Praktikum Algorithmischen Anwendungen WS 2006/07
Bioinformatik
Sequence-Aligment Team C rot Ala0607

Blöink, Frank 11038619 ai600@gm.fh-koeln.de
Wolters, Benjamin 11037092 ai651@gm.fh-koeln.de

26. Januar 2007

Inhaltsverzeichnis

1 Einführung **1**

2 Grundlagen **2**
2.1 Bioinformatik 2
2.2 Alignment 2
2.2.1 Globales Alignment 2
2.2.2 Lokales Alignment 2

3 Dynamische Programmierung **3**

4 Needleman-Wunsch-Verfahren **3**
4.1 Mathematische Definition 4
4.2 Bildung des Alignments 5
4.3 Laufzeitverhalten 6

5 Smith-Waterman-Algorithmus **7**
5.1 Mathematische Definition 7
5.2 Bildung des Alignments 8
5.3 Laufzeitverhalten 8

6 Quellen **8**

1 Einführung

Diese Ausarbeitung enstand zum Praktikum im Fach Algorithmische Anwendungen und behandelt ausgewählte Algorithmen der Bioinformatik. Im speziellen wollen wir im folgenden Dokument Grundlegende Algorithmen zur DNA-Sequenzanalyse vorstellen. Die unter dem Begriff *Sequence-Alignment* bekannten Verfahren beruhen auf den aus dem allgemeineren Themenbereich des Pattern-Matching bekannten String-Algorithmen, da die DNA-Sequenzen generell als

einfache Zeichenstrings ber einem begrenzten Alphabet repräsentiert werden. Wir werden daher nur kurz auf den biologischen Hintergrund als einem vom vielen Anwendungsgebieten fr diese Algorithmen eingehen. Zu den wohl bekanntesten Verfahren aus diesem Themenbereich zählen neben dem Needleman-Wunsch auch der Smith-Waterman Algorithmus, welche wir im folgenden näher vorstellen werden.

2 Grundlagen

2.1 Bioinformatik

Bioinformatik ist eine interdisziplinäre Wissenschaft, die biologische Fragestellungen mit Methoden der Informatik zu beantworten versucht. Die Bioinformatik beschätigt sich dabei unter anderem mit Fragen von biologischen Sequenzen (DNA und Proteine), deren Speicherung und deren Vergleich. Ziel ist dabei, aus den gewonnenen genomischen Datenmengen zur Interpretation funktionelle oder evolutionäre Ähnlichkeiten herauszufiltern. Der Vergleich der DNA- und Proteinsequenzen geschieht dabei über das Sequence Alignment, also das gegeneinander Ausrichten zweier Sequenzen durch Einfügen von Gaps.

2.2 Alignment

Der Grundgedanke bei dem Alignment, also dem Vergleich der Sequenzen besteht darin, die Sequenzen so untereinander zu schreiben, dass möglichst viele gleiche Buchstaben in einer Spalte stehen und in die Sequenzen möglichst wenig Gaps, also Lücken oder Platