegebenen Triplett–Position l, $l \in \{1, 2, 3\}$, sei $N(b|l)$, die relative Häufigkeit (*relative frequency*) dieser Base $f(b|l) = N(b|l)/N$.

Daraus kann die *frame dependence matrix* $\boldsymbol{F}$ erstellt werden, eine 4×3 (Base $\times$ Position) Element Matrix mit den $f(b|l)$ als Elementen. Da für jedes l die Normalisierung jeder Spalte von $\boldsymbol{F}$ zu $\sum_{b=1}^{4} f(b|l) = 1$ führt, sind nur 9 der 12 Zahlen unabhängig.

Als nächstes wird die mittlere Häufigkeit jeder Base $f(b) = \sum_{l=1}^{3} f(b|l)/3$ über die drei Positionen l des Tripletts aus $f(b|l) = f(b) + d(b|l)$ errechnet und das Ergebnis als Vektor $\boldsymbol{f}$ geschrieben. Die Residuen $d(b|l)$ werden ihrerseits in einer Matrix $\boldsymbol{D}$ zusammengefaßt, die die Abweichung der Basenzusammensetzung an einer Position vom zufällig erwarteten Auftreten angibt.

■ **human beta–myosin heavy chain (HUMBMYH7) gene** Zur Verdeutlichung soll das oben gesagte an einem Ausschnitt des Gens (HUMBMYH7) durchgerechnet werden. Die Sequenz des Gen–Ausschnittes lautet dabei wie folgt (es handelt sich um die ersten 18 Tripletts nach dem Starcodon `ATG`):

$$\underbrace{\texttt{GGAGATTCGGAGATGGCAGTCTTTGGGGCTGCCGCCCCCTACCTGCGCAAGTCA}}_{\text{Fenster der Länge } 3N = 54 \text{ bp}}$$

$$\underbrace{\texttt{GGA GAT TCG GAG ATG ... ... TCA}}_{N = 18 \text{ Tripletts des oberen Fensters}}$$

Aus diesen Daten können nun die 12 Zahlen $N(b|l)$ und die zugehörigen Häufigkeiten $f(b|l)$, $f(b)$ und $d(b|l)$ berechnet und $\boldsymbol{F}$, $\boldsymbol{f}$ und $\boldsymbol{D}$ dargestellt werden (Rundungsgenauigkeit 10^{-2}):

$$\boldsymbol{F} = \begin{pmatrix} 0.11 & 0.22 & 0.17 \\ 0.17 & 0.39 & 0.33 \\ 0.50 & 0.17 & 0.33 \\ 0.22 & 0.22 & 0.17 \end{pmatrix}$$

$$\boldsymbol{f} = \begin{pmatrix} 0.17 \\ 0.30 \\ 0.33 \\ 0.20 \end{pmatrix}$$

$$\boldsymbol{D} = \begin{pmatrix} -0.06 & 0.05 & 0.01 \\ -0.13 & 0.09 & 0.04 \\ 0.17 & -0.16 & -0.01 \\ 0.02 & 0.02 & -0.04 \end{pmatrix}$$

Diese Matrizen zeigen unter anderem den Überschuß (bzw. das Fehlen) von G in der ersten (zweiten) Codonposition und den hohen GC–Gehalt der dritten Position. [47]

biologische Hintergründe Die unterschiedlichen Häufigkeiten von Nukleotiden für jede Position in einem Triplett (Codierungspotential) haben ihre Ursache im genetischen Code und der (ebenfalls nicht zufälligen) Anordnung und Häufigkeit von Aminosäuren in Proteinen. Folgende Häufigkeiten und biologische Begründungen lassen sich festhalten:

- **1. Position im Triplett**: G

  Häufig in Proteinen auftretende Aminosäuren werden mit G an der 1. Stelle des Codons im genetischen Code codiert.

- **2. Position im Triplett**: A/T

  Die zweite Position des Codons entscheidet meist darüber, ob die betreffende Aminosäure hydrophil oder hydrophob ist.

- **3. Position im Triplett**: relativ variabel

  Die dritte Stelle des Codons ist für die Aminosäure–Codierung meist redundant. Sie dient daher der Anpassung an das Milieu auf der DNA (GC–reich oder AT–reich).

Weiter kommt hinzu, daß nicht alle tRNA–Species in allen Organismen verhanden sind. Das führt zu sogenannten **codon usage statistics** [41], Abweichungen der Codonhäufigkeiten in codierenden Bereichen von den mittleren Häufigkeiten, die ihrerseits als schwaches Signal zur Identifizierung von Genen herangezogen werden können.

4.3.1 Position Asymmetry

Es gibt verschiedene Ansätze, das Codierungspotential zu quantifizieren: *base composition asymmetry* [29], *uneven positional base frequencies* [82], *positional asymmetry* [31], *fourier transform* [79, 60], *positional information* und *average mutual information* [38, 39]. Wir wollen uns hier auf die *positional asymmetry* beschränken, für eine Übersicht über die anderen Verfahren und eine Verallgemeinerung vgl. Holste et al. [47].

▲ **position asymmetry** [31] Das Codierungspotential wird aus der Summe über die Streuung von $f(b|l)$ um $f(b)$

$$D_2 = \sum_{b=1}^{4} \sum_{l=1}^{3} d^2(b|l) \qquad d(b|l) = f(b|l) - f(b) \tag{4.1}$$

berechnet. Für codierende Sequenzen ist $D_2 > 0$, für nicht–codierende Sequenzen gilt $D_2 = 0$. Die *psition asymmetry* mißt also die Abweichung der Spaltenvektoren der Matrix $\boldsymbol{F}$ vom mittleren Vektor $\boldsymbol{f}$. Sie ist verwandt mit χ^2, Fourier–Transformation, Korrelationsfunktion und *mutual information* Holste et al. [48].

Ausblick

- Codon, Hexamerhäufigkeiten (Kap. 5)
  - Diskriminationsanalyse
  - Bayes–Formel
  - neuronale Netze
  - hidden Markov models (HMM)
- Periodizitäten, "Gedächtnis" der Sequenz (Kap. 6)
  - Korrelationsfunktion, Entropien

Kapitel 5

Markov–Modelle, Wortentropien (Informationstheorie II)

5.1 Markov–Modelle

Markov–Modelle und hidden Markov models (HMM) stellen eine sehr allgemeine Form von Wahrscheinlichkeitsmodellen für Symbol–Sequenzen dar. Fragen, die an solche Modelle gestellt werden können, sind z.B.:

- Gehört diese Sequenz zu einer bestimmten Familie von Sequenzen?
- Angenommen diese Sequenz kommt aus einer bestimmten Familie, was können wir dann über ihre interne Struktur sagen?

Ein Beispiel für die zweite Art von Fragen ist z.B. der Versuch, α–Helix– oder β–Faltblatt–Strukturen in einer Proteinsequenz zu identifizieren.

Die überwältigende Mehrheit von Veröffentlichungen zum Thema HMM kommt aus dem Gebiet der Spracherkennung, wo diese Prozesse erstmals in den frühen 1970er Jahren angewandt wurden. Die bekannteste Anwendung für HMMs in der Biologie sind die Suche und das Alignment von Sequenzen.

Die Darstellung in diesem Abschnitt stützt sich in weiten Teilen auf die Ausführungen in Durbin et al. [21]. Zunächst werden Markov–Ketten (*Markov chains*) eingeführt und ausführlich besprochen, um mit diesem Wissen dann auf Hidden Markov Modelle (HMM) überzugehen. Letztere werden nur mit ihrem Prinzip vorgestellt, für weitergehende Fragestellungen sei auf die einschlägige Literatur (insbes. auch hier Durbin et al. [21]) verwiesen.

5.1.1 Markov–Ketten

Die Markov–Ketten sollen anhand eines einfachen Beispiels, den *CpG islands*, eingeführt werden. Für Hintergründe zu *CpG islands* vgl. Kap. 7 (ab S. 51). Zwei Fragen können in diesem Zusammenhang gestellt (und mit Markov Modellen beantwortet) werden:

- Vorausgesetzt wir haben einen kurzen Abschnitt genomischer Sequenz: Wie können wir entscheiden, ob er aus einer solchen *CpG island* kommt oder nicht?

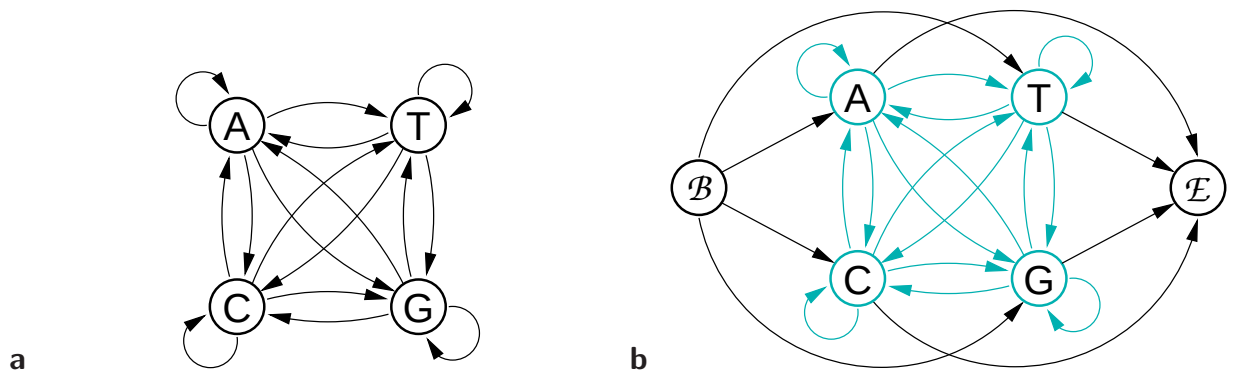

Abb. 5.1: Graphische Darstellung einer Markov–Kette für DNA. Die vier Buchstaben in den Kreisen entsprechen den vier Zuständen der Markov–Kette, die Pfeile zwischen den Kreisen den Übergängen zwischen den Zuständen. Die Zustände entsprechen den vier Basen der DNA (`A`, `C`, `G`, `T`). (**a**) Einfache Markov–Kette für DNA mit vier Zuständen. (**b**) Anfangs– und Endzustände können zu einer Markov–Kette (graues Modell) hinzugefügt werden, um Anfang und Ende einer Sequenz zu modellieren. Aus [21]

- Gegeben sei ein langer Sequenz–Abschnitt: Wie können wir in diesem Abschnitt *CpG islands* identifizieren, sofern er welche enthält?

Zur Modellierung von *CpG islands* wird ein Wahrscheinlichkeitsmodell benötigt, das Sequenzen generiert, in denen die Wahrscheinlichkeit eines Symbols vom vorausgegangenen Symbol abhängt[1]. Der einfachste stochastische Prozeß mit dieser Eigenschaft ist eine Markov–Kette (auch Markov–Prozeß oder AR(1)–Prozeß[2] genannt).

Eine Markov–Kette kann graphisch als eine Ansammlung definierter Zustände dargestellt werden, die jeweils von einem bestimmten Stadium des Systems abhängen (vgl. Abb. 5.1a). Zwischen diesen Zuständen sind Pfeile eingezeichnet, die die Übergänge von einem Zustand in den nächsten anzeigen und jeweils mit einem Wahrscheinlichkeits–Parameter verknüpft sind.

Diese Parameter werden **Übergangswahrscheinlichkeiten** genannt. Die Wahrscheinlichkeit, vom Zustand s in den Zustand t überzugehen, wird geschrieben als:

$$a_{st} = P(x_i = t | x_{i-1} = s) \tag{5.1}$$

Die Übergangswahrscheinlichkeiten geben an, mit welcher Wahrscheinlichkeit das System aus einem Stadium in ein anderes bzw. aus einem Zustand in einen anderen übergeht.

Für jede stochastische Modellierung von Sequenzen kann die Wahrscheinlichkeit $P(x)$ einer Sequenz x allgemein geschrieben werden als

$$\begin{aligned} P(x) &= P(x_L, x_{L-1}, \ldots, x_1) \\ &= P(x_L | x_{L-1}, \ldots, x_1) P(x_{L-1} | x_{L-2}, \ldots, x_1) \cdots P(x_1) \end{aligned} \tag{5.2}$$

indem die Definition der Verbundwahrscheinlichkeit, $P(X, Y) = P(X|Y)P(Y)$, viele Male angewandt wird. Die entscheidende Eigenschaft einer Markov–Kette ist, daß die Wahrscheinlichkeit des Symbols x_i ausschließlich vom Wert des vorhergehenden Symbols x_{i-1} abhängt,

[1] Die Abhängigkeit ist deshalb wichtig, weil es bei *CpG islands* ja auf Dinukleotide, maßgeblich `CG`, ankommt.

[2] AR(1)–Prozeß: autoregressiver Prozeß 1. Ordnung

nicht von der gesamten Sequenz, d.h.

$$P(x_i|x_{i-1},\ldots,x_i) = P(x_i|x_{i-1}) = a_{x_{i-1}x_i}$$

Damit ergibt sich Gl. (5.2) zu

$$\begin{aligned} P(x) &= P(x_L|x_{L-1})P(x_{L-1}|x_{L-2})\cdots P(x_2|x_1)P(x_1) \\ &= P(x_1)\prod_{i=2}^{L} a_{x_{i-1}x_i} \end{aligned} \tag{5.3}$$

Diese Gleichung für die Wahrscheinlichkeit $P(x)$ einer bestimmten Sequenz x gilt allgemein für jede Markov–Kette, auch wenn sie an einem konkreten Beispiel hergeleitet wurde.

Modellierung des Beginns und Endes von Sequenzen

Genauso, wie die Übergangswahrscheinlichkeiten a_{st} definiert werden müssen, muß die Wahrscheinlichkeit $P(x_1)$ in einem bestimmten Zustand zu beginnen, vorgegeben werden. Es ist möglich, dem Modell einen eigenen **Anfangsstatus** hinzuzufügen, um die Inhomogenität der Gl. (5.3) zu vermeiden. Gleichzeitig wird damit dem Alphabet ein neuer Buchstabe $\mathcal{B}$ zugefügt. Durch die Definition $x_0 = \mathcal{B}$ ist der Anfang einer Sequenz ebenfalls in Gl. (5.3) eingeschlossen: Die Wahrscheinlichkeit des ersten Buchstabens in der Sequenz ist also z.B.

$$P(x_1 = s) = a_{\mathcal{B}s}$$

In gleicher Weise kann das Ende einer Sequenz durch die Hinzunahme eines Symbols $\mathcal{E}$ modelliert werden. Dann ist die Wahrscheinlichkeit, mit dem Stadium t zu enden

$$P(\mathcal{E}|x_L = t) = a_{t\mathcal{E}}$$

Die Anwendung dieser beiden neuen Symbole $\mathcal{B}$ und $\mathcal{E}$ im DNA–Modell ist in Abb. 5.1b gezeigt.

In der Praxis müssen keine neuen Buchstaben zum Alphabet hinzugefügt werden. Stattdessen können die beiden neuen Zustände auch als *silent states* (ruhende Zustände) behandelt werden, die jeweils nur als Beginn– bzw. Endpunkt dienen.

Traditionell wird das Ende einer Markov–Kette nicht modelliert, die Sequenz kann überall enden. Wird ein expliziter Endstatus mit in die Modellierung integriert, führt das zu einer Modellierung der Längenverteilung der Sequenz. In diesem Fall beschreibt das Model eine Wahrscheinlichkeitsverteilung über alle möglichen Sequenzen aller Länge. Die Verteilung über die Längen fällt exponentiell ab.

Nutzung von Markov–Ketten zur Unterscheidung (Diskriminationsanalyse)

Die primäre Verwendung von Gl. (5.3) ist die Berechnung der Werte für einen Test der Verhältnisse von Wahrscheinlichkeiten (*likelihood ratio test*). Das soll im folgenden an realen Daten von *CpG islands* illustriert werden: Aus einer Menge menschlicher DNA–Sequenzen wurden insgesamt 48 mutmaßliche *CpG islands* ausgewählt. Dann wurden zwei Markov–Ketten hergeleitet: eines für die *CpG island*–Regionen (das + Modell), das andere für die restlichen Sequenzen (das − Modell).

Tab.5.1: Übergangswahrscheinlichkeiten für einen Test der Wahrscheinlichkeits–Verhältnisse *likelihood ratio test*. Zugrunde liegen reale Daten für *CpG islands* menschlicher DNA–Sequenzen. 48 mutmaßliche *CpG islands* wurden ausgewählt und für diese zwei Markov–Ketten modelliert, eine für die als *CpG islands* bezeichneten Regionen (das '+' Modell) und die andere für die verbleibenden Sequenzen (das '−' Modell). Die erste Reihe jeder Tabelle gibt die Häufigkeiten wieder, mit denen ein A von jeder anderen der vier Basen gefolgt wird. Das gleiche gilt für die anderen Basen in den verbleibenden Reihen. Jede Reihe ergibt aufsummiert 1. Daten aus [21]

+	A	C	G	T
A	0.180	0.274	0.426	0.120
C	0.171	0.368	0.274	0.188
G	0.161	0.339	0.375	0.125
T	0.079	0.355	0.384	0.182

−	A	C	G	T
A	0.300	0.205	0.285	0.210
C	0.322	0.298	0.078	0.302
G	0.248	0.246	0.298	0.208
T	0.177	0.239	0.292	0.292

Die Übergangswahrscheinlichkeiten für jedes Modell wurden über die Gleichung

$$a_{st}^{+} = \frac{c_{st}^{+}}{\sum_{t'} c_{st'}^{+}} \tag{5.4}$$

und ihr Gegenstück für a_{st}^{-} berechnet. Dabei ist c_{st}^{+} die Zahl der Fälle, in denen der Buchstabe t auf den Buchstaben s folgt. Die a_{st}^{+} sind also *maximum likelihood* (ML) Schätzer für die Übergangswahrscheinlichkeit. Die errechneten Werte sind für beide Modelle in Tab. 5.1 wiedergegeben.

Die Wahrscheinlichkeiten für die einzelnen Basen sind nicht identisch. Ein Beispiel: Es kommt wesentlich häufiger vor, daß A von G gefolgt wird als von C. Weiterhin sind die Tabellen auch nicht symmetrisch: Für beide Tabellen gilt, daß die Wahrscheinlichkeit, daß die Base G auf C folgt, wesentlich geringer als andersherum (C folgt auf G). Dieser Effekt ist allerdings für das '−' Modell wesentlich ausgeprägter, was aufgrund der CpG–Anomalie zu erwarten ist.

Um dieses Modell zur Unterscheidung zwischen *CpG islands* und anderen Regionen zu nutzen, werden die Logarithmen der Quotienten der Wahrscheinlichkeiten berechnet:

$$\begin{aligned} S(x) = \log \frac{P(x|\text{model }+)}{P(x|\text{model }-)} &= \sum_{i=1}^{L} \log \frac{a_{x_{i-1}x_i}^{+}}{a_{x_{i-1}x_i}^{-}} \\ &= \sum_{i=1}^{L} \beta_{x_{i-1}x_i} \end{aligned}$$

Die Werte $S(x)$ werden auch *Score* (Punktzahl) genannt, mit der Sequenz x und den Gewichten (Scores, Logarithmen der Wahrscheinlichkeits–Quotienten) $\beta_{x_{i-1}x_i}$. Für die Werte für β vgl. Tab. 5.2.

Allgemeine Definition eines Markov–Prozesses

Bisher haben wir uns nur Markov–Prozesse erster Ordnung (auch: Markov–Kette, AR(1)–Prozeß) angesehen. Die charakteristische (und die Ordnung des Prozesses bestimmende)

Tab. 5.2: Scores (Logarithmen der Wahrscheinlichkeits–Quotienten, *log likelihood ratios*) einander entsprechender Übergangswahrscheinlichkeiten für die beiden Modelle ('+' und '−') zur Unterscheidung von *CpG islands* von anderen Sequenzabschnitten. Da zur Berechnung der Logarithmus zur Basis 2 ($\log_2$) verwandt wurde, sind die Scores in der Einheit "bit" angegeben. Daten aus [21]

β	A	C	G	T
A	−0.740	0.419	0.580	−0.803
C	−0.913	0.302	1.812	−0.685
G	−0.624	0.461	0.331	−0.730
T	−1.169	0.573	0.393	−0.679

Eigenschaft dieser Klasse von Modellen ist, daß der Zustand des Systems nur vom unmittelbar vorausgegangenen Zustand abhängt. Oder alternativ formuliert: Der nächste Zustand des Systems hängt nur von seinem augenblicklichen Zustand ab.

Es existieren jedoch auch Markov–Prozesse höherer Ordnung, deren gegenwärtiger Zustand dann von entsprechend mehr vorausgegangenen Zuständen abhängig ist. Das führt zur allgemeinen Definition eines Markov–Prozesses:

▲ **Markov–Prozeß** Ein Markov–Prozeß der Ordnung k ist ein stochastischer Prozeß, bei dem jedes Ereignis von den vorausgegangenen k Ereignissen abhängt, so daß gilt:

$$P(x_i|x_{i-1}, x_{i-2}, \ldots, x_1) = P(x_i|x_{i-1}, \ldots, x_{i-n})$$

[21]. Ein solcher Prozeß entspricht den Übergängen vom k–Wort zum $(k+1)$sten Symbol. Ein Markov–Prozeß der Ordnung 1 wird auch als Markov–Kette (*Markov chain*) oder AR(1)–Prozeß[3] bezeichnet.

Markov–Prozesse höherer Ordnung spielen ebenfalls in der Sequenzanalyse eine große Rolle. So können z. B. Hexamer–Häufigkeiten durch einen Markov–Prozeß der 5. Ordnung modelliert werden.

5.1.2 Hidden Markov models (HMM)

Wenden wir uns der zweiten der beiden eingangs zu den *CpG islands* gestellten Fragen zu: Wie können wir sie in einer langen nichtannotierten (*unannotated*) Sequenz finden?

Eine Möglichkeit wäre, die oben besprochenen Markov–Ketten zu verwenden: Berechne für ein Fenster definierter Länge N die Scores um jedes Nukleotid in der Sequenz und stelle die Ergebnisse graphisch dar. *CpG islands* ergäben hier positive Scores. Doch dieser Ansatz ist in mancher Hinsicht unbefriedigend: *CpG islands* haben zum einen scharfe Grenzen, zum anderen variable Längen. Ein besserer Weg ist die Erstellung eines Modells für die gesamte Sequenz, das beide Markov–Ketten in sich vereint. Das führt direkt zu den Hidden Markov Modellen (HMM).

Um in einem Modell die "Inseln" in einem "Meer" von "Nicht–Insel–Sequenzen" zu simulieren, werden beide im vorherigen Abschnitt besprochenen Markov–Ketten in einem Modell

[3] AR(1)–Prozeß: autoregressiver Prozeß 1. Ordnung

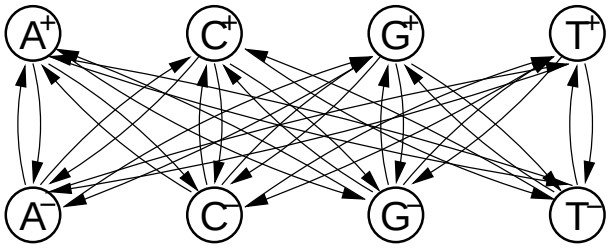

Abb.5.2: Graphische Darstellung eines HMM für *CpG islands*. Zusätzlich zu den hier gezeigten Übergängen gibt es ebenfalls noch den kompletten Satz von Übergängen zwischen den einzelnen Zuständen einer Kette, wie vorher bei den einfachen Markov–Ketten (vgl. Abb. 5.1). Aus [21]

vereint — mit einer kleinen Wahrscheinlichkeit, an jedem Übergangspunkt von einer Kette zur anderen überzuwechseln. Das bringt den Effekt mit sich, daß es jetzt für jedes Symbol (Nukleotid) zwei Zustände des Systems gibt. Dieses Problem kann zunächst dadurch gelöst werden, daß die Zustände umbenannt werden: für *CpG island*–Bereiche die Zustände $\mathtt{A}^+$, $\mathtt{C}^+$, $\mathtt{G}^+$ und $\mathtt{T}^+$ und entsprechend für die restlichen Bereiche $\mathtt{A}^-$, $\mathtt{C}^-$, $\mathtt{G}^-$ und $\mathtt{T}^-$ (vgl. Abb. 5.2)

Die Übergangswahrscheinlichkeiten der beiden Komponenten in diesem Modell entsprechen größtenteils denen für die beiden getrennten Markov–Ketten im vorherigen Abschnitt, mit dem Unterschied, daß für jeden Zustand eine geringe Wahrscheinlichkeit existiert, zur jeweils anderen Komponente zu wechseln. Wird realistischerweise die Wahrscheinlichkeit des Übergangs von '+' nach '−' größer gewählt als anders herum, dann wird sich das System im freien Lauf die längere Zeit in den '−' (nicht–Insel) Bereichen aufhalten.

Der entscheidende Schritt ist die Umbenennung der Zustände: Der grundlegende Unterschied zwischen Markov–Ketten und Hidden Markov Modellen besteht darin, daß bei letzteren kein Eins–zu–eins–Zusammenhang mehr zwischen den Zuständen und den Symbolen besteht. In der Konsequenz bedeutet das: Es ist nicht möglich zu bestimmen, in welchem Zustand sich das System befand, als x_i generiert wurde, wenn wir nur x_i betrachten. Im Beispiel gesprochen: Wenn wir auf ein einzelnes, isoliertes Symbol $\mathtt{C}$ sehen, haben wir keine Möglichkeit, herauszubekommen, ob es vom Zustand $\mathtt{C}^+$ oder $\mathtt{C}^-$ generiert wurde.

Formale Definition

Der große Unterschied zu den Markov–Ketten besteht darin, daß bei HMM zwischen der Abfolge der Zustände und der Abfolge der Symbole unterschieden werden muß. Die Abfolge der Zustände wird als **Pfad** π bezeichnet. Der Pfad folgt einer einfachen Markov–Kette, d.h. die Wahrscheinlichkeit eines Zustandes hängt nur vom vorherigen Zustand ab. Die Markov–Kette für den Pfad wird charakterisiert durch die Parameter

$$a_{kl} = P(\pi_i = l | \pi_{i-1} = k) \tag{5.5}$$

mit dem i–ten Zustand π_i des Pfades. Für die Modellierung des Prozeßbeginns kann wieder ein Anfangszustand eingeführt werden, wie für die Markov–Ketten besprochen. Die Wahrscheinlichkeit, im Zustand k zu starten, wird dabei mit a_{0k} bezeichnet. Analog dazu kann auch wieder ein Endzustand modelliert werden, der ebenfalls als Zustand 0 bezeichet wird (was aber keine Schwierigkeiten bereitet, da man sich nur aus dem Anfangszustand heraus und in den Endzustand hinein bewegen kann.)

Da die Symbole b von den Zuständen k getrennt sind, muß ein neuer Satz an Parametern $e_k(b)$ für das Modell eingeführt werden. Auch wenn im Beispiel mit den *CpG islands* jeder Zustand mit nur einem Symbol verknüpft ist, gilt allgemein: Ein Zustand kann ein beliebiges Symbol aus einer Verteilung über alle möglichen Symbole erzeugen. Daher ist die Wahrscheinlichkeit, daß das Symbol b vom Zustand k erzeugt wird

$$e_k(b) = P(x_i = b|\pi_i = k) \tag{5.6}$$

Diese Wahrscheinlichkeiten werden auch als **Emissionswahrscheinlichkeiten** (*emission probabilities*) bezeichnet.

Was ist "versteckt" (*hidden*) an einem Hidden Markov Modell? Normalerweise sind nur die Symbole bekannt, nicht aber der Pfad.

Häufig werden HMMs genutzt, um Symbolfolgen (Sequenzen) zu erzeugen. Ein allgemeines Schema dieses Vorgangs sieht wie folgt aus:

- Ein Zustand π_1 wird entsprechend der Wahrscheinlichkeit a_{0i} gewählt.
- Entsprechend der Verteilung e_{π_1} für diesen Zustand wird eine Beobachtung ausgegeben.
- Anschließend wird gemäß der Übergangswahrscheinlchkeit $a_{\pi_1 i}$ ein neuer Zustand π_2 gewählt.

Auf diese Weise kann eine Folge zufälliger, künstlicher Beobachtungen erstellt werden. So ist z.B. die Formulierung zu verstehen, $P(x)$ sei die Wahrscheinlichkeit, daß x vom Modell generiert wurde.

Analog zur Wahrscheinlichkeit einer Sequenz für eine Markov–Kette, Gl. (5.3), kann die Verbundwahrscheinlichkeit für eine beobachtete Symbolfolge x und eine Zustandsfolge π eines Hidden Markov Modells geschrieben werden als

$$P(x, \pi) = a_{0\pi_1} \prod_{i=1}^{L} e_{\pi_i}(x_i) a_{\pi_i \pi_{i+1}} \tag{5.7}$$

mit $\pi_{L+1} = 0$. Diese Gleichung ist in der Praxis allerdings von untergeordneter Bedeutung, da meist nur die Beobachtungen bekannt sind, nicht aber der ihnen zugrundeliegende Pfad.

Daher gibt es verschiedene Methoden, den einem HMM zugrundliegenden Pfad zu schätzen. Für Details sei auf die Literatur verwiesen (z.B. Durbin et al. [21]).

Abschließend seien noch zwei kurzgefaßte, in ihrem Schwerpunkt verschiedene Definitionen für HMMs wiedergegeben, wie sie in der Literatur zu finden sind:

▲ **Hidden Markov model** Ein diskretes HMM erster Ordnung ist ein stochastisches (Fortpflanzungs–)Modell für Zeitreihen, das durch eine endliche Menge S von Zuständen, ein diskretes Alphabet A von Symbolen, eine Matrix von Übergangswahrscheinlichkeiten (*probability transition matrix*) $\boldsymbol{T} = (t_{ji})$ und eine Matrix von Emissionswahrscheinlichkeiten (*probability emission matrix*) $\boldsymbol{E} = (e_{iX})$. Das System entwickelt sich zufällig von Status zu Status, während es Symbole aus dem Alphabet ausgibt (emittiert). Ist

das System in einem gegebenen Zustand i, dann hat es die Wahrscheinlichkeit t_{ji}, in den Zustand j überzugehen und die Wahrscheinlichkeit e_{ix}, das Symbol X auszugeben.

Ein HMM kann bildlich durch zwei verschiedene Würfel dargestellt werden, die mit jedem Zustand verbunden sind: einem Würfel für die Übergänge und einem Würfel für die Symbol–Ausgabe. Die grundlegende Annahme von Markov–Prozessen erster Ordnung ist, daß die Emissionen und Übergänge ausschließlich vom gegenwärtigen Zustand abhängen, aber nicht von der Vergangenheit des Systems.

Ausschließlich die Symbole, die das System ausgibt, sind beobachtbar, nicht der darunterliegende *random walk* zwischen den Zuständen. Daher die Bezeichnung "*hidden*" (verdeckt). Die verdeckten *random walks* können als verdeckte Variablen dargestellt werden, die der Beobachtung zugrundeliegen. [1]

▲ **Hidden Markov Modelle** sind Modelle, die einen Algorithmus zur Erzeugung einer Sequenz repräsentieren. Sie enthalten eine Reihe von Elementen (Positionen) die verschiedene Zustände annehmen können (z.B. Nukleotide oder Aminosäuren) und die statistischen Übergangswahrscheinlichkeiten von jedem Zustand eines Elements zu jedem Zustand des Folgeelements. Diese Übergangswahrscheinlichkeiten erhält man durch Sequenzvergleiche mit Sequenzen, die eine gewünschte Eigenschaft besitzen (z. B. ein Promoter zu sein, oder eine α–Helix). Damit läßt sich quantifizieren, wie wahrscheinlich es ist, daß eine neue beobachtete Sequenz durch das Modell erzeugt werden würde. Ist die Wahrscheinlichkeit hoch, wird impliziert, daß die beobachtete Sequenz ähnliche Eigenschaften besitzt wie die Gruppe der Sequenzen, mit denen man das Modell konstruiert hat. [85]

5.2 Wortentropien

Die Entropie ist ein Maß für die mittlere Unbestimmtheit eines Ergebnisses. Für eine Zufallsvariable X mit den Wahrscheinlichkeiten $P(x_i)$ für eine diskrete Menge von K Ereignissen $x_1, \ldots, x_k$ ist die **Shannon–Entropie** definiert als

$$H(X) = -\sum_i P(x_i) \log P(x_i) \tag{5.8}$$

In dieser Definition wird der Term $P(x_i) \log P(x_i)$ dann null wenn gilt $P(x_i) = 0$. Normalerweise wird für den Logarithmus der natürliche Logarithmus (ln) gewählt. Hier ist es üblich, den Logarithmus zur Basis 2 ($\log_2$) zu wählen. In diesem Fall ist die Einheit der Entropie ein 'bit'.

Die Entropie ist maximal, wenn alle $P(x_i)$ identisch sind ($P(x_i) = \frac{1}{K}$). Das entspricht einer maximalen Unsicherheit über das Ergebnis einer zufälligen Stichprobe. Das Maximum ist

$$-\sum_{i=1}^{K} \frac{1}{K} \log \frac{1}{K} = \log K \tag{5.9}$$

Wenn das Resultat der Stichprobe sicher ist, z.B. $P(x_k) = 1$ für ein k und für alle anderen $P(x_i) = 0$, dann ist die Entropie null. [21]

Für DNA–Sequenzen gilt für die Shannon–Entropie H_1 eines Nukleotids

$$H_1 = \sum_{i=1}^{4} -p_i \log_2 p_i \tag{5.10}$$

mit $i = 1 \ldots 4$ (für die Basen A,C,G,T) und der Wahrscheinlichkeit p_i für die Base i. Entsprechend gilt für die Entropie H_2 zweier Nukleotide

$$H_2 = \sum_{i,j=1}^{4} -p_{ij} \log_2 p_{ij} \tag{5.11}$$

Sind die Nukleotide statistisch unabhängig, so gilt die Beziehung

$$H_2 = 2 \cdot H_1 \tag{5.12}$$

wie nachstehend bewiesen wird.

Beweis. Zwei Ereignisse i und j sind statistisch unabhängig, wenn sich ihre Verbundwahrscheinlichkeit $p(i, j)$ aus dem Produkt der Einzelwahrscheinlichkeiten ergibt:

$$p_{ij} = p_i \cdot p_j$$

Unter Ausnutzung dieser Beziehung läßt sich H_2 schreiben als

$$\begin{aligned}
H_2 &= \sum_{i,j=1}^{4} -p_i \cdot p_j \log_2(p_i \cdot p_j) \\
&= \sum_{i,j=1}^{4} -p_i \cdot p_j \log_2 p_i + \sum_{i,j=1}^{4} -p_i \cdot p_j \log_2 p_j \\
&= \underbrace{\sum_{i=1}^{4} -p_i \log_2 p_i}_{H_1} \cdot \underbrace{\sum_{j=1}^{4} p_j}_{=1} + \underbrace{\sum_{j=1}^{4} -p_j \log_2 p_j}_{H_1} \cdot \underbrace{\sum_{i=1}^{4} p_i}_{=1} \\
&= 2H_1
\end{aligned}$$

D.h. die Shannon–Entropie verhält sich für statistisch unabhängige Ereignisse additiv. □

Wenn Nucleotide abhängig sind, gilt:

$$H_2 < 2H_1$$

$2H_1 - H_2$ ist ein Maß für statistische Unabhängigkeit

$$p_{j|i} = \frac{p_{ij}}{p_i}$$

mit der Übergangswahrscheinlichkeit (bedingten Wahrscheinlichkeit) $p_{j|i}$ von i nach j und der Verbundwahrscheinlichkeit p_{ij}; vgl. dazu auch Gl. (3.5) und (3.6), S. 22.

Bei einem Markov–Prozeß 1. Ordnung mit vier Buchstaben (also z.B. für ein DNA–Sequenz–Modell) gibt es 16 Übergangswahrscheinlichkeiten. Siehe z.B. das Jukes–Cantor–Modell in Abschnitt 9.2.4, Seite 76 und die dortigen Abbildungen.

▲ **Markov–Prozeß** Ein Markov–Prozeß der Ordnung k ist ein stochastischer Prozeß, bei dem jedes Ereignis (nur) von den vorausgegangenen k Ereignissen abhängt, so daß gilt:

$$P(x_i|x_{i-1}, x_{i-2}, \ldots, x_1) = P(x_i|x_{i-1}, \ldots, x_{i-n})$$

[21]. Ein solcher Prozeß entspricht den Übergängen vom k–Wort zum $(k+1)$sten Symbol. Ein Markov–Prozeß der Ordnung 1 wird auch als Markov–Kette (*Markov chain*) oder AR(1)–Prozeß[4] bezeichnet.

Die neue Information des $(n+1)$ten Buchstabens, wenn die n vorherigen bekannt sind, ist:

$$h_n = H_{n+1} - H_n \tag{5.13}$$

Die Information pro Buchstabe unter Berücksichtigung der Redundanz, auch "Entropie der Quelle" genannt, lautet:

$$H = \lim_{n\to\infty} h_n = \lim_{n\to\infty} (H_{n+1} - H_n) = \lim_{n\to\infty} \frac{H_n}{n} \tag{5.14}$$

➤ Die Informationstheorie blendet die Semantik (Wortbedeutung) aus. Hier liegen die Grenzen der Informationstheorie.

Shannon [77, 76]

terminology (assuming stationary sequences)

alphabet $\{A_1, A_2, \ldots, A_\lambda\}$

k–word probabilities $p(A_1, A_2, \ldots, A_n) \equiv p(\{S_n\})$

word entropies mean number of alternative questions to guess a k–word using an optimal stategy and knowing all $p(\{S_n\})$

$$H_1 = \sum_{i=1}^{k} -p(A_i) \log p(A_i)$$

$$H_k = \sum_{\{S_n\}}^{4^n} -p(S_n) \log_2 p(S_n) \qquad p(S_n) \triangleq \text{WK eines Wortes der Länge } n$$

[4] AR(1)–Prozeß: autoregressiver Prozeß 1. Ordnung

differential entropies new information of the (k+1)th symbol knowing the preceeding k symbols

$$h_k = H_{k+1} - H_k$$

entropy of the source (relatetd to k–entropy) mean information per letter knowing correlations

$$H = \lim_{k\to\infty} \frac{H_k}{k} = \lim_{k\to\infty} h_k \qquad\qquad H \approx 0.7 \ldots 1.3 \text{ bit}$$

McMillan theorem:

$$\text{effective \# of words} \propto \lambda^{kH} \qquad\qquad p(\{S_n\}) \propto \lambda^{-kH}$$

mutual information measures dependencies of symbols in a distance k

$$p_{ij}^{(k)} \Leftrightarrow \ldots A_i \ldots A_j \ldots$$

statistical independence

$$p_{ij}^{(k)} = p_i \cdot p_j$$

$$I(k) = \sum p_{ij}^{(k)} \log \frac{p_{ij}^{(k)}}{p_i \cdot p_j}$$

Die Schätzung von H_n, h_n, H ist problematisch (Bias, Varianz, Rate–Experimente)

Kapitel 6

Periodizitäten in DNA– und Proteinsequenzen

Ziele:

- DNA
  - Periode 3 $\Rightarrow$ Identifikation von Exons
  - Periode von 10–11 $\Leftrightarrow$ DNA–Struktur (10.55 bp Periode)
  - lange Korrelationen: Isochoren (GC–Schwankungen auf langen Skalen)
- Proteine
  - Periode 3–4: AS
  - 7 AS $\Leftrightarrow$ α–Helices, Leucine–Zipper
  - Periode von 10–11: Proteinstruktur

➤ DNA: Stark konservierte Bindungsstellen treten oft im Abstand von 10.55 bp auf. Dieser Abstand entspricht der Windung der DNA–Helix.

Bindungssstellen sind oft palindromisch[1] aufgebaut.

Die Frage nach Periodizitäten führt zu zwei mathematischen Konzenpten, die dabei helfen, dieses Phänomen zu beschreiben und zu analysieren: Korrelationsfunktion und *mutual information* (Transinformation, beantwortet die Frage nach der Unabhängigkeit zweier Ereignisse).

6.1 Korrelationsfunktionen (Spektren)

Gibt es Paarkorrelationen im Abstand k? Wie stark sind sie?

[1] A palindrome is a word, phrase, number or any other sequence of units (like a thread of DNA) which has the property of reading the same in either direction (the adjustment of spaces between letters is generally permitted). The word palindrome comes from the Greek words palin (back) and dramein (to run) meaning running back. [...] In genetics, a palindromic DNA sequence can form a hairpin. `http://www.wikipedia.org/wiki/Palindrome`

Die Zahl aller A–A–Paare im Abstand k ist definiert als $N_{AA}(k)$ Die Schätzung der Paarwahrscheinlichkeit $\hat{p}_{AA}(k)$ im Abstand k mit der Gesamtzahl N der Paare ergibt sich zu

$$\hat{p}_{AA}(k) = \frac{N_{AA}(k)}{N} \tag{6.1}$$

Die Basen eines Paares werden als statistisch unabhängig betrachtet:

$$\hat{p}_{AA}(k) \approx \hat{p}_A \cdot \hat{p}_A \qquad\qquad \hat{p}_A = \frac{N_A}{N} \tag{6.2}$$

Dann lautet die zugehörige Korrelationsfunktion

$$C_{AA}(k) = p_{AA}(k) - p_A \cdot p_A \tag{6.3}$$

Insgesamt gibt es 16 verschiedene Korrelationsfunktionen für die Basen auf der DNA, 9 davon sind statistisch unabhängig.

Die Fourier–Transformierte der Korrelationsfunktion

$$S_{AA}(f) = \mathcal{F}\left(C_{AA}(k)\right) \tag{6.4}$$

wird auch als **Leistungsspektrum** bezeichnet.

6.2 mutual information (Transinformation)

Sind Nucleotide im Abstand k statistisch unabhängig? Das Kriterium für die statistische Unabhängigkeit lautet:

$$p_{ij}(k) = p_i \cdot p_j \qquad\qquad i, j \in \{1, 2, 3, 4\}$$

wobei i, j für die Basen (A,C,G,T) stehen.

Eine Möglichkeit, diese Frage zu beantworten, ist über die Berechnung der sogenannten *mutual information* (Transinformation)

$$I(k) = \sum_{i,j=1}^{4} p_{ij}^{(k)} \log_2 \frac{p_{ij}^{(k)}}{p_i \cdot p_j} \tag{6.5}$$

Sie ist ein Maß für den Abstand der $\{p_{ij}^k\}$ und $\{p_i \cdot p_j\}$ und wird auch als Kullback–Leibler–Distanz, die Gleichung als Boltzmann–Theorem bezeichnet.

Für $I(k) = 0$ gilt, daß die Paare statistisch unabhängig sind.

Die *mutual information* läßt sich auch schreiben als

$$I(k) = 2H_1 - H_2^{(k)} \qquad\qquad H_2^{(k)} = \sum_{i,j=1}^{4} -p_{ij}^{(k)} \log_2 p_{ij}^{(k)} \tag{6.6}$$

6.3 Beispiele für Periodizitäten

Struktur des Nucleosoms Luger et al. [57], Rhodes [70]

Bei Bakterien auch Periode 11, aber *keine* Nucleosomen. Erklärung: negatives Supercoiling (weniger eng gedreht), führt zur Aufwicklung.

Zusammenfassung

- Eukarya: $T \lesssim 10.55$ bp (Nucleosomen)
- Bacteria: $T > 10.55$ bp (negative supercoiling)
- Archaea: $T \approx 10$ bp (suggests positive supercoiling)

Kapitel 7

Heterogenität von Genomen: Repeats & Isochoren

7.1 Repeats

hier: Nucleotidhäufigkeiten: $p(A)$, $p(C)$, $p(G)$, $p(T) \Rightarrow$ 3 unabhängige Variablen (da gilt (2. Kolmogorov–Axiom): $\sum_i p(i) = 1$mit $i \in \{\texttt{A}, \texttt{C}, \texttt{G}, \texttt{T}\}$)

nicht: "Wörter" (Oligos); repeats (Satelliten–DNA, Retrotransposons)

poly(A), poly(T) $\Rightarrow$ introns;

CpG[1] suppresion (CpG islands — regions with normal CG–content)

human: $p(CG) \approx 0.2 \cdot p(C) \cdot p(G)$

Isochores compositional homogeneous stretches of DNA in vertebrate genomes. 100–300 kb or even larger in size. [9, 6]

C–value (C: "constant", "characteristic") genome size of an organism; defined as the amount of DNA in the haploid genomic set [55]

C–value paradox lack of correspondence between C values and the presumed amount of genetic information contained within genomes [55]

[1] In the human genome wherever the dinucleotide `CG` occurs (frequently written `CpG` to distinguish it from the `C`–`G` base pair across the two strands) the `C` nucleotide (cytosine) is typically chemically modified by methylation. There is a relatively high chance of this methyl–`C` mutating into a `T`, with the consequence that in general `CpG` dinucleotides are rarer in the genome than would be expected from the independent probabilities of `C` and `G`. For biologically important reasons the methylation process is suppressed in short stretches of the genome, such as around the promoters or 'start' regions of many genes. In these regions we see many more `CpG` dinucleotides than elsewhere, and in fact more `C` and `G` nucleotides in general. Such regions are called `CpG` islands [Bird 1987]. They are typically a few hundred to a few thousand bases long. [21]

7.2 Isochoren

stand symmetry

$$p(A) \approx p(T) \qquad\qquad p(C) \approx p(G)$$

$\Rightarrow$ nur ein freier Parameter: GC–Gehalt (*GC–content*)
z.B. GC–content von 40% $\rightarrow p(G) = p(C) \approx 0.2$, $p(A) = p(T) \approx 0.3$

Bakterien: "GC–skew" (leading and lagging strand are different)
bei Eukaryoten wegen der Komplexität der Gene nicht nachgewiesen
Bedeutung des GC–Gehaltes: bestimmt den Schmelzpunkt der DNA und beeinflußt darüber

- regulatorische Regionen: AT–Reich $\rightarrow$ leichter aufschmelzbar
- Repeat–Dichte
- Chromosomen–Dichte
- Replikation, Rekombination

Prokaryoten: starke Unterschiede im GC–Gehalt zwischen den Arten ($< 30\% \cdots > 70\%$)
relativ homogen innerhalb von Genomen (Ausnahmen: horizontaler Gentransfer[2], Prophagen)

höhere Eukaryoten (Säuger, Vögel) sehr starke Variationen des GC–Gehaltes ("Isochoren")
wirkt so, als sei es aus Bakteriengenomen zusammengesetzt
Hypothesen für die Herkunft der Isochoren:

1. Selektion (*selectionist hypothesis*, [55])
   Temperaturstabilität, Gene bevorzugt in GC–reichen Regionen
   Bernardi et al. [9, 8], Bernardi [7]
2. Mutationsvariabilität (*mutationalist hypothesis*, [55])
   Reparaturenzyme, frühe/späte Replikation, translation–coupled repair, DNA–Methylierung[3]
   Filipski [32], Sueoka [88, 89], Wolfe et al. [94], Holmquist et al. [46]

[2] Der Einfluß des horizontalen Gentransfers ist umstritten, da keine Stammbaumrekonstruktion möglich ist, wenn er zu hoch ist.

[3] unterdrückt die Mutation von C über U nach T

7.3 Alignment homologer Sequenzen

“mutation pattern”

Mutationsmatrix ⇒ Markov–Prozeß

$$\bar{p}_{t+\Delta t} = M \cdot \bar{p}_t$$

Sueoka (1962) Sueoka [87, 88]

$$\overset{(S)}{G|C} \quad \underset{v}{\overset{u}{\rightleftarrows}} \quad \overset{(S)}{A|T}$$

$$p_{n+1} = p_n - u \cdot p_n + v(1 - p_n) \qquad \text{lineare, iterierte Abb.}$$

$$p_\infty = \frac{v}{u+v}$$

Störung:

$$\Delta p_{n+1} = (1 - u - v) \cdot \Delta p_n \qquad \text{geometrische Folge}$$

Kapitel 8

Sequenzalignment

Nature is a tinkerer and not an inventor.

Jacob [49]

8.0 Motivation

> New sequences are adapted from pre–existing sequences rather than invented *de novo*. This is very fortunate for computational sequence analysis. We can often recognise a significant similarity between a new sequence and a sequence about which something is already known; when we do this we can transfer information about structure and/or function to the new sequence. We say that the two related sequences are *homologous* and that we are transfering information *by homology*. [21, p. 2]

> Many of the most powerful sequence analysis methods are now based on principles of probabilistic modelling. Examples of such methods include the use of probabilistically derived score matrices to determine the significance of sequence alignments, the use of hidden Markov models as the basis for profile searches to identify distant members of sequence families, and the inference of phylogenetic trees using maximum likelihood approaches. [21, *cover text*]

> The optimality of pairwise alignments between two sequences is not given by some canonical or unique criterion with universal applicability throughout the entire domain of sequences. The matches produced by alignment algorithms depend entirely on the parameters quantiatively defining the similarity of corresponding monomers, the cost of gaps and deletions, and most notably whether the algorithms are designed to optimize a score globally or locally.
>
> [...]
>
> Classical alignment algorithms are based on dynamic programming—for optimal global alignments, the Needleman–Wunsch algorithm [401,481], and for optimal local alignments, the Smith–Waterman algorithm [492]. [1, p. 34]

typisches Problem in der Bioinformatik

```
(a)   HBA_HUMAN   GSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKL
                  G+ +VK+HGKKV  A+++++AH+D++ +++++LS+LH  KL
      HBB_HUMAN   GNPKVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKL

(b)   HBA_HUMAN   GSAQVKGHGKKVADALTNAVAHV---D--DMPNALSALSDLHAHKL
                  ++ ++++h+ KV   + +A  ++          +L+ L+++H+ K
      LGB2_LUPLU  NNPELQAHAGKVFKLVYEAAIQLQVTGVVVTDATLKNLGSVHVSKG

(c)   HBA_HUMAN   GSAQVKGHGKKVADALTNAVAHVDDMPNALSALSD----LHAHKL
                  GS+ + G +   +D L  ++ H+ D+  A +AL D    ++AH+
      F11G11.2    GSGYLVGDSLTFVDLL--VAQHTADLLAANAALLDEFPQFKAHQE
```

Abb. 8.1: Drei Sequenzalignments eines Fragmentes von menschlichem α–Globin. (a) Klare Ähnlichkeit zu menschlichem β–Globin (b) Ein strukturell plausibles Alignment zum Leghämoglobin der gelben Lupine (*Lupinus luteus L.*). (c) Ein falsches *high–scoring* Alignment zu einem Glutathion S–Transferase Homolog eines Nematoden (F11G11.2). Aus [21]

- neue Sequenz liegt vor: Was ist die Funktion
- neues Genom liegt vor: Was kann ich über den Organismus lernen?

Vorgehensweise: Übertrage bekanntes Vorwissen auf die neue Sequenz

Annahme: Ähnliche Sequenzen haben ähnliche Funktion (Homologie–Annahme). Rechtfertigung dieser Annahme über die gemeinsame Abstammung.

Aufgabe: inexact pattern matching

Typische Veränderungen in der Evolution

(a) Punktmutationen (singe site nuclear polymorphisms)

(b) Insertionen/Deletionen

(c) Verschiebung von Teilsequenzen (shuffling)
meist größere Einheiten (Domäne in Protein, ganzes Gen)
wird meist beim Alignment ignoriert

▲ **Alignment** (Ausrichtung) ist die buchstabenweise Gegenüberstellung zweier oder mehrerer Sequenzen.

8.1 Bewertungsfunktionen für den Sequenzvergleich: Das Scoring Modell

Wie schon der Abschnitt über die Markov–Modelle (Abschnitt 5.1, S. 35) folgt die Darstellung in diesem Kapitel in weiten Teilen den entsprechenden Ausführungen von Durbin et al. [21]. Dieses Buch sei zur weiterführenden Lektüre sehr empfohlen, da es einen gut lesbaren Einstieg in die Materie gibt.

Der Gesamtscore eines Alignments ist die Summe der Terme des Scores für jedes einzelne Paar von Buchstaben der Sequenz und der Terme für jede Lücke. In der wahrscheinlichkeitstheoretischen Betrachtung sind das die Logarithmen des Vergleichs der relativen Wahrscheinlichkeiten, daß (a) die beiden Sequenzen verwandt bzw. (b) nicht verwandt sind.

Übereinstimmungen und konservative Substitutionen werden in Alignments häufiger erwartet als es dem Zufall entspräche und sollten deshalb positive Scorewerte beisteuern. Analog dazu sollten in einem realen Alignment nichtkonservative Veränderungen seltener auftreten als in einer Zufallssequenz. Sie tragen daher mit negativen Scorewerten zum Gesamtscore bei.

Die Verwendung von additiven Scoring–Modellen (Bewertungsfunktionen) bei den im folgenden vorgestellten Alignment–Verfahren impliziert die Annahme daß Mutationen an verschiedenen Orten auf der Sequenz als unabhängige Ereignisse betrachtet werden können (eine Lücke beliebiger Länge wird dabei als einzelne Mutation gezählt). Diese Annahme ist für DNA– und Proteinsequenzen durchaus vernünftig, auch wenn bekannt ist, daß Wechselwirkungen zwischen Aminosäure–Resten in Proteinen eine große Rolle spielen. Der Ansatz versagt dagegen völlig beim Alignment von RNA–Strukturen, da hier Basenpaarungen zu Abhängigkeiten über weite Bereiche der Sequenzen führen.

8.1.1 Ähnlichkeitsmatrizen (substitution matrices)

Betrachtet werden zwei Sequenzen x und y mit den Längen n und m. Dabei sei x_i das i–te Symbol von x und entsprechend y_j das j–te Symbol von y. Diese Symbole kommen aus einem Alphabet $\mathcal{A}$. Für die DNA ist $\mathcal{A} = \{\mathtt{A}, \mathtt{G}, \mathtt{C}, \mathtt{T}\}$, für Proteine enthält $\mathcal{A}$ die zwanzig proteinogenen Aminosäuren. Symbole aus einem Alphabet werden mit kleinen Buchstaben (a, b) bezeichnet.

Ziel des Alignments ist, eine quantitative Aussage darüber treffen zu können, mit welcher relativen Wahrscheinlichkeit zwei Sequenzen verwandt miteinander sind. Dazu werden zwei Wahrscheinlichkeitsmodelle für jeden der beiden Fälle (verwandt, nicht verwandt) genutzt, die jedem Alignment in beiden Fällen eine Wahrscheinlichkeit zuordnen. Anschließend werden die Verhältnisse (Quotienten) beider Wahrscheinlichkeiten betrachtet.

Das Wahrscheinlichkeitsmodell für den Fall, daß beide Sequenzen nicht miteinander verwandt sind, ist einfach: Es geht vom zufälligen, unabhängigen Auftreten des Buchstabens a mit einer Häufigkeit q_a aus (und wird daher auch *random model* genannt). Wegen der Unabhängigkeit jedes Buchstabens ist die Wahrscneinlichkeit der zwei Sequenzen das Produkt der Wahrscheinlichkeiten für jeden Buchstaben:

$$P(x, y|R) = \prod_i q_{x_i} \prod_j q_{y_j} \tag{8.1}$$

Alternativ dazu treten die Alignment–Paare im Modell für die Übereinstimmung (*match model*) mit einer Verbundwahrscheinlichkeit p_{ab} auf. Daraus folgt als Wahrscheinlichkeit des Gesamt–Alignments:

$$P(x, y|M) = \prod_i p_{x_i y_i} \tag{8.2}$$

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	**5**	−2	−1	−2	−1	−1	−1	0	−2	−1	−2	−1	−1	−3	−1	1	0	−3	−2	0
R	−2	**7**	−1	−2	−4	1	0	−3	0	−4	−3	3	−2	−3	−3	−1	−1	−3	−1	−3
N	−1	−1	**7**	2	−2	0	0	0	1	−3	−4	0	−2	−4	−2	1	0	−4	−2	−3
D	−2	−2	2	**8**	−4	0	2	−1	−1	−4	−4	−1	−4	−5	−1	0	−1	−5	−3	−4
C	−1	−4	−2	−4	**13**	−3	−3	−3	−3	−2	−2	−3	−2	−2	−4	−1	−1	−5	−3	−1
Q	−1	1	0	0	−3	**7**	2	−2	1	−3	−2	2	0	−4	−1	0	−1	−1	−1	−3
E	−1	0	0	2	−3	2	**6**	−3	0	−4	−3	1	−2	−3	−1	−1	−1	−3	−2	−3
G	0	−3	0	−1	−3	−2	−3	**8**	−2	−4	−4	−2	−3	−4	−2	0	−2	−3	−3	−4
H	−2	0	1	−1	−3	1	0	−2	**10**	−4	−3	0	−1	−1	−2	−1	−2	−3	2	−4
I	−1	−4	−3	−4	−2	−3	−4	−4	−4	**5**	2	−3	2	0	−3	−3	−1	−3	−1	4
L	−2	−3	−4	−4	−2	−2	−3	−4	−3	2	**5**	−3	3	1	−4	−3	−1	−2	−1	1
K	−1	3	0	−1	−3	2	1	−2	0	−3	−3	**6**	−2	−4	−1	0	−1	−3	−2	−3
M	−1	−2	−2	−4	−2	0	−2	−3	−1	2	3	−2	**7**	0	−3	−2	−1	−1	0	1
F	−3	−3	−4	−5	−2	−4	−3	−4	−1	0	1	−4	0	**8**	−4	−3	−2	1	4	−1
P	−1	−3	−2	−1	−4	−1	−1	−2	−2	−3	−4	−1	−3	−4	**10**	−1	−1	−4	−3	−3
S	1	−1	1	0	−1	0	−1	0	−1	−3	−3	0	−2	−3	−1	**5**	2	−4	−2	−2
T	0	−1	0	−1	−1	−1	−1	−2	−2	−1	−1	−1	−1	−2	−1	2	**5**	−3	−2	0
W	−3	−3	−4	−5	−5	−1	−3	−3	−3	−3	−2	−3	−1	1	−4	−4	−3	**15**	2	−3
Y	−2	−1	−2	−3	−3	−1	−2	−3	2	−1	−1	−2	0	4	−3	−2	−2	2	**8**	−1
V	0	−3	−3	−4	−1	−3	−3	−4	−4	4	1	−3	1	−1	−3	−2	0	−3	−1	**5**

Tab. 8.1: The BLOSUM50 substitution matrix. The log–odds values have been scaled and rounded to the nearest integer for purposes of computational efficiency. Entries on the main diagonal for identical residue pairs are highlighted bold. From [21]

Der Quotient dieser beiden Wahrscheinlichkeiten wird als *odds ratio* ("Unterschiedlichkeits–Quotient") bezeichnet:

$$\frac{P(x,y|M)}{P(x,y|R)} = \frac{\prod_i p_{x_i y_i}}{\prod_i q_{x_i} \prod_j q_{y_j}} = \prod_i \frac{p_{x_i y_i}}{q_{x_i} q_{y_i}}$$

Um ein additives Bewertungssystem (*additive scoring system*) zu erhalten, wird jetzt noch der Logarithmus dieses Quotienten genommen, auch *log–odds ratio* oder Score genannt:

$$S = \sum_i s(x_i, y_i) \tag{8.3}$$

mit

$$s(a,b) = \log\left(\frac{p_{ab}}{q_a q_b}\right) \tag{8.4}$$

als dem Logarithmus des Quotienten der Wahrscheinlichkeiten, daß das Symbol–Paar (a, b) als*aligned pair* erscheint, im Gegensatz zu einem *unaligned pair*.

Wie eingangs gefordert ist Gl. (8.3) die Summe der einzelnen Scores $s(a, b)$ für jedes Paar des Alignments. Diese Scores können in einer Matrix angeordnet werden. Für Proteine ergibt

sich z.B. eine 20×20 Matrix mit $s(a_i, a_j)$ in den Positionen i, j der Matrix (a_i, a_j sind dabei die i–te bzw. j–te Aminosäure in einer vorher definierten Reihenfolge) Das entspricht 210 unabhängigen Parametern. Eine solche Matrix wird als **Scoring Matrix** (Ähnlichkeitsmatrix, *substitution matrix*) bezeichnet. Ein Beispiel einer solchen Matrix ist die BLOSUM50–Matrix (vgl. Tab. 8.1). Eine solche Matrix macht Aussagen darüber, wie groß die Wahrscheinlichkeit ist, ein Paar ab in einem realen Alignment anzutreffen.

Einer solchen Matrix liegen verschiedene Annahmen zugrunde: zum einen physikalische bzw. biologische Aussagen über die verglichenen Moleküle, zum anderen evolutionär interpretiert die Erwartung der Substitution von Nukleotiden (und in der Folge von Aminosäuren).

Bei Nukleotidsequenzen unterscheiden wir dabei zwischen Transition (Purin $\leftrightarrow$ Purin, Pyrimidin $\leftrightarrow$ Pyrimidin, also: A $\leftrightarrow$ G, C $\leftrightarrow$ T) und Transversion (Purin $\leftrightarrow$ Pyrimidin).[1] Bei Aminosäuresequenzen spielen hysikochemische Ähnlichkeiten wie Größe, Polarität und Ladung eine Rolle. Da die Bewertung dieser Kriterien in hohem Maße der Beliebigkeit unterliegt, gibt es entsprechend viele verschiedene Bewertungsmatrizen.

Eine weitere Möglichkeit ist die statistische Analyse. Zwei prominente Beispiele für Aminosäuresequenzen sind die PAM Matrix (M. Dayhoff, [18]), die auf einem Alignment von verschiedenen Cytochrom–Sequenzen beruht, und die BLOSUM Matrix [42] auf der Basis lokaler Alignments von Blöcken.

8.1.2 Additive Bewertungsfunktion ohne Lücken

einfachstes Schema: zähle die Übereinstimmungen (Hamming–Distanz)

besser: Scoring matrix

$$\mathrm{SIM}(X, X') = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

$$\mathrm{SCORE} = \sum_{\substack{\text{aligned} \\ \text{pairs}}} \mathrm{SIM}(X_k, X_{k'})$$

8.1.3 Additive Bewertungsfunktion mit Lücken

$$\mathrm{SCORE} = \sum_{\substack{\text{aligned} \\ \text{pairs}}} \mathrm{SIM}(X_k, X_{k'}) + \mathrm{GAPSCORE}$$

(a) lineare Bewertungsfunktion

$$\mathrm{GAPSCORE} = \sum_{\text{gaps}} \mathrm{length(gap)} - \mathrm{GAP}$$

$$\gamma(g) = -gd$$

[1] Dabei ist die Wahrscheinlichkeit einer Transition (Purin $\leftrightarrow$ Purin, Pyrimidin $\leftrightarrow$ Pyrimidin) wesentlich höher, sie liegt bei $1 \cdot 10^{-10} \cdot \mathrm{a}^{-1} \cdot \mathrm{Base}^{-1}$ gegenüber $2 \cdot 10^{-9} \cdot \mathrm{a}^{-1} \cdot \mathrm{Base}^{-1}$ für eine Transversion

GAP ist die gap penalty (Bestrafung), etwa GAP $= -1$

(b) affine Bewertungsfunktion

$$\text{GAPSCORE} = \sum_{\text{gaps}} \text{length(gap)} \cdot \text{GEP} + \#\text{ gaps} \cdot \text{GOP}$$

$$\gamma(g) = -d - (g-1)e$$

GOP — gap opening penalty
GEP — gap extension penalty

Auch *gap penalties* entsprechen einem Wahrscheinlichkeitsmodell des Alignments: Wir nehmen an, daß die Wahrscheinlichkeit für eine Lücke an einer bestimmten Stelle in einer gegebenen Sequenz das Produkt einer Funktion $f(g)$ der Länge g der Lücke und der Verbundwahrscheinlichkeit der Menge der eingefügten Buchstaben ist:

$$P(\text{gap}) = f(g) \prod_{i \text{ in gap}} q_{x_i} \tag{8.5}$$

Das entspricht der Annahme, daß die Länge g der Lücke nicht mit den in ihr enthaltenen Buchstaben x_i korreliert ist.

Suche von Proteinen in DNA–Sequenzen Wegen der Degeneration des genetischen Codes (eine Aminosäure wird meist durch mehr als ein Nucleotid–Triplett codiert) können auch stark unterschiedliche Basenfolgen $(x_{\text{DNA}}, y_{\text{DNA}})$ auf der DNA für die gleiche Aminosäuresequenz $(x_{\text{AS}}, y_{\text{AS}})$ codieren. Ein Beispiel:

$$x_{\text{DNA}} = \texttt{CAC GCG TCC GAA} \longrightarrow x_{\text{AS}} = \text{HASE}$$
$$y_{\text{DNA}} = \texttt{CAT GCA AGT GAG} \longrightarrow y_{\text{AS}} = \text{HASE}$$

Die Übereinstimmungen auf DNA–Ebene entsprechen in diesem (sicherlich extremen) Beispiel 50%, die auf Aminosäureebene 100%. Solche Mutationen in der DNA, die ohne Einfluß auf das Genprodukt bleiben, werden als **silent mutations** bezeichnet.

8.2 Optimales globales Alignment: Needleman–Wunsch–Algorithmus

Die einfachste Form des Alignments ist der vollständige Vergleich zweier Sequenzen mit dem Ziel, die beiden Sequenzen so zu alignen (aneinander auszurichten), daß eine gegebene additive Bewertungsfunktion (vgl. Abschnitt 8.1) maximiert wird.

▲ **globales Alignment** ist die *vollständige* Ausrichtung zweier (oder mehrerer) Sequenzen

Für zwei Sequenzen der gleichen Länge n gibt es zunächst nur eine Möglichkeit des globalen Alignments der ganzen Sequenzen. Das ändert sich mit der Einführung von Lücken (*gaps*), wie das nachfolgende Beispiel zeigt.

■ **globales Alignment zweier Strings** Zwei kurze Aminosäuresequenzen

$$x = \texttt{ARMER} \qquad\qquad y = \texttt{HASE}$$

sollen global aligned werden. Dabei wird von einer statistischen Gleichverteilung der Aminosäuren in jeder Sequenz ausgegangen. Die Scoring Matrix wird nach folgendem Schema erstellt:

$$\mathrm{SIM}(x_i, y_j) = \delta(x_i, y_j) = \begin{cases} 1 & \forall\, x_i = y_j \\ -1 & \forall\, x_i \neq y_j \end{cases}$$

Lücken werden mit einem Score von -1 bestraft.

$$\mathrm{SIM}(x_i, -) = -1 = \mathrm{GAP}$$

Einige mögliche globale Alignments sehen wie folgt aus (jeweils mit dazugehörigem Score)

Alignment:	ARMER HASE_	ARMER _HASE	ARMER____ _____HASE	_ARMER HAS_E_
score:	-3	-5	-9	-2

Das letzte Alignment hat den niedrigsten Score und stellt damit ein bestes mögliche Alignment (das "S" von "HASE" kann verschoben werden, ohne den Score zu beeinflussen) dieser beiden Sequenzen dar.

8.2.1 Dynamic Programming Algorithm/Table

Für globale Alignments mit Lücken gibt es für zwei Sequenzen der Länge n

$$\binom{2n}{n} = \frac{(2n)!}{(n!)^2} \simeq \frac{2^{2n}}{\sqrt{2\pi n}} \tag{8.6}$$

Möglichkeiten, und es ist schlicht nicht möglich, diese alle zu berechnen. Hier kommt die Technik des *dynamic programming* (Dynamischen Programmierung, D.P.) ins Spiel.

▲ **Dynamic programming** [3, 10] ist sehr allgemein gesprochen eine Optimierungstechnik, die immer dann angewandt werden kann, wenn sich ein Problem rekursiv in zwei ähnliche, kleinere Unterprobleme zerlegt werden kann, so daß die Lösung des größeren Problems durch Zusammensetzen der Lösungen der beiden Unterprobleme erreicht werden kann. [1]

Das Wesen der *dynamic programming* ist Richard Bellmans "Prinzip der Optimalität". Dieses Prinzip ist, auch ohne strikte Definition der Begriffe, intuitiv:

> Eine optimale Strategie hat die Eigenschaft, daß die verbleibenden Entscheidungen eine optimale Strategie im Hinblick auf den aus der ersten Entscheidung resultierenden Zustand bilden müssen, unabhängig vom anfänglichen Zustand und der anfänglichen Entscheidung.

	H	E	A	G	A	W	G	H	E	E
P	−2	−1	−1	−2	−1	−4	−2	−2	−1	−1
A	−2	−1	**5**	0	**5**	−3	0	−2	−1	−1
W	−3	−3	−3	−3	−3	**15**	−3	−3	−3	−3
H	**10**	0	−2	−2	−2	−3	−2	**10**	0	0
E	0	**6**	−1	−3	−1	−3	−3	0	**6**	**6**
A	−2	−1	**5**	0	**5**	−3	0	−2	−1	−1
E	0	**6**	−1	−3	−1	−3	−3	0	**6**	**6**

Tab. 8.2: Scoring–Matrix der zwei Beispielsequenzen, die für die Illustration der *dynamic programming* Alignment Algorithmen benutzt werden. Gezeigt ist eine Matrix zugehöriger BLOSUM50–Werte für jedes Alignment zweier Aminosäuren der beiden Sequenzen. Positive Scores (identische oder konservierte Aminosäurereste) sind durch Fettdruck hervorgehoben. Aus [21]

[`http://benli.bcc.bilkent.edu.tr/~omer/research/dynprog.html`]

Alignment Algorithmen können als Lösung des Problems dargestellt werden, den kürzesten Weg im zugehörigen Graphen mit der zugehörigen Metrik (den Abständen) zu finden. Das Alignment zweier Sequenzen der Länge n erfordert, den kürzesten Weg in einem Graphen mit n^2 Eckpunkten zu finden. Da bei *dynamis programming* jeder solcher Eckpunkte einmal durchlaufen wird, ist die zeitliche Komplexität eines solchen Algorithmus $\mathcal{O}(n^2)$. [1]

Dynamic programming erfolgt immer in drei Schritten: (i) Initialisierung, (ii) Füllen der Matrix (*Matrix fill*, Scoring) und (iii) Traceback (Alignment).

8.2.2 Needleman–Wunsch Verfahren zum globalen Alignment

Der *dynamic programming* Algorithmus für das optimale globale Alignment zweier Sequenzen (mit Lücken) ist als Needleman–Wunsch algorithm [63] bekannt. Die hier vorgestellte, verbesserte Form stammt von Gotoh [37].

Der Algorithmus soll an zwei kurzen Aminosäuresequenzen (`HEAGAWGHEE` und `PAWHEAE`) verdeutlicht werden. Für das Scoring wurde eine BLOSUM50 Matrix verwendet (vgl. Tab. 8.2), die Kosten für ein *gap* seien $d = -8$ pro Aminosäure.

Tab. 8.2 zeigt eine Matrix $\boldsymbol{S}_{ij}$ der lokalen Scores $s(x_i, y_j)$ für jedes mögliche Alignment zweier Aminosäuren der beiden Sequenzen. Identische oder konservierte Aminosäurereste sind durch Fettdruck hervorgehoben. Einfach gesagt ist es das Ziel eines Alignment Algorithmus, möglichst viele dieser positiven Scorings in das Alignment zu integrieren und gleichzeitig die Kosten von nichtkonservierten Resten, *gaps* und anderen Bedingungen zu minimieren.

Der Kern des Algorithmus ist der Aufbau einer Matrix $\boldsymbol{F}$. Jeder Wert $\boldsymbol{F}(i, j)$ entspricht dabei dem Score des besten Alignments zwischen dem anfänglichen Segment $x_{1\ldots i}$ von x bis x_i und dem anfänglichen Segment $y_{1\ldots j}$ von y bis y_i. Die Matrix $\boldsymbol{F}$ kann rekursiv erstellt werden.

Begonnen wird mit der **Initialisierung** der Matrix: $\boldsymbol{F}(0, 0) = 0$.

Füllen der Matrix

Im nächsten Schritt wird die **Matrix** dann sukzessive von links oben nach rechts unten **gefüllt**. Sind die Werte für $\boldsymbol{F}(i-1, j-1)$, $\boldsymbol{F}(i-1, j)$ und $\boldsymbol{F}(i, j-1)$ bekannt, kann der Wert $\boldsymbol{F}(i, j)$ berechnet werden. Es gibt drei Möglichkeiten, den besten Wert für das Alingment von x_i, y_j zu erreichen:

1. x_i wird mit y_j aligned, $\boldsymbol{F}(i, j) = \boldsymbol{F}(i-1, j-1) + s(x_i, y_j)$
2. x_i wird mit einem *gap* aligned, $\boldsymbol{F}(i, j) = \boldsymbol{F}(i-1, j) - d$
3. y_j wird mit einem *gap* aligned, $\boldsymbol{F}(i, j) = \boldsymbol{F}(i, j-1) - d$

Die Möglichkeit mit dem größten resultierenden Wert entspricht dem besten Score für (i, j). Es gilt also:

$$\boldsymbol{F}(i, j) = \max \begin{cases} F(i-1, j) - d \\ F(i-1, j-1) + s(x_i, y_j) \\ F(i, j-1) - d \end{cases} \tag{8.7}$$

Dabei entspricht d der Strafe für ein *gap*, $s(x_i, y_j)$ dem Score für das Alignment von x_i, y_j entsprechend der Scoring Matrix (vgl. Tab. 8.2).

Mit dem Füllen der Matrix mit den Werten $\boldsymbol{F}(i, j)$ wird gleichzeitig für jede Zelle ein Zeiger eingeführt, der auf die Zelle zeigt, aus der $\boldsymbol{F}(i, j)$ berechnet wurde (vgl. die *dynamic programming matrix*, Tab. 8.3).

Zur Vervollständigung gilt es, noch einige Randbedingungen zu klären: In der obersten Zeile der *dynamic programming matrix* gilt $j = 0$, d.h. die Werte $\boldsymbol{F}(i, j-1)$ und $\boldsymbol{F}(i-1, j-1)$ sind nicht definiert. Daher müssen die Werte $\boldsymbol{F}(i, 0)$ gesondert behandelt werden. Da sie ein Alignment zwischen einem Präfix von x und y darstellen, gilt $\boldsymbol{F}(i, 0) = -id$ und analog dazu für die erste Spalte $\boldsymbol{F}(0, j) = -jd$.

Der **Wert in der letzten Zelle der Matrix** (rechts unten), $\boldsymbol{F}(n, m)$, ist *per definitionem* der beste Score für das Alignment der Sequenzen $x_{1\ldots n}$ und $y_{1\ldots m}$. Damit ist das Ziel erreicht: Der **beste Score für das globale Alignment** von x und y.

Traceback (backtracking)

Um das Alignment selbst zu finden, wird der Pfad von Wahlen gemäß Gl. (8.7) gesucht, der zum letzten Wert in der Matrix führte. Dieses Verfahren wird als **Traceback** oder **backtracking** bezeichnet.

▲ **Traceback, backtracking** Rekonstuktion der optimalen Pfade im Dynamic Programming Table

Die Durchführung ist einfach: Gestartet wird mit dem finalen Wert der Matrix ($\boldsymbol{F}(n, m)$). Von hier ausgehend werden die vorher bei der Generierung der Matrix erstellten Zeiger verfolgt. An jeder Stelle im Prozeß gibt es drei mögliche Schritte: $(i-1, j-1)$, $(i-1, j)$ und $(i, j-1)$. Gleichzeitig wird jeweils ein Paar von Symbolen zum Alignment hinzugefügt: x_i und y_j für den Schritt $(i-1, j-1)$, x_i und das *gap*–Symbol '$-$' für $(i-1, j)$ oder '$-$' und y_j für $(i, j-1)$. Der Endpunkt des Traceback ist $i = j = 0$.

		H	E	A	G	A	W	G	H	E	E
	0	**−8**	**−16**	−24	−32	−40	−48	−56	−64	−72	−80
P	−8	−2	−9	**−17**	**−25**	−33	−42	−49	−57	−65	−73
A	−16	−10	−3	−4	−12	**−20**	−28	−36	−44	−52	−60
W	−24	−18	−11	−6	−7	−15	**−5**	**−13**	−21	−29	−37
H	−32	−14	−18	−13	−8	−9	−13	−7	**−3**	−11	−19
E	−40	−22	−8	−16	−16	−9	−12	−15	−7	**3**	−5
A	−48	−30	−16	−3	−11	−11	−12	−12	−15	**−5**	2
E	−56	−38	−24	−11	−6	−12	−14	−15	−12	−9	**1**

HEAGAWGHE-E
--P-AW-HEAE

Tab. 8.3: Above, the global dynamic programming matrix for our example sequences, with arrows indication traceback pointers; values on the optimal alignment path are shown in bold. Below, a corresponding optimal alignment, which has total score 1. From [21]

➤ Jedes Alignment kann als Pfad in einem Alignment Graphen (*dynamic programming matrix*) dargestellt werden. Die Position i, j im Graphen entspricht einem Alignment der Teilwörter $x_{1\ldots i}$ und $y_{1\ldots j}$.

Der Grund, warum dieser Algorithmus funktioniert, ist die Tatsache, daß der Score aus der Summe unabhängiger Teilstücke berechnet wird, so daß der beste Score bis zu einem bestimmten Punkt im Alignment der Summe des besten Scores bis einen Punkt zuvor und dem Score des neuen Schrittes entspricht.

8.2.3 Komplexität von Algorithmen (*big–O notation*)

In der Praxis ist es vorteilhaft zu wissen, wie sich Rechenzeit und Speicherbedarf eines Algorithmus mi zunehmender Größe der bearbeiteten Sequenzen verhalten. Für den Needleman–Wunsch–Algorithmus gilt z.B., daß $(n + 1) \times (m + 1)$ Zahlen gespeichert werden und jede Zahl eine konstante Zahl an Rechenoperationen erfordert (drei Summen und ein Maximum). Der Algorithmus benötigt also $\mathcal{O}(nm)$ Rechenzeit und $\mathcal{O}(nm)$ Speicherkapazität, mit den Längen n und m der Sequenzen. Die Notation $\mathcal{O}(nm)$ (*big–O notation*) bedeutet, daß der Algorithmus von der Ordnung nm ist. Die für die Berechnung notwendige Rechenzeit und der Speicherbedarf sind also vom Produkt der Länge der beiden Sequenzen abhängig. Da die Sequenzen meist vergleichbare Längen haben, wird die Ordnung des Algorithmus meist kurz als $\mathcal{O}(n^2)$ geschrieben.

Je größer der Exponent ist, desto weniger praktikabel ist die Methode für große Exponenten. Als Faustregel kann gelten:

➤ Für biologische Sequenzen und Standard–Computer sind $\mathcal{O}(n^2)$ Algorithmen durchführbar, wenn auch etwas langsam. $\mathcal{O}(n^3)$ Algorithmen sind dagenen nur für sehr kurze Sequenzen praktisch anwendbar.

8.3 Optimales lokales Alignment: Smith–Waterman–Algorithmus

8.3.1 Motivation

▲ **lokales Alignment** Suche eines möglichst gut passenden (optimalen) Teilwortes (substring) in den Sequenzen x und y

Einsatz: eigentlich typsischer als globales Alignment

- globales Alignment: Proteinfamilie
  alle Sequenzen ± gleich lang
- lokales Alignment: Motiv in einem Protein, Sequenz in einer Datenbank

naive Abschätzung der Komplexität: alle möglichen globalen Alignments von Teilwörtern werden ausprobiert

8.3.2 Smith–Waterman Verfahren zum lokalen Alignment

Smith und Waterman [80]

Vorgehensweise: Berechne den Dynamic Programming Table wie in Abschnitt 8.2.1

terminale gaps sind umsonst

$$\boldsymbol{F}(i0) = \boldsymbol{F}(0j) = 0$$

$$\boldsymbol{F}(i,j) = \max \begin{cases} F(i, j-i) - d \\ F(i-1, j) - d \\ F(i+1, j+1) + s(x_i, y_j) \\ 0 \end{cases} \tag{8.8}$$

Backtracking

- Starte vom größten Score–Wert im Dynamic Programming Table
- verfolge alle optimalen Pfade, bis der Score 0 erreicht ist

zwei entscheidende Änderungen gegenüber dem globalen Alignment

1. Start vom größten Score
2. Stop bei Score=0

		H	E	A	G	A	W	G	H	E	E
	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	5	0	**5**	0	0	0	0	0
W	0	0	0	0	2	0	**20**	**12**	4	0	0
H	0	10	2	0	0	0	12	18	**22**	14	6
E	0	2	16	8	0	0	4	10	18	**28**	20
A	0	0	8	21	13	5	0	4	10	20	27
E	0	0	6	13	18	12	4	0	4	16	26

AWGHE
AW-HE

Tab. 8.4: Above, the local dynamic programming matrix for the example sequences. Below, the optimal local alignment, which score 28. In this case the local alignment is a subset of the global alignment, but that is not always the case. From [21]

8.4 Heuristische Alignment Algorithmen

8.4.1 BLAST

8.4.2 FASTA

Kapitel 9

Stammbaumrekonstruktion

9.1 Einige wichtige Ideen zur Evolutionstheorie

1. Darwin und Neo–Darwinisten
   - Evolution geht in kleinen Schritten voran
   - treibender Faktor ist die Selektion
2. neutrale Theorie der Evolution, Kimura [52]
   treibender Faktor ist die Mutation (Zufall)
   random drift, silent mutations, polymorphisms
3. Genduplikation
   1 Kopie als Sicherheit, 1 Kopie "zum Spielen"
   Bsp.: Globin–Familie: Myoglobin, Hämoglobin
4. Gentransfer
   Transposons, Vektoren

1 und 2 lassen sich gut als Stammbaum darstellen. 4 entspricht eher einem allgemeinen Graphen.

neo–Darwinism

> In time, neo–Darwinism became a dogma in evolutionary biology, and selection came to be considered the only force capable of driving the evolutionary process, while other factors such as mutation and random drift where thought of as minor contributors at best. This particular brand of neo–Darwinism was called **selectionism**.
>
> According to the selectionist or neo–Darwinian perception of the evolutionary process, gene substitutions occur as a consequence of selection for advantageous mutations. Polymorphism, on the other hand, is maintained by balancing selection. Thus, neo–Darwinists regard substitution and polymorphism as two separate

phenomena driven by different evolutionary forces. Gene substitution is the end result of a positive adaptive process, whereby a new allele takes over future generations of the population of and only if it improves the fitness of the organism, while polymorphism is maintained when the coexistence of two or more alleles at a locus is advantageous for the organism or the population. Neo–Darwinian theories maintain that most genetic polymorphisms in nature are stable.

Li [55, p. 55]

Neutral theory of molecular evolution

In 1968, Kimura postulated that the majority of molecular changes in evolution are due to the random fixation of neutral or nearly neutral mutations [51]; it was also independently proposed by King and Jukes [53]. This hypothesis, now known as the **neutral theory of molecular evolution**, contends that at the molecular level the majority of evolutionary changes and much of the variability within species are caused neither by positive selection of advantageous alleles nor by balancing selection, but by random genetic drift of mutant alleles that are selectively neutral or nearly so. Neutrality, in the sense of the theory, does not imply strict equality in fitness for all alleles. It only means that the fate of alleles is determined largely by random genetic drift. In other words, selection may operate, but its intensity is too weak to offset the influences of chance effects. For this to be true, the absolute value of the selective advantage or disadvantage of an allele must be smaller than $1/(2N_e)$ (where N_e is the effective population size).

Li [55, p. 55]

Gene duplication

The evolutionary significance of gene duplication was first recognized by Haldane [40] (1932) and Muller [61] (1935), who suggested that a redundant duplicate of a gene may aquire divergent mutations and eventually emerge as a new gene. [...] Ohno [65] (1970) put forward the view that gene duplication is the only means by which a new gene can arise. Although other means of creating new genes or new functions are now known [...], Ohno's view remains largely valid.

Li [55, p. 269]

Transposition and horizontal transfer

"Jumping genes" (transposable genetic elements) were first discovered in maize by Barbara McClintock in the late 1940s. She found that genes associated with the development of color pigments in kernels could be turned on or off at abnormal times by the action of certain genetic elements, which she called "controlling elements," that apparently could move from site to site on different maize chromosomes. [...] With the great advances in molecular biology techniques in the 1970s and 1980s, transposition of genetic elements was found to be

> a widespread phenomenon not only in maize but also in other organisms. DNA sequences that possess an intrinsic capability to change their genomic location are called **mobile elements** or **transposable elements** (TEs). The new data stimulatiet vigorous debates of the biological significance of TEs. Since TEs can "jump about" in a genome and can move genetic materials from one genomic location to another, they can produce novel gene mutations an chromosomal rearrangements that cannot be produced by other known mechanisms.
>
> Li [55, p. 335]

9.2 Zeiten, Raten und Distanzen

9.2.1 Die Grundlage: Molekulare Uhr–Hypothese

Zuckerkandl und Pauling [96], Zuckerkandl [95]

▲ Jedes Protein sammelt im Laufe der Evolution mit gleichbleibender Rate Mutationen an.

> Molecular Clock Molecular clock hypothesis - The molecular clock is a hypothesis that mutation rates and substitution rates do not vary among lineages in a tree. Therefore, if all the lineages of a tree are from the same time they should all have the same genetic distance from the root. An extension of the molecular clock concept to sequences from different times implies that the distance of a particular sequence from the root of the tree should be proportional to the amount of time that has accumulated from the root to the sampling time of that sequence. Thus a plot of root-to-tip distances against sampling times should yield a positive linear correlation with a slope equal to the mutation rate. The molecular clock hypothesis is a fundamental assumption of all models in BEAST.
>
> `http://evolve.zoo.ox.ac.uk/beast/glossary.html`

9.2.2 Einige Beispiele

Ein–Sequenz–Population

Betrachtet werde eine einzelne Sequenz S über die Zeit t. Unter Annahme der Gültigkeit der Hypothese von Zuckerkandl und Pauling (Molekulare Uhr–Hypothese, s.o.) gilt: Die Zahl N_{mut} der Mutationen ist proportional zur Zeit t:

$$N_{\text{mut}} \propto t$$

Tatsächlich hängt die Zahl der Mutationen neben der Zeit auch noch von der Mutationsrate r und der Länge l der Sequenz ab, so daß gilt:

$$N_{\text{mut}}(\text{Sequenz}) = r \cdot l \cdot t \tag{9.1}$$

Die Zeit wird dabei in Jahren gemessen ($[t] = \mathrm{a}$), die Mutationsrate in Mutationen pro Buchstabe (hier: Basenpaar, bp) und Jahr ($[r] = \text{Mutationen}\cdot\mathrm{bp}^{-1}\cdot\mathrm{a}^{-1}$). Die Zeit steckt nicht implizit in diesem Modell, sie muß von außen aus anderen Quellen kommen (Paläontologie etc.).

Divergente Entwicklung zweier Sequenzen

Für die Stammbaumrekonstruktion ist es wesentlich realistischer, zwei Sequenzen miteinander zu vergleichen, als einzelne Sequenzen zu betrachten, wie oben. Hinzu kommt, daß die Ausgangssequenz in der Regel nicht zugänglich ist und wir von gegenwärtigen Sequenzen ausgehend extrapolieren müssen.

Die einfachste Art, den Abstand (Distanz) zweier Sequenzen (S, S') zu berechnen, ist, die Zahl der nicht übereinstimmenden Buchstaben zu zählen. Dieses Maß wird oft als Hamming–Distanz

$$d_{\mathrm{Ham}}(S, S') = \text{Zahl der ungleichen Buchstaben} \tag{9.2}$$

bezeichnet.

■ **Hamming–Distanz zweier kurzer DNA–Sequenzen** Gegeben sei die ursprüngliche Sequenz $S^0 =$ `AAGTAGA` und die Sequenz $S =$ `AACTAGC` in der Gegenwart, die sich von S^0 durch zwei Mutationen (an den Stellen 3 und 7) unterscheidet. Dann ist die Hamming–Distanz dieser beiden Sequenzen

$$d(S^0, S) = d\begin{pmatrix} \mathtt{A} & \mathtt{A} & \mathtt{G} & \mathtt{T} & \mathtt{A} & \mathtt{G} & \mathtt{A} \\ \mathtt{A} & \mathtt{A} & \mathtt{C} & \mathtt{T} & \mathtt{A} & \mathtt{G} & \mathtt{C} \end{pmatrix} = 2$$

Normalerweise ist S^0 *nicht* bekannt, nur zwei Sequenzen in der Gegenwart. Deshalb muß die Zeit, die von der ursprünglichen Sequenz bis zur Gegenwart vergangen ist, zweifach in die Bestimmung der Hamming–Distanz einfließen. Dann ergibt sich Gl. (9.1) zu

$$d(S, S') = r \cdot l \cdot 2 \cdot t \qquad\qquad r = \frac{d(S, S')}{2 \cdot l \cdot t} \tag{9.3}$$

Zur Bestimmung evolutionärer Distanzen von Sequenzen sind Hamming–Distanzen aber nur von bedingtem Nutzen. Auf Dauer ist der Abstand zwischen den Sequenzen nicht mehr proportional zur Zeit, da sich Substitutionen auf der gleichen Base wiederholen.[1] Die Zeit der Aufspaltung (t bzw. t_0) muß nach wie vor aus der Palöobiologie kommen, d.h. durch prähistorische Funde datiert werden. t_0 ist dabei der Zeitpunkt der divergenten Entwicklung.

Der älteste Ansatz, diese Schwäche zu umgehen und ein besseres Maß für die Distanz zweier Sequenzen zu finden, ist der von Jukes und Cantor [50], vgl. das Jukes–Cantor–Modell, Abschnitt 9.2.4, S. 76.

Einige Raten für Aminosäure–Substitution

Und hier noch realistische Daten für die Mutationsraten r von Nukleotiden auf der DNA. Dabei wird unterschieden zwischen **synonymen Substitutionen**, die keinen Einfluß auf

[1] Das Problem: Mehrfachsubstitutionen in derselben Position können *nicht* beobachtet werden!

die translatierte Aminosäure und damit das Protein haben[2] und **nichtsynonymen Substitutionen**, die zur Translation einer anderen Aminosäure führen[3].

Synonyme Substitutionen (typischerweise an der 3. Position im Codon)

$$r_{ss} = 3 - 5 \cdot 10^{-9} \cdot \mathrm{a}^{-1} \cdot \mathrm{bp}^{-1}$$

Nichtsynonyme Substitutionen

$$r_{ns} = 1 \cdot 10^{-9} \cdot \mathrm{a}^{-1} \cdot \mathrm{bp}^{-1}$$

Dieser auf den ersten Blick vielleicht verblüffende Unterschied wird meist dadurch erklärt, daß der Evolutionsdruck bei synonymen Substitutionen wesentlich geringer sei als bei nichtsynonymen Substitutionen. Daher rühre die wesentlich höhere Substitutionsrate für synonyme Substitutionen.

Einige Ergebnisse scheinen das zu belegen: Histone[4] sind z.B. fast vollständig konserviert ($r_{ns} \approx 0 \cdot 10^{-9} \cdot \mathrm{a}^{-1} \cdot \mathrm{bp}^{-1}$), Interferon γ weist dagegen eine sehr hohe Rate der nichtsynonymen Substitution von $r_{ns} = 3 \cdot 10^{-9} \cdot \mathrm{a}^{-1} \cdot \mathrm{bp}^{-1}$ auf.

9.2.3 Ein zwei–Buchstaben Markov–Modell

Für die Modellierung von Zufallssequenzen von Symbolen x_i aus einem Alphabet $\mathcal{A}$ können grundsätzlich Markov–Prozesse angewendet werden (vgl. Abschnitt 5.1, S. 35ff.). Unter der hier getroffenen Annahme, daß die Symbole sich unabhängig voneinander entwickeln, kann ein Markov–Prozeß erster Ordnung (Markov–Kette) genutzt werden.

Das hier besprochene Zwei–Buchstaben Markov–Modell ist zur Hinführung auf das etwas kompliziertere, aber nach dem gleichen Muster aufgebaute *Jukes–Cantor–Modell* (JC–Modell) gedacht. Das JC–Modell [50] dient der realen Simulation genomischer Sequenzen.

Gegeben sei ein Alphabet $\mathcal{A}$ mit zwei Buchstaben, $\mathcal{A} = \{A, B\}$. Jede Position in der Sequenz entwickele sich unabhängig und zufällig. Die Substitutionsrate (r) sei konstant. Eine graphische Darstellung des zugrundeliegenden Markov–Modells gibt Abb. 9.1a.

Da sich das System stochastisch verhält, können nur noch Wahrscheinlichkeitsaussagen über eine bestimmte Stelle in der Sequenz (d.h. den Zustand des Systems zu einer gegebenen Zeit t) gemacht werden:

$$A \to p_A$$
$$B \to p_B$$

[2] Aber: Unterschiedliche synonyme Codons für eine Aminosäure könnten durchaus eine unterschiedliche Fitneß haben, da die tRNA–Konzentrationen für die betreffenden Codons in der Zelle durchaus in unterschiedlichen Konzentrationen vorhanden sind und tRNAs, die mehrere synonyme Codons erkennen, unterschiedliche Bindungsaffinitäten zu den verschiedenen Codons könnten. [16, 72]

[3] Was nicht zwangsläufig mit gravierenden Folgen für das Protein enden muß, wenn die veränderte Aminosäure entweder ähnliche physikochemische Eigenschaften aufweist oder in einer strukturell unsensiblen Region des Proteins angesiedelt ist (die weder für Faltung noch Funktion große Bedeutung hat).

[4] Histone sind Proteine, die für die Sekundärstruktur der DNA von entscheidender Bedeutung sind und denen dadurch eine unverzichtbare Funktion zukommt.

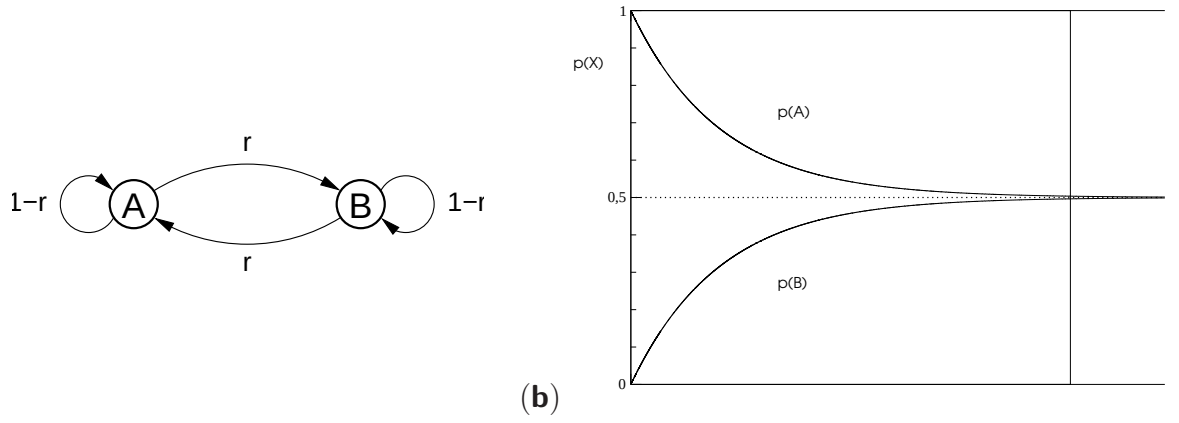

Abb. 9.1: Ein zwei–Buchstaben Markov–Modell für die evolutionäre Veränderung zweier Sequenzen. (**a**) Graphische Darstellung der dem Modell zugrundeliegenden Markov–Kette (vgl. auch Abb. 5.1, S. 36); (**b**) die zeitliche Entwicklung des Systems: Die Wahrscheinlichkeit für alle Basen läuft für $t \to \infty$ gegen $p_X(t) = \frac{1}{2}$, d.h. beide Buchstaben sind im Mittel für lange Zeiten gleich häufig.

Die Entwicklung des Systems ist durch einen Vektor von Wahrscheinlichkeiten

$$\vec{p} = \begin{pmatrix} p_A \\ p_B \end{pmatrix} \tag{9.4}$$

und eine Übergangsmatrix (Ratenmatrix)

$$\boldsymbol{M} = \begin{pmatrix} 1-r & r \\ r & 1-r \end{pmatrix} \tag{9.5}$$

bestimmt. Die Anfangsbedingung ist

$$p_A(0) = 1 \qquad\qquad p_B(0) = 0$$

was in Vektorschreibweise dem Vektor

$$\vec{p}(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} p_A \\ p_B \end{pmatrix} (t = 0) \tag{9.6}$$

für den Zustand $t = 0$ entspricht.

Die Zeitentwicklung des Systems läßt sich in dieser Form durch Anwendung der Übergangsmatrix $\boldsymbol{M}$ auf den Vektor $\vec{p}(0)$ der Wahrscheinlichkeiten berechnen. Damit ergibt sich für die Vektoren $\vec{p}(t)$ für den ersten Zeitschritt ($t = 1$):

$$\begin{aligned} \vec{p}(1) &= \boldsymbol{M}\vec{p}(0) \\ &= \begin{pmatrix} 1-r & r \\ r & 1-r \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} 1-r+0-1 \\ r+0(1-r) \end{pmatrix} = \begin{pmatrix} 1-r \\ r \end{pmatrix} \\ &= \begin{pmatrix} p_A(1) \\ p_B(1) \end{pmatrix} \end{aligned}$$

und für den zweiten Zeitschritt ($t = 2$):

$$\begin{aligned}\vec{p}(2) &= \boldsymbol{M}\vec{p}(1) \\ &= \begin{pmatrix} 1-r & r \\ r & 1-r \end{pmatrix}\begin{pmatrix} 1-r \\ r \end{pmatrix} \\ &= \begin{pmatrix} (1-r)^2 + r^2 \\ r(1-r)\cdot 2 \end{pmatrix} \\ &= \begin{pmatrix} 1-2r+2r^2 \\ 2(r-r^2) \end{pmatrix}\end{aligned}$$

Ein wesentlich eleganterer Weg als die Berechnung obige Berechnung ist die Verwendung der Eigenvektoren und Eigenwerte der Matrix $\boldsymbol{M}$.

Zur Matrix $\boldsymbol{M}$ gehört ein Satz von Eigenvektoren $\vec{e}$ und Eigenwerten λ, so daß gilt:

$$\boldsymbol{M}\cdot\vec{e} = \lambda\cdot\vec{e} \tag{9.7}$$

Die beiden Eigenvektoren $\vec{e}$ und ihre dazugehörigen Eigenwerte λ lauten für die Übergangsmatrix $\boldsymbol{M}$:

$$\begin{aligned} \boldsymbol{M}\vec{e}_+ &= \lambda\cdot\vec{e}_+ & \qquad \boldsymbol{M}\vec{e}_- &= \lambda\cdot\vec{e}_- \\ \vec{e}_+ &= \begin{pmatrix} 1 \\ 1 \end{pmatrix} & \qquad \vec{e}_- &= \begin{pmatrix} 1 \\ -1 \end{pmatrix} \\ \lambda_+ &= 1 & \qquad \lambda_- &= 1-2r \end{aligned} \tag{9.8}$$

Ein beliebiger (Start–)Vektor läßt sich in eine Linearkombination der beiden Eigenvektoren $\vec{e}_+$ und $\vec{e}_-$ zerlegen:

$$\vec{p}_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{2}\left(\vec{e}_+ + \vec{e}_-\right) \tag{9.9}$$

Die **Zeitentwicklung der Eigenvektoren** soll für jeden der beiden Eigenvektoren getrennt betrachtet werden. Sie wird wieder dadurch Anwendung von $\boldsymbol{M}$ auf den jeweiligen Eigenvektor $\vec{e}$ berechnet. Für $\vec{e}_+$ ergibt sich für den ersten Zeitschritt ($t = 1$):

$$\vec{e}_+(1) = \boldsymbol{M}\vec{e}_+ = \lambda_+\cdot\vec{e}_+ = \vec{e}_+$$

und da in diesem Fall gilt, daß $\lambda_+^t = \lambda_+$, für beliebige Zeiten t

$$\vec{e}_+(t) = \boldsymbol{M}^t\vec{e}_+ = \lambda_+^t\cdot\vec{e}_+ = \vec{e}_+ \tag{9.10}$$

Für den Eigenvektor $\vec{e}_-$ ergibt sich die zeitliche Entwicklung für beliebige Zeiten t zu:

$$\vec{e}_-(t) = \boldsymbol{M}^t\vec{e}_- = \boldsymbol{M}^{t-1}\boldsymbol{M}\vec{e}_- \tag{9.11}$$

Da weiterhin gilt, daß sich $\boldsymbol{M}\vec{e}_-$ umschreiben läßt zu

$$\boldsymbol{M}\vec{e}_- = \lambda_{-1}\vec{e}_- \tag{9.12}$$

ergibt sich der Eigenvektor $\vec{e}_(t)$ zu

$$\vec{e}_-(t) = \lambda_- \boldsymbol{M}^{t-1}\vec{e}_- = \lambda_-^t \vec{e}_- \tag{9.13}$$

und durch Einsetzen der Definition von λ_- (Gl. 9.8) zu

$$\vec{e}_-(t) = (1-2r)^t\vec{e}_- = \exp(t \cdot \ln(1-2r))\vec{e}_- \tag{9.14}$$

Unter der (biologisch gut motivierbaren[5]) Annahme, daß für die Mutationsrate r gilt, daß $r \ll 1$, ergibt sich schließlich

$$\vec{e}_-(t) \simeq \exp(-t \cdot 2r)\vec{e}_- \tag{9.15}$$

Mit dem Wissen um die Zeitentwicklung der Eigenvektoren $\vec{e}$ ist es nun möglich, die Zeitentwicklung der Wahrscheinlichkeiten $\vec{p}(t)$ des Systems zu berechnen.

Die Zeitentwicklung der Wahrscheinlichkeiten $\vec{p}(t)$ werden durch t–fache Anwendung der Übergangsmatrix $\boldsymbol{M}$ auf die Wahrscheinlichkeiten $\vec{p}(0)$ zum Zeitpunkt $t = 0$ berechnet:

$$\vec{p}(t) = \boldsymbol{M}^t\vec{p}(0) = \boldsymbol{M}^t\frac{1}{2}(\vec{e}_+ + \vec{e}_-) = \frac{1}{2}\boldsymbol{M}^t\vec{e}_+ + \frac{1}{2}\boldsymbol{M}^t\vec{e}_- \tag{9.16}$$

Unter Einsetzen der Gleichungen (9.10) und (9.15) ergibt sich $\vec{p}(t)$ zu

$$\vec{p}(t) = \frac{1}{2}\vec{e}_+(t) + \frac{1}{2}\vec{e}_-(t) = \frac{1}{2}\vec{e}_+ + \frac{1}{2}\exp(-2rt)\vec{e}_- \tag{9.17}$$

Für $t \to 0$ geht der Exponent der Exponentialfunktion gegen 0, damit verschwindet der Exponentialausdruck und wir erhalten

$$\vec{p}(t \to 0) = \frac{1}{2}(\vec{e}_+ + \vec{e}_-) = p_0 \tag{9.18}$$

Das ist konsistent mit Gl. (9.9). Für $t \to \infty$ geht dagegen der Exponentialausdruck in Gl. (9.17) gegen 0, damit verschwindet der Ausdruck für $\vec{e}_-$ ganz und wir erhalten

$$\vec{p}(t \to \infty) = \frac{1}{2}\vec{e}_+ \tag{9.19}$$

In ausführlicher Schreibweise erhalten wir damit für $\vec{p}(t)$

$$\begin{aligned}\vec{p}(t) &= \frac{1}{2}\begin{pmatrix}1\\1\end{pmatrix} + \frac{1}{2}\exp(-2rt)\begin{pmatrix}1\\-1\end{pmatrix} \\ &= \frac{1}{2}\begin{pmatrix}1+\exp(-2rt)\\1-\exp(-2rt)\end{pmatrix}\end{aligned} \tag{9.20}$$

und damit für die Zeitentwicklung für $t \to \infty$

$$\vec{p}(t \to \infty) = \frac{1}{2}\begin{pmatrix}1\\1\end{pmatrix} = \begin{pmatrix}\frac{1}{2}\\\frac{1}{2}\end{pmatrix} \tag{9.21}$$

Nach langer Zeit gilt also, daß $p_A = p_B = \frac{1}{2}$.

[5] typische Werte für r liegen bei ca. 10^{-9}, $[r] = \mathrm{bp}^{-1} \cdot \mathrm{a}^{-1}$

➤ Die Verteilung p^∞ oder $\vec{p}(\infty)$ heißt stationäre Verteilung und ist durch den Eigenvektor mit dem zugehörigen Eigenwert 1 beschrieben.

Ihre graphische Veranschaulichung ist in Abb. 9.1b (S. 72) wiedergegeben.

Zahl der Substitutionen

Die erwartete Anzahl d^{obs} von beobachtbaren Substitutionen in einer Sequenz der Länge l ergibt sich aus dem Produkt der Sequenzlänge l und der Wahrscheinlichkeit p_{subst} einer Substitution

$$d^{\text{obs}} = l \cdot p_{\text{subst}} \tag{9.22}$$

Da für die gegebene Anfangsbedingung $p_A(0) = 1$ die Wahrscheinlichkeit p_{subst} einer Substitution entsprechend $p_{\text{subst}} = 1 - p_A(0)$ ist, ergibt sich für die erwartete Anzahl beobachteter Substitutionen

$$d^{\text{obs}} = l \cdot (1 - p_A(t)) \tag{9.23}$$

und unter Einsetzen der entsprechenden Teile von Gl. (9.21)

$$\begin{aligned} d^{\text{obs}} &= l \cdot \left(1 - \left(\frac{1}{2} + \frac{1}{2}\exp(-2rt)\right)\right) \\ &= l \cdot \frac{1}{2}(1 - \exp(-2rt)) \end{aligned} \tag{9.24}$$

Die Anzahl der realen Substitutionen d^{real} ist gemäß Gl. (9.1)

$$d^{\text{real}} = r \cdot t \cdot l \qquad\qquad r \cdot t = d^{\text{real}} \cdot l^{-1} \tag{9.25}$$

Daraus folgt unter der Annahme, daß die Zahl beobachteter Substitutionen d^{obs} eine Funktion der Zahl der realen Substitutionen d^{real}, $d^{\text{obs}} = f\left(d^{\text{real}}\right)$ sei, durch Einsetzen von Gl. (9.24) in Gl. (9.25):

$$d^{\text{obs}} = f\left(d^{\text{real}}\right) = l \cdot \frac{1}{2}\left(1 - \exp\left(-2\frac{d^{\text{real}}}{l}\right)\right) \tag{9.26}$$

Durch Umstellen und Umformen

$$\begin{aligned} \frac{2d^{\text{obs}}}{l} &= 1 - \exp\left(-2\frac{d^{\text{real}}}{l}\right) \\ \exp\left(-2\frac{d^{\text{real}}}{l}\right) &= 1 - \frac{2d^{\text{obs}}}{l} \\ -2\frac{d^{\text{real}}}{l} &= \ln\left(1 - \frac{2d^{\text{obs}}}{l}\right) \end{aligned}$$

erhalten wir schließlich

$$\frac{d^{\text{real}}}{l} = -\frac{1}{2}\ln\left(1 - \underbrace{\frac{2d^{\text{obs}}}{l}}_{-\varepsilon}\right) \tag{9.27}$$

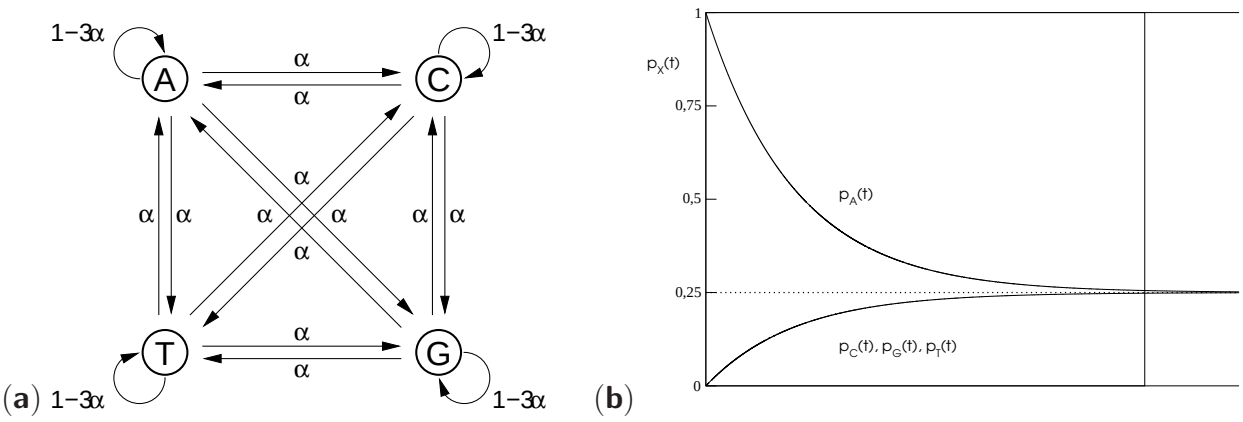

Abb. 9.2: Das Jukes–Cantor Modell der Nukleotid–Substitution in einer DNA–Sequenz. Da das System nur einen Parameter (α) beinhaltet, wird es auch ein–Parameter Modell (*one–parameter model*) genannt. Dieses Modell geht von einer Gleichberechtigung aller Übergänge zwischen den Symbolen aus. Das entspricht einer zufälligen Mutation der Nukleotide. (**a**) Graphische Darstellung der dem JC–Modell zugrundeliegenden Markov–Kette (vgl. auch Abb. 5.1, S. 36); (**b**) die zeitliche Entwicklung des Systems: Die Wahrscheinlichkeit für alle Basen läuft für $t \to \infty$ gegen $p_X(t) = \frac{1}{4}$, d.h. alle Nukleotide sind für lange Zeiten in der Sequenz gleich häufig.

Für die (biologisch gut motivierbare) Annahme, daß die Zahl der Substitutionen in der Sequenz wesentlich geringer als die Sequenzlänge sei, $d^{obs} \ll l$, vereinfacht sich Gl. (9.27) zu:

$$\frac{d^{\text{real}}}{l} = \frac{1}{2}\left(\frac{2d^{\text{obs}}}{l}\right) = \frac{d^{obs}}{l} \tag{9.28}$$

Das aber bedeutet nichts anderes, als daß für die getroffene Annahme (die Zahl der Substitutionen ist wesentlich geringer als die Zahl der Buchstaben in der Sequenz, $d^{obs} \ll l$) das Verhältnis von realen und beobachteten Substitutionen zur Sequenzlänge gleich ist. Bei gleicher Sequenzlänge ergibt sich damit für diesen Fall die Identität von beobachteten Substitutionen d^{obs} und realen Substitutionen d^{real}.

9.2.4 Das Jukes–Cantor–Modell

Das Jukes–Cantor Modell (JC–Modell) der Nukleotid–Substitution in einer DNA–Sequenz wurde von Jukes und Cantor (1969) vorgeschlagen [50]. Es ist das einfachste Modell für die Dynamik der Substitution von Nukleotiden. Da das System nur einen Parameter (α) beinhaltet, wird es in der Literatur auch als ein–Parameter Modell (*one–parameter model*) bezeichnet [55].

Das Modell besteht aus einem Alphabet $\mathcal{A}$ mit vier Buchstaben, $\mathcal{A} = \{A, C, G, T\}$ (den vier Nukleotiden) und einer Substitutionsrate α, der Übergangsrate zwischen den Symbolen. D.h., es geht von einer Gleichberechtigung aller Übergänge zwischen den Symbolen aus, was einer zufälligen Mutation der Nukleotide in der Sequenz entspricht. Eine graphische Darstellung der dem JC–Modell zugrundeliegenden Markov–Kette ist in Abb. 9.2a wiedergegeben (vgl. auch Abb. 5.1, S. 36).

Die Übergangsmatrix (oder Ratenmatrix), die der graphischen Darstellung der Markov–

Kette (Abb. 9.2a) entspricht, lautet:

$$\boldsymbol{M} = \begin{pmatrix} 1-3\alpha & \alpha & \alpha & \alpha \\ \alpha & 1-3\alpha & \alpha & \alpha \\ \alpha & \alpha & 1-3\alpha & \alpha \\ \alpha & \alpha & \alpha & 1-3\alpha \end{pmatrix} \tag{9.29}$$

Da jedes Nukleotid drei Möglichkeiten der Substitution hat, ist die Mutationsrate $r = 3\alpha$.

Zur Berechnung des Zeitverhaltens des Systems (bzw. des Vektors der Wahrscheinlichkeiten $\vec{p}(t)$) wird nach dem gleichen Schema wie im vorherigen Abschnitt bei dem zwei–Buchstaben Markov–Modell verfahren: Die stationäre Verteilung $\vec{p}(\infty)$ ist durch den Eigenvektor mit dem Eigenwert 1 beschrieben.

Der Eigenvektor $\vec{e}_{\max}$ mit maximalem Eigenwert $\lambda_{\max}$ für das JC–Modell ist

$$\vec{e}_{\max} = (1,1,1,1)^T \qquad \text{mit} \qquad \lambda_{\max} = 1 \tag{9.30}$$

Wie es die Definition der Eigenvektoren und Eigenwerte (Gl. 9.7) verlangt, bleibt der Eigenvektor $\vec{e}_{\max}$ bei Multiplikation mit der Matrix $\boldsymbol{M}$ erhalten:

$$\boldsymbol{M}\vec{e}_{\max} = 1 \cdot \vec{e}_{\max} \tag{9.31}$$

Alle anderen Eigenwerte λ_i sind kleiner als 1. Gemäß Gl. (9.9) kann jeder Vektor $\vec{p}$ des Systems in eine Linearkombination der Eigenvektoren zerlegt werden:

$$\vec{p} = c_0\vec{e}_{\max} + \sum_k c_k\vec{e}_k \tag{9.32}$$

Nach langer Zeit ($t \to \infty$) ist der Vektor der Wahrscheinlichkeiten proportional dem Eigenvektor mit dem Eigenwert 1, $\vec{p}(t) \propto (1,1,1,1)^T$. Daraus folgt unmittelbar für die Wahrscheinlichkeiten jedes einzelnen Nukleotids:

$$p_A = p_C = p_G = p_T = \frac{1}{4} \tag{9.33}$$

Die Wahrscheinlichkeit p_{mut}, eine Mutation zu beobachten, ist identisch mit der Wahrscheinlichkeit, nicht dasselbe Nukleotid zu erhalten wie zum Ausgangszeitpunkt. Beispielhaft für p_A (und mit Gl. 9.33 gilt das identisch für die anderen Nukleotide) ergibt sich:

$$1 - p_A(t) = \frac{3}{4} - \frac{3}{4}\exp(-4\alpha t) = p_C = p_G = p_T = p_{\text{mut}} \tag{9.34}$$

Die Mutationsrate kann also direkt aus der Wahrscheinlichkeit $p_{\text{mut}}(0)$, eine Mutation zum Zeitpunkt $t = 0$ zu beobachten, errechnet werden:

$$p_{\text{mut}}(0) = 4\alpha \cdot \frac{3}{4}\underbrace{\exp(-4\alpha t)}_{1}\Bigg|_{0=t} = 3\alpha = r \tag{9.35}$$

Die Zahl beobachteter Substitutionen d^{obs} ist das Produkt aus der Wahrscheinlichkeit $p_{\text{mut}}(t)$ einer Mutation zur Zeit t und der Länge l der betrachteten Sequenz:

$$d^{\text{obs}} = l \cdot p_{\text{mut}}(t) \tag{9.36}$$

Daraus ergibt sich durch Einsetzen von Gl. (9.34) für die längennormierte Zahl beobachteter Substitutionen

$$\frac{d^{\text{obs}}}{l} = \frac{3}{4}(1 - \exp(-4\alpha t)) \tag{9.37}$$

Die "wahre" Anzahl von Substitutionen ist gemäß Gl. (9.1)

$$d^{\text{true}} = 3\alpha \cdot l \cdot t \qquad\qquad \alpha t = \frac{1}{3}\frac{d^{\text{true}}}{l}$$

Durch Einsetzen von Gl. (9.38) in Gl. (9.37) kann die Zahl der beobachteten Substitutionen als Funktion der Zahl der "wahren" Substitutionen ausgedrückt werden:

$$\frac{d^{\text{obs}}}{l} = \frac{3}{4}\left(1 - \exp\left(-\frac{4}{3}\frac{d^{\text{true}}}{l}\right)\right) \tag{9.38}$$

Durch Umstellen ergibt sich daraus für die auf Sequenzlänge normierte Zahl "wahrer" Substitutionen

$$\frac{d^{\text{true}}}{l} = -\frac{3}{4}\ln\left(1 - \frac{4}{3}\frac{d^{\text{obs}}}{l}\right) \tag{9.39}$$

Hierin liegt eine Stärke des Jukes–Cantor–Modells: Es liefert eine korrigierte Formel für die **Bestimmung der Zahl der wahren Substitutionen aus der Zahl beobachteter Substitutionen** für eine Sequenz.

9.2.5 Die Feng–Doolittle–Formel

Wie kommt man von den Scoring–Werten, die aus Sequenzalignments gewonnen wurden, zu Distanzen? Hier hilft die von Feng und Doolittle [27] aufgestellte Formel. Sie berechnen die Distanz D zu

$$D = -\log S_{\text{eff}} = -\log \frac{S_{\text{obs}} - S_{\text{rand}}}{S_{\text{max}} - S_{\text{rand}}} \tag{9.40}$$

mit dem beobachteten Scoringwert S_{obs} eines paarweisen Alignments, dem maximalen Score S_{max} (dem mittleren Scoringwert für das Alignment jeder der beiden Sequenzen mit sich selbst) und dem für das Alignment zweier zufälliger Sequenzen (mit gleicher Menge und gleicher Basenzusammensetzung) erwarteten Score S_{rand}. Der Wert für S_{rand} kann entweder dadurch berechnet werden, daß die beiden Sequenzen jeweils zufällig durchmischt werden, oder durch geeignete Berechnung, wie von Feng und Doolittle [28] angegeben.

Der effektive Scoringwert S_{eff} kann als normalisierter prozentualer Wert der Ähnlichkeit der beiden alignten Sequenzen betrachtet werden. Der Logarithmus in der obigen Formel rührt daher, daß der Wert für S_{eff} mit steigender evolutionärer Distanz ungefähr exponentiell abfällt. Durch die berechnung des Logarithmus verhält sich der Wert linearer zur evolutionären Distanz.

Diese Formel reicht für eine erste grobe Abschätzung der Distanz. Für die Konstruktion molekularer Stammbäume muß mehr Sorgfalt auf die Berechnung gelegt werden. [21]

Die Formel (9.40) ist konsistent mit dem Jukes–Cantor–Modell, wenn man für die Scoring Matrix $\boldsymbol{S}$ die Einheitsmatrix $\boldsymbol{E}$

$$\boldsymbol{S}(X, X') = \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix} = \boldsymbol{E}$$

wählt. Mit den Gleichsetzungen für die Werte für den beobachteten Scoringwert S_{obs}, den zufälligen Scoringwert S_{rand} und den maximalen Scoringwert S_{rand} mit entsprechenden Parametern des Jukes–Cantor–Modells

$$S_{\text{obs}} \to l - d^{\text{obs}} \qquad S_{\text{rand}} \to \frac{1}{4} \cdot l \qquad S_{\text{max}} \to l$$

gilt für die Distanz D

$$D = \ln \frac{l - d^{\text{obs}} - \frac{1}{4}l}{l - \frac{1}{4}l} = \ln \frac{\frac{3}{4}l - d^{\text{obs}}}{\frac{3}{4}l}$$

und nach Umformen

$$D = \ln \left(1 - \frac{4}{3}\frac{d^{\text{obs}}}{l}\right) \tag{9.41}$$

Diese Formel stimmt mit der für das Jukes–Cantor–Modell errechneten Gl. (9.39) bis auf einen Vorfaktor überein.

9.3 Verfahren zur Konstruktion von Stammbäumen

9.3.1 Überblick

Für eine Übersicht vgl. Abb. 9.3

- UPGMA (average linkage)
  - unweighted pair group matching using arithmetic averages
  - Sokal und Michener [81]
- NJ
  - Neighbour Joining
  - Saitou und Nei [73]
- MP
  - Maximum Parsimony
  - Fitch [33]

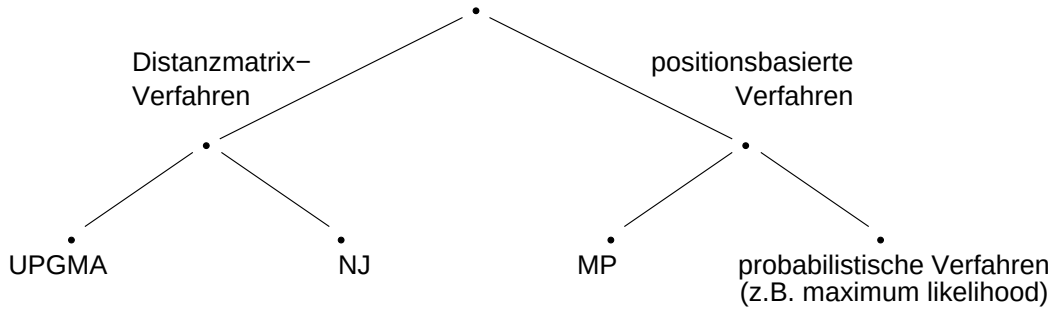

Abb. 9.3: Überblick über verschiedene Verfahren der Bioinformatik zur Konstruktion phylogenetischer Bäume (Stammbäume). Im Text näher behandelt werden ausschließlich die Distanzmatrix–Verfahren, namentlich der UPGMA–Algorithmus.

> A long–standing controversy in taxonomy has been the often acrimonious dispute between cladists and pheneticists. The term cladistics can be defined as the study of the pathway of evolution. In other words, cladists are interested in such questions as: how many branches are there among a group of organisms; which branch connects to which other branch; and what is the branching sequence (Sneath and Sokal 1973). A treelike network that expresses such ancestor–descendant relationships is called a cladogram. To put it another way, a cladogram refers to the topology of a rooted phylogenetic tree.
>
> On the other hand, phenetics is the study of relationships among a group of organisms on the basis of the degree of similarity between them, be that similarity molecular, phenotypic, or anatomical. A treelike network expressing phenetic relationships is called a phenogram. While a phenogram may serve as an indicator of cladistic relationships, it is not necessarily identical with the cladogram. If there is a linear relationship between the time of divergence and the degree of genetic (or morphological) divergence, the two types of trees may become identical to each other.
>
> [55, p. 126]

9.3.2 UPGMA–Algorithmus

Die einfachste Methode, Stammbäume zu rekonstruieren, ist UPGMA, kurz für *unweighted pair–group method with arithmetic mean*. Sie wurde von Sokal und Michener [81] ursprünglich für die Konstruktion taxonomischer Phenogramme (Bäumen, die die phänotypischen Ähnlichkeiten zwischen OTUs wiedergeben) entwickelt. Sie kann aber ebenso gut für die Konstruktion von Stammbäumen (*phylogenetic trees*) verwandt werden. [55]

Die Voraussetzung ist, daß die Mutationsraten über die Zeit ungefähr konstant sind, so daß ein ungefährer linearer Zusammenhang zwischen dem evolutionären Abstand und der vergangenen Zeit besteht [64]. Das entspricht den Annahmen der Molekularen Uhr–Hypothese. Um diese Bedingung zu erfüllen, sollten lineare Abstandsverfahren wie z.B. die Hamming–Distanz oder die Feng–Doolittle–Formel verwendet werden.

Ausgangspunkt des UPGMA–Algorithmus ist ein Satz von Sequenzen $(S_1 \dots S_n)$. Der eigentliche Algorithmus besteht aus zwei (drei) Schritten: (1) Initialisierung, (2) Iterative Gruppierung (clustering) und (3) Termination, wobei sich der dritte vom zweiten Schritt

nur dadurch unterscheidet, daß nur noch zwei Cluster vorhanden sind, die es zusammenzufassen gilt.

Initialisierung

- Ordne jeder Sequenz $i = 1 \dots n$ eine eigene Gruppe (*cluster*) $C_i = C_1 \dots C_n$ zu.
- Konstruiere die Distanzmatrix d_{ij} zu den Clustern $C_1 \dots C_n$.

  Z.B. durch paarweises Alignment und Benutzung der Feng–Doolittle–Formel oder der Hamming–Distanz.

Iteration

Solange es mehr als zwei Gruppen gibt:

- Bestimme die beiden Gruppen (*cluster*) C_p, C_q, deren Abstand d_{pq} minimal ist. (Gibt es mehrere Gruppen mit gleichem Abstand, dann wähle durch Zufall eine aus.)
- Fasse die beiden Gruppen C_p, C_q zu einem neuen Cluster $C_r = C_p \cup C_q$ zusammen.
- Berechne die reduzierte Distanzmatrix ($d = d^{\min}$).

  Verwende dazu die Abkürzung

  $$d_{rs} = \frac{n_p \cdot d_{ps} + n_q \cdot d_{qs}}{n_p + n_q}$$

  mit der Zahl n_i der Sequenzen im Cluster C_i.
- Definiere als Abstandsmaß $d(C_p, C_q)$ zwischen zwei Gruppen von Sequenzen (*clustern*)

  $$d(C_p, C_q) = \frac{1}{n_p n_q} \sum_{\substack{i \in C_p \\ j \in C_q}} d_{ij}$$

  Das entspricht dem mittleren Abstand zwischen den Paaren der Sequenzen jeder Gruppe mit der Zahl n_i der Sequenzen im Cluster C_i.
- Definiere einen Knoten r mit den Tochterknoten p und q und plaziere ihn auf der Höhe $d_{pq}/2$.
- Füge C_r zu den Clustern hinzu und entferne C_p und C_q.

Termination

Wenn es nur noch zwei Gruppen (*cluster*) C_i, C_j gibt:

- Fasse beide Gruppen zusammen und plaziere die Wurzel des Baumes auf der Höhe $d_{ij}/2$.

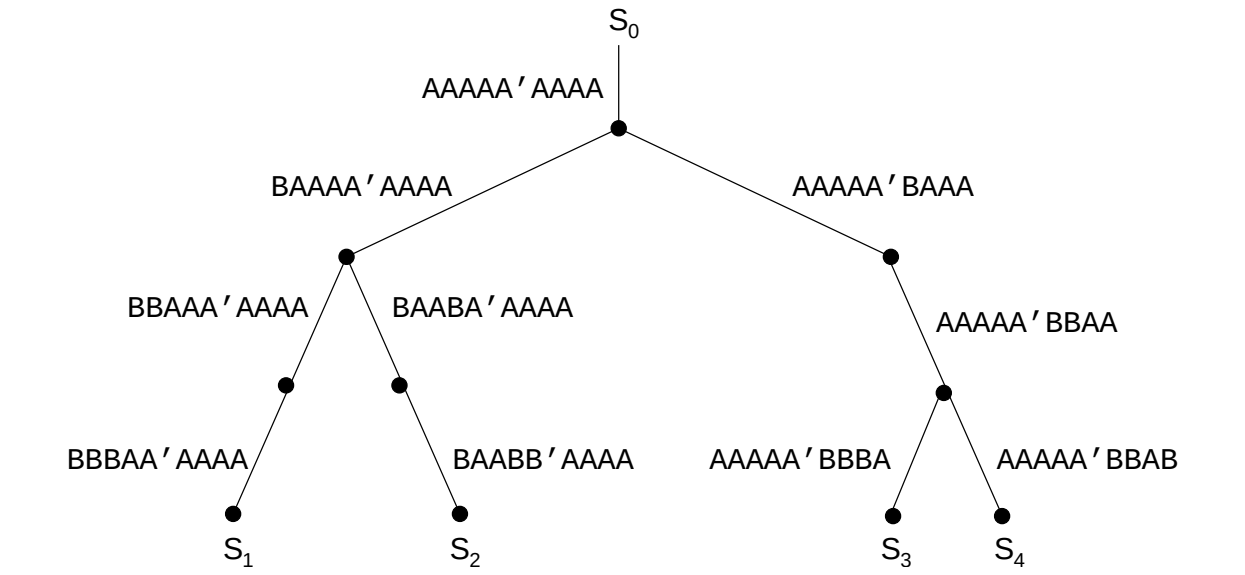

Abb. 9.4: Ein Beispiel für divergente Evolution. Die Endpunkte des Stammbaumes (*external, terminal nodes*) werden als *operational taxonomic unit* (OTU) bezeichnet [55]. Sie sind die Sequenzen, die verglichen werden.

■ **Ein Stammbaum von vier Sequenzen mittels UPGMA** Gegeben seien vier Sequenzen $S_1 \ldots S_4$ (S_1 = `BBBAA AAAA`, S_2 = `BAABB AAAA`, S_3 = `AAAAA BBBA` und S_4 = `AAAAA BBAB`), für die ein Stammbaum mit UPGMA erstellt werden soll. Die realen Abstammungsverhältnisse seien in Abb. 9.4 wiedergegeben.

Zunächst wird jeder Sequenz $S_{1\ldots4}$ ein Cluster $C_{1\ldots4}$ zugeordnet. Anschließend wird die Distanzmatrix d für alle vier Cluster unter der Verwendung der Hamming–Distanz zur Distanzbestimmung berechnet:

$$d(C_i, C_j) = \begin{pmatrix} 0 & 4 & 6 & 6 \\ 4 & 0 & 6 & 6 \\ 6 & 6 & 0 & 2 \\ 6 & 6 & 2 & 0 \end{pmatrix} = d^0$$

Aus ihr geht hervor, daß die Cluster C_3 und C_4 den geringsten Abstand zueinander aufweisen. Sie werden zu einem neuen Cluster C_5 zusammengefaßt. Die reduzierte Distanzmatrix d_{5k} mit $k = 1, 2$ ergibt sich damit zu:

$$d_{5k} = \begin{pmatrix} 0 & 4 & 6 \\ 4 & 0 & 6 \\ 6 & 6 & 2 \end{pmatrix} = d^1$$

Jetzt sind es die beiden Cluster C_1 und C_2, die den geringsten Abstand zueinander aufweisen. Sie werden ebenfalls zu einem neuen Cluster, C_6, zusammengefaßt und die reduzierte Distanzmatrix d_{65} berechnet:

$$d_{65} = \begin{pmatrix} 4 & 6 \\ 6 & 2 \end{pmatrix} = d^2$$

Jetzt ist der Fall eingetreten, daß es nur noch zwei Cluster, C_5 und C_6, gibt. Ihr Abstand zueinander ergibt sich zu $d^3 = 6$. In diesem Abstand wird die Wurzel des Stammbaumes

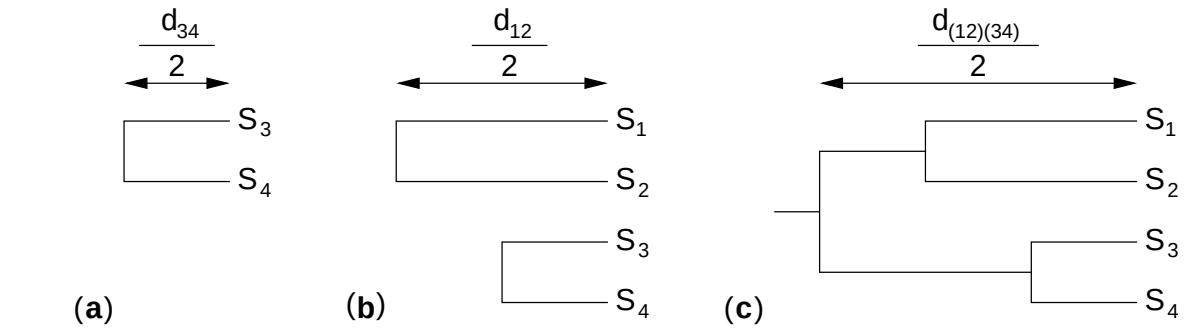

Abb. 9.5: UPGMA–Stammbaum der Sequenzen aus Abb. 9.4. Die Distanzen d wurden über die Hamming–Distanzen der Sequenzen zueinander berechnet. Vergleicht man die Teilabbildung (c) mit dem "realen" Stammbaum in Abb. 9.4, stellt man fest, daß der Algorithmus aus den Sequenzen die richtige Abstammung berechnet hat.

eingezeichnet und der Algorithmus beendet (terminiert). Die einzelnen Schritte und das Ergebnis des UPGMA–Algorithmus für dieses Beispiel sind in Abb. 9.5 wiedergegeben.

▲ **Ultrametrizität** Für jedes Tripel von OTUs[6] (Sequenzen) gilt: Zwei der drei paarweisen Distanzen sind gleich und die dritte ist kleiner, etwa

$$d_{32} = d_{42} > d_{34}$$

Sie ist eine Voraussetzung für die erfolgreiche Verwendung des UPGMA

Ein heuristisches Verfahren für die Erstellung sowohl von Alignment als auch der dazugehörigen Stammbäume ist das CLUSTAL Programm [45]. Charakteristisch ist hier die Variation der gap opening penalty (GOP). Trotz daß es sich um ein heuristisches Verfahren handelt, kommen seine Ergebnisse der Realität nahe.

9.3.3 Phylogenetic Profiling

Beim *phylogenetic profiling* werden Gene aufgrund ihres Verteilungsmusters bei verschiedenen Arten gruppiert. Diese Genverteilungs–Muster wurden als nützliches Werkzeug für das Studium der Evolution als auch als Ansatz in der Erforschung der Funktion von Proteinen vorgeschlagen [z.B. 90, 22, 23, 24]. Erstmals im Detail beschrieben von Gaasterland und Ragan [35, 36, 69] wurde das Verfahren später unter dem Namen *phylogenetic profiling* [68] bekannt. [25]

> Since genes that function together in the same cellular process are frequently inherited or lost as a unit, their distribution patterns across species are often similar. Thus, the function of a query gene can possibly be predicted if its presence/absence pattern across species is the same as genes with known functions. In the initial phylogenetic profiling study (Pellegrini et al., 1999), the profile for a gene was binary in nature. A gene was considered present in another genome if there was a match better than some threshold using a similarity search tool such as BLAST.

[6] OTU = operational taxonomic unit

[25]

Es liegt ein sequenzierter Organismus vor, ein sogenannter Modellorganismus, O_0. Die Summe G all seiner Gene g_n bzw. ORFs orf_n läßt sich mathematisch schreiben als $G \in g_1 \dots g_n$ bzw. $G \in \text{orf}_1 \dots \text{orf}_n$.

Das Ziel des *phylogenetic profiling* ist ein zweifaches: sowohl die Erstellung eines Stammbaumes der Organismen als auch die Bestimmung von Gen–/Proteinfunktionen für noch nicht funktional charakterisierte Gene/Proteine. Je nach Ausgangslage können dabei zwei Ansätze unterschieden werden: (i) Liegen weitere zu vergleichende Organismen ($O_1 \dots O_m$) sequenziert vor, kommen bioinformatische Methoden (*alignment approach*) zum Einsatz. (ii) Liegt dagegen nur die DNA weiterer Organismen vor, werden Array–Experimente (*array experiments*) durchgeführt (*array approach*).

Alignment Approach

Gegeben sei ein Organellorganismus O_0 mit seinen ORFs $\text{orf}_1 \dots \text{orf}_n$. Liegt der zweite Organismus O_1 sequenziert vor, dann kann ein lokales Alignment gegen das gesamte Genom G_1 von O_1 durchgeführt werden. Das Resultat wird in einer Matrix $\boldsymbol{M}$ zusammengefaßt.

Das Ziel ist jetzt, eine sogenannte Gen Indikator Matrix (*gene indicator matrix*) $\boldsymbol{H}$ zu erzeugen, die binäre Aussagen über das Vorhandensein eines ORFs aus dem Modellorganismus O_0 im Organismus O_1 enthält. Dazu wählt man einen geeigneten Schwellenwert ϑ und transformiert die Matrix $\boldsymbol{M}$, so daß gilt:

$$\boldsymbol{M} \rightarrow \boldsymbol{M}' = \boldsymbol{H}[\boldsymbol{M} - \vartheta] = \begin{cases} 1 & x > \vartheta \\ 0 & x < \vartheta \end{cases}$$

Diese Matrix $\boldsymbol{H}$ enthält für jedes "erfolgreiche" Alignment eines ORFs des Modellorganismus O_0 mit einem DNA–Abschnitt des Organismus O_1 eine 1, für alle anderen Alignments eine 0.

Da dieser Ansatz die DNA–Sequenzen der Organismen aus Sequenzbibliotheken bezieht, wird er auch als *library screening* bezeichnet.

Array Approach

Ist kein sequenziertes Genom eines zweiten Organismus O_1 vorhanden, dann muß auf den experimentellen Ansatz, die sogenannten *DNA microarrays* ausgewichen werden.

Die ORFs des Modellorganismus O_0 werden amplifiziert (PCR) und als Array auf einen Glas– oder Nylonträger gespottet. Aus den Testorganismen $O_1 \dots O_m$ wird die genomische DNA extrahiert und anschließend fragmentiert. Die Fragmente genomischer DNA jedes Organismus O_i werden mit einem spezifischen Fluoreszenzmarker gelabelt und anschließend m Hybridisierungen mit m Array ausgeführt. Die Arrays werden eingescannt und die Fluoreszenzinformationen in Falschfarben übersetzt. Anschließend können die so erzeugten Bilder (*images*) ausgewertet werden.

Die Auswertung der *images* erfolgt analog dem Alignment Approach durch Einführung eines Schwellwertes ϑ. Das Ergebnis ist auch hier eine Gen Indikator Matrix $\boldsymbol{H}$, die für jedes

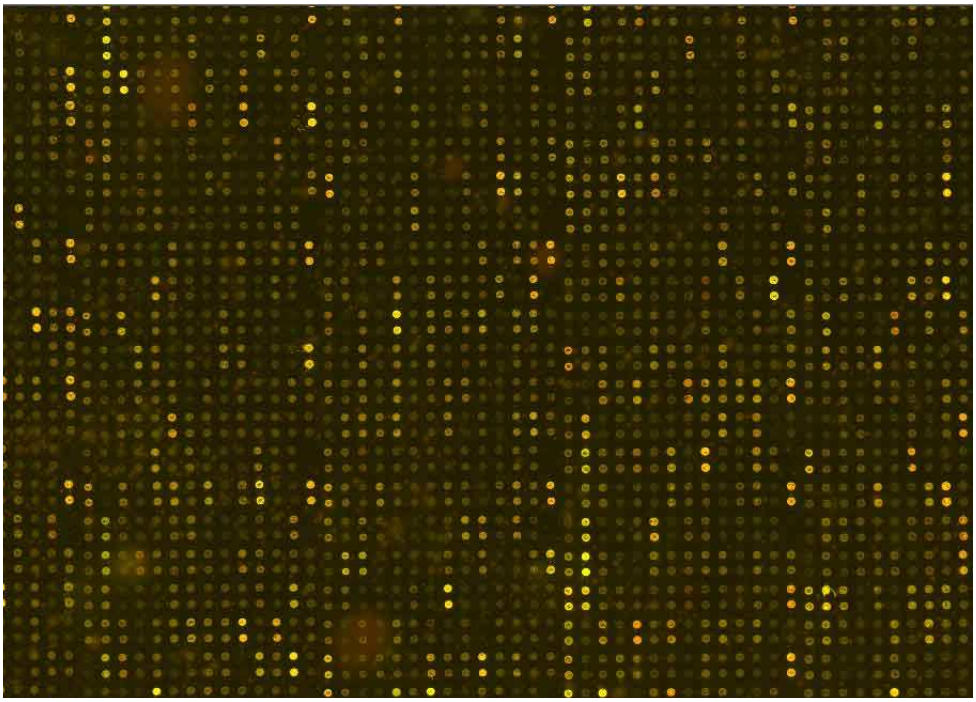

Abb. 9.6: Eingescanntes Bild des Arrays. Intensität und Farbe jedes Spots enthalten Information über das jeweilige Gen der getesteten Probe. Quelle: `http://www.microdiscovery.com/`

"erfolgreiche" Alignment eines ORFs des Modellorganismus O_0 mit einem DNA–Abschnitt des Organismus O_1 eine 1, für alle anderen Alignments eine 0 enthält.

Eine Erweiterung des DNA Microarrays ist die **kompetitive Hybridisierung**, d.h. es wird die genomische DNA zweier Modellorganismen gleichzeitig verwendet und mit unterschiedlichen Fluoreszenzmarkern (z.B. rot und grün) versehen. Kommt ein Gen in beiden Organismen vor, überlagert sich die Farbe der Fluoreszenz (hier zu gelb).

Mit beiden Ansätzen können zwei Fragestellungen bearbeitet werden: Die Frage nach der Verwandtschaft der Organismen und nach der funktionalen Gruppierung von Genen.

Verwandtschaft von Organismen

Die Daten der Gen Indikator Matrix $\boldsymbol{H}$ können zur Konstruktion eines Stammbaumes, z. B. mittels UPGMA, verwendet werden. Dazu wird zunächst die Distanzmatrix d_{ij} zwischen zwei Organismen O_i und O_j aufgestellt:

$$d_{ij} = (\vec{O}_i - \vec{O}_j)^2$$

Zur Anwendung kommt auch hier wieder die Hamming–Distanz. Der Vektor $\vec{O}_i$ ist der Spaltenvektor, der den Organismus O_i charakterisiert. Aus der Distanzmatrix läßt sich dann mittels UPGMA–Algorithmus ein Stammbaum der Organismen erstellen.

Funktionale Gruppierung der Gene

Ein weiterer interessanter Ansatz ist die Erstellung einer funktionalen Gruppierung der Gene. Diesem Ansatz liegt die Annahme zugrunde, daß Gene, die funktional zusammengehören, z.B. weil sie dem gleichen Stoffwechselweg angehören, auch immer nur gemeinsam in Organismen auftreten.

Auch hier kann eine Distanzmatrix d, diesmal für die Gene (ORFs) g_α und g_β, erstellt werden:

$$d_{\alpha,\beta} = (g_\alpha - g_\beta)^2 \qquad \Rightarrow \qquad n \times n\text{–Matrix}$$

Dabei ist g_α^T der Zeilenvektor, der das Gen α charakterisiert. Mittels UPGMA können nun Gruppen zusammengehöriger Gene bestimmt werden.

Abbildungsverzeichnis

Tabellenverzeichnis

Anhang A

Vokabelverzeichnis engl.–dt.

accuracy	Genauigkeit
address	behandeln
ahead of sth.	einer Sache voraus sein
alignment	Ausrichtung, Alignment
amenable	offen, zugänglich
approach	Ansatz
appropriate	angemessen, angebracht; zugehörig
arbitrary	beliebig
assessment	Bewertung, Abschätzung
be associated with	im Zusammenhang stehen mit
be capable	fähig sein
challenge	anzweifeln, in Frage stellen
clustering	
comparison	Vergleich
computational biology	
concern	betreffen
consensus sequence	Konsensussequenz
constrain	einschränken, behindern
constraint	Bedingung, Einschränkung; Nebenbedingung; Zwang
context–free grammar	kontextfreie Grammatik
contribute	beitragen
correspond to	entsprechen, sich beziehen auf
crucial	ausschlaggebend, entscheidend
denote	bezeichnen, kennzeichnen
derive	ableiten, herleiten
die, *pl.* dice	Würfel
discrimination	Unterscheidung
drawback	Nachteil
dynamic programming	dynamische Programminerung
elucidate	erläutern
erratic	schwankend, sprunghaft
estimate [*Verb*]	schätzen, abschätzen

estimate [*Subst.*]	Schätzer
examine	behandeln
false positive	Falschpositives
feasible	machbar, möglich
gap	Lücke, Spalte
genefinding	Genidentifikation
homology	Homologie
impact	Auswirkung
inference	Rückschluß, Folgerung, Deduktion
inquiry	Forschung, Nachforschung
intimately interrelated	in enger Wechselbeziehung miteinander stehend
likelihood	Wahrscheinlichkeit
lineage	Abstammung, Abstammungslinie
locate	auffinden
maximum likelihood	größte Wahrscheinlichkeit
mismatch	Fehlanpassung
obtain	erhalten, erreichen, erlangen
odds *Pl.*	Ungleichheit
pairwise	paarweise
penalty	Strafe, Bestrafung
phylogenetic tree	Phylogenetischer Stammbaum
policy	Strategie
posterior	später
prediction	Vorhersage
preliminary	einleitend, vorausgehend, vorläufig
preprocessing	Vorverarbeitung
probabilistic	wahrscheinlichkeitstheoretisch
probabilistic model	Wahrscheinlichkeitsmodell
probability	Wahrscheinlichkeit
probability distribution	Wahrscheinlichkeitsverteilung
profile	Profil, Querschnitt
ratio	Anteil, Quotient
realm	Bereich, Fachgebiet
reasonable	angemessen, vernünftig
with regard to	im Hinblick auf
reliable	verläßlich, zuverlässig
residual	Residuum
residue	Rest [auch chem.], Überrest
reveal	aufdecken, aufzeigen
score	Punktzahl, Score
scoring scheme	

similarity	Ähnlichkeit
sophisticated	anspruchsvoll
speech recognition	Spracherkennung
spread	Streuung
spurious	falsch
state	Zustand
state machine	Automat
statistical significance	statistische Bedeutung/Signifikanz
stretch	Strecke, Ausdehnung
swep [swap, swapped]	tauschen
tackle	angehen
in terms of	in Form von
threshold	Schwelle, Grenzwert
trace back	zurückführen auf
tractable	lenkbar
transition probabilities	Übergangswahrscheinlichkeiten
treat	behandeln
unambiguously	unzweideutig, eindeutig
underpin	unterbauen, untermauern
variety	Vielzahl
weight	Gewicht
weight matrix	Gewichtsmatrix

Literaturverzeichnis

[1] **Baldi P, Brunak S** (2001) Bioinformatics, Adaptive Computation and Machine Learning, MIT Press, Massachusetts, 2. Aufl.

[2] **Baltimore D** (1970) RNA-dependent DNA polymerase in virions of RNA tumour viruses. Nature **226**(252):1209–11

[3] **Bellman RE** (1957) Dynamic Programming, Princeton University Press, Princeton, NJ

[4] **Berg JM** (1990) Zinc Finger Domains: Hypotheses and Current Knowledge. Annual Review of Biophysics and Biophysical Chemistry **19**:405–421

[5] **Berg O, von Hippel P** (1987) Selection of DNA binding sites by regulatory proteins. Statistical-mechanical theory and application to operators and promoters. J Mol Biol **193**(4):723–50

[6] **Bernardi G** (1995) The human genome: Organization and evolutionary history. Annu Rev Genet **29**:445–476

[7] **Bernardi G** (2000) Isochores and the evolutionary genomics of vertebrates. Gene **241**(1):3–17

[8] **Bernardi G, Mouchiroud D, Gautier C, Bernardi G** (1988) Compositional patterns in vertebrate genomes: Conservation and change in evolution. J Mol Evol **28**:7–18

[9] **Bernardi G, Olofsson B, Filipski J, Zerial M, Salinas J, Cuny G, Meunier-Rotival M, Rodier F** (1985) The mosaic genome of warm–blooded vertebrates. Science **228**:953–958

[10] **Bertsekas D** (1995) Dynamic Programming and Optimal Control, Athena Scientific, Belmont, MA

[11] **Bird A** (1987) CpG islands as gene markers in the vertebrate nucleus. Trends in Genetics **3**:342–347

[12] **Borodovsky M, McIninch J** (1993) Genmark: Parallel gene recognition for both DNA strands. Computers Chem **17**:123–133

[13] **Borodovsky M, McIninch JD, Koonin EV, Rudd KE, Medigue C, Danchin A** (1995) Detection of new genes in a bacterial genome using Markov models for three gene classes. Nucl Acids Res **23**:3554–3562

[14] **Borodovsky M, Rudd KE, Koonin EV** (1994) Intrinsic and extrinsic approaches for detecting geges in a bacterial genome. Nucl Acids Res **22**:4756–4767

[15] **Bronstein IN, Semendjaev KA, Musiol G, Mühlig H** (1999) Taschenbuch der Mathematik, Harri Deutsch, Frankfurt/M.

[16] **Clarke B** (1970) Selective constraints on amino acid substitutions during the evolution of proteins. Nature **228**:159–160

[17] **Crick FHC** (1958) On Protein Synthesis, in The Biological Replication of Macromolecules, Bd. XII von *Symp. Soc. Exp. Biol.*, S. 138–163

[18] **Dayhoff MO, Schwartz RM, Orcutt BC** (1978) A model of evolutionary change in proteins, in MO Dayhoff, (Hg.) Atlas of Protein Science and Structure, Bd. 5, S. 345–352, National Biomedical Research Foundation, Washington D.C., supplement 3

[19] **Dessy R** (2001) What is Bioinformatics? Virginia Tech Research Magazine

[20] **Douglas J, Trulove S** (2001) Bioinformatics. Virginia Tech Research Magazine

[21] **Durbin R, Eddy SR, Krogh A, Mitchison G** (1998) Biological sequence analysis, Cambridge University Press, Cambridge

[22] **Eisen JA** (1998) A phylogenomic study of the MutS family of proteins. Nucleic Acids Res **26**:4291–4300

[23] **Eisen JA** (1998) Phylogenomics: Improving functional predictions for uncharacterized genes by evolutionary analysis. Genome Res **8**:163–167.

[24] **Eisen JA** (1999) Evolution of DNA Repair Genes, Proteins, and Processes, Stanford University, Stanford, CA

[25] **Eisen JA, Wu M** (2002) Phylogenetic Analysis and Gene Functional Predictions: Phylogenomics in Action. Theoretical Population Biology **61**:481–487

[26] **Feistel R, Ebeling W** (1982) Models of darwinian processes and evolutionary principles. Biosystems **15**(4):291–9

[27] **Feng DF, Doolittle RF** (1987) Progressive sequence alignment as a prerequisite to correct phylogenetic trees. Journal of Molecular Evolution **25**:351–360

[28] **Feng DF, Doolittle RF** (1996) Progressive alignment of amino acid sequences and construction of phylogenetic trees from them. Methods in Enzymology **266**:368–382

[29] **Fickett JW** (1982) Recognition of protein coding regions in DNA sequences. Nucleic Acids Res **10**:5305–5318

[30] **Fickett JW, Hatzigeorgiou AG** (1997) Eukaryotic Promoter Recognition. Genome Res **7**(9):861–878

[31] **Fickett JW, Tung CS** (1992) Assessment of protein coding measures. Nucleic Acid Res **20**:6441–6450

[32] **Filipski J** (1987) Correlation between molecular clock ticking, codon usage, fidelity of DNA repair, chromosome banding and chromatin compactness in germline cells. FEBS Letters **217**:184–186

[33] **Fitch WM** (1971) Toward defining the course of evolution: minimum change for a specified tree topology. Systematic Zoology **20**:406–416

[34] **Frishman D, Mironov A, Mewes HW, Gelfand M** (1998) Combining diverse evidence for gene recognition in completely sequenced bacterial genomes. Nucleic Acids Research **26**(12):2941–2947

[35] **Gaasterland T, Ragan MA** (1998) Constructing multigenome views of whole microbial genomes. Microb Comp Genomics **3**:177–192

[36] **Gaasterland T, Ragan MA** (1998) Microbial genescapes: Phyletic and functional patterns of ORF distribution among prokaryotes. Microb Comp Genomics **3**:199–217

[37] **Gotoh O** (1982) An improved algorithm for matching biological sequences. Journal of Molecular Biology **162**:705–708

[38] **Grosse I, Buldyrev SV, Stanley HE, Holste D, Herzel H** (2000) Average mutual information ofcoding and non-coding DNA. Pacific Symp Biocomput **5**:611–620

[39] **Grosse I, Herzel H, Buldyrev SV, Stanley HE** (2000) Species independence of mutual information in coding and non-coding DNA. Phys Rev E **61**:5624–5629

[40] **Haldane JBS** (1932) The Causes of Evolution, Longmans and Green, London

[41] **Hebsgaard S, Korning P, Tolstrup N, Engelbrecht J, Rouze P, Brunak S** (1996) Splice site prediction in Arabidopsis thaliana pre-mRNA by combining local and global sequence information. Nucl Acids Res **24**(17):3439–3452

[42] **Henikoff S, Henikoff JG** (1992) Amino acid substitution matrices from protein blocks. Proceedings of the National Academy of Sciences of the USA **89**:10915–10919

[43] **Hennig W** (1998) Genetik, Springer, Berlin Heidelberg, 2. Aufl.

[44] **Herzel H, Weiss O, Trifonov E** (1999) 10-11 bp periodicities in complete genomes reflect protein structure and DNA folding. Bioinformatics **15**(3):187–193

[45] **Higgins D, Thompson J, Gibson T** (1996) Using CLUSTAL for multiple sequence alignments, in R Doolittle, J Abelson, M Simon, (Hg.) Computer Methods for Macromolecular Sequence Analysis, Bd. 266 von *Methods in Enzymology*, S. 383–402, Academic Press

[46] **Holmquist GP, , Filipski J** (1994) Organization of mutations along the genome: A prime determinant of genome evolution. Trends Ecol Evol **9**:65–69

[47] **Holste D, Grosse I, Buldyrev SV, Stanley HE, Herzel H** (2000) Optimization of coding potentials using positional dependence of nucleotide frequencies. J theor Biol **206**:525–537

[48] **Holste D, Weiss O, Grosse I, Herzel H** (2000) Are noncoding sequences of Rickettsia prowazekii remnants of "neutralized" genes? J Mol Evol **51**(4):353–62

[49] **Jacob F** (1977) Evolution and tinkering. Science **196**:1161–1166

[50] **Jukes TH, Cantor C** (1969) Evolution of protein molecules, in Mammalian Protein Metabolism, S. 21–132, Academic Press, New York

[51] **Kimura M** (1968) Evolutionary rate at the molecular level. Nature **217**:624–626

[52] **Kimura M** (1983) The Neutral Theory of Molecular Evolution, Cambridge University Press

[53] **King JL, Jukes TH** (1969) Non–Darwinian evolution. Science **164**:788–798

[54] **Landschulz W, Johnson P, McKnight S** (1988) The leucine zipper: a hypothetical structure common to a new class of DNA binding proteins. Science **240**(4860):1759–64

[55] **Li WH** (1997) Molecular Evolution, Sinauer Associates, Sunderland

[56] **Lottspeich F, Zorbas H**, (Hg.) (1998) BioAnalytik, Spektrum Akademischer Verlag

[57] **Luger K, Mäder AW, Richmond RK, Sargent DF, Richmond TJ** (1997) Crystal structure of the nucleosome core particle at 2.8 Å resolution. Nature **389**:251–60

[58] **Lukashin A, Borodovsky M** (1998) GeneMark.hmm: new solutions for gene finding. Nucl Acids Res **26**(4):1107–1115

[59] **Madigan MT, Martinko JM, Parker J** (1997) Brock Biology of Microorganisms, Prentice Hall, New Jersey, eighth edition Aufl.

[60] **Michel CJ** (1986) New statistical approach to discriminate protein coding from noncoding DNA regions in DNA sequences and its evaluation. J theor Biol **120**:223–236

[61] **Muller JJ** (1935) The origination of chromatin deficiencies as minute deletions subject to insertion elsewhere. Genetics **17**:237–252

[62] **Mulligan M, Hawley D, Entriken R, McClure W** (1984) Escherichia coli promoter sequences predict in vitro RNA polymerase selectivity. Nucleic Acids Res **12**:789–800

[63] **Needleman SB, Wunsch CD** (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology **48**:443–453

[64] **Nei M** (1975) Molecular Population Genetics and Evolution, North–Holland, Amsterdam

[65] **Ohno S** (1970) Evolution by Gene Duplication, Springer–Verlag, Berlin

[66] **Pavletich N, Pabo C** (1991) Zinc finger-DNA recognition: crystal structure of a Zif268-DNA complex at 2.1 Å. Science **252**(5007):809–17

[67] **Pedersen AG, Nielsen H** (1997) Neural network prediction of translation initiation sites in eukaryotes: Perspectives for EST and genome analysis., in Proceedings of the Fifth Interational Conference on Intelligent Systems for Molecular Biology, S. 226–233, AAAI Press, Menlo Park, CA

[68] **Pellegrini M, Marcotte EM, Thompson MJ, Eisenberg D, Yeates TO** (1999) Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles. PNAS **96**(8):4285–4288

[69] **Ragan MA, Gaasterland T** (1998) Microbial genescapes: A prokaryotic view of the yeast genome. Microb Comp Genomics **3**:219–235

[70] **Rhodes D** (1997) Chromatin structure: The nucleosome core all wrapped up. Nature **389**:231–232

[71] **Rhodes D, Klug A** (1986) An underlying repeat in some transcriptional control sequences corresponding to half a double helical turn of DNA. Cell **46**(1):123–32

[72] **Richmond R C** (1970) Non–Darwinian Evolution: A critique. Nature **225**:1025–1028

[73] **Saitou N, Nei M** (1987) The neibor–joining method: a new method for reconstructing phylogenetic trees. Molecular Biology and Evolution **4**:406–425

[74] **Schneider TD, Stephens RM** (1990) Sequence Logos: A New Way to Display Consensus Sequences. Nucleic Acids Res **18**:6097–6100

[75] **Schneider TD, Stormo GD, Gold L, Ehrenfeucht A** (1986) Information content of binding sites on nucleotide sequences. J Mol Biol **188**:415–431

[76] **Shannon C** (1948) A mathematical theory of communication. Bell System Tech J **27**:379–423

[77] **Shannon C** (1951) Prediction and entropy of printed English. Bell Systems Technical Journal **30**:50–64

[78] **Shine J, Dalgarno L** (1974) The 3'–terminal sequence of *E. coli* 16S rRNA: Complementary to nonsense triplets and ribosome binding sites. Proc Natl Acad Sci USA **71**:1342–1346

[79] **Silverman BD, Linsker R** (1986) A measure of DNA periodicity. J theor Biol **118**:295–300

[80] **Smith TF, Waterman MS** (1981) Identification of common molecular subsequences. Journal of Molecular Biology **147**:195–197

[81] **Sokal RR, Michener CD** (1958) A statistical method for evaluating systematic relationships. University of Kansas Scientific Bulletin **28**:1409–1438

[82] **Staden R** (1984) Measurements of the effects that coding for a protein has on a DNA sequence and their use for finding genes. Nucleic Acids Res **12**:551–556

[83] **Staden R** (1988) Methods to define and locate patterns of motivs in sequences. Comuter Applications in the Biosciences **4**:53–60

[84] **Stein L** (2001) Genome annotation: from sequence to biology. Nature Reviews Genetics **2**:493–503

[85] **Steipe B** (1998) Sequenzdatenanalyse, Kap. 21, in [56]

[86] **Stephens RM, Schneider TD** (1992) Features of spliceosome evolution and function inferred from an analysis of the information at human splice sites. J Mol Biol **228**:1124–1136

[87] **Sueoka N** (1962) On the Genetic Basis of Variation and Heterogeneity of DNA Base Composition. Proc Natl Acad Sci USA **48**:582–592

[88] **Sueoka N** (1988) Directional mutation pressure and neutral molecular evolution. Proc Natl Acad Sci USA **85**:2653–7

[89] **Sueoka N** (1992) Directional mutation pressure, selective contraints, and genetic equilibria. J Mol Evol **34**:95–114

[90] **Tatusov RL, Koonin EV, Lipman DJ** (1997) A genomic perspective on protein families. Science **278**:631–637

[91] **Temin H, Mizutani S** (1970) RNA-dependent DNA polymerase in virions of Rous sarcoma virus. Nature **226**(252):1211–3

[92] **Williams KL, Gooley AA, Packer NH** (1996) Proteome: Not just a made–up name. Todays Life Sciences S. 16–21

[93] **Winkler H** (1920) Verbreitung und Ursache der Parthenogenesis im Pflanzen und Tierreich, Fischer, Jena

[94] **Wolfe KH, Sharp PM, Li WH** (1989) Rates of synonymous substitution in plant nuclear genes. J Mol Evol **29**:208–211

[95] **Zuckerkandl E** (1987) On the molecular evolutionary clock. J Mol Evol **26**:34–46

[96] **Zuckerkandl E, Pauling L** (1965) Molecules as documents of evolutionary history. J Theor Biol **8**(2):357–66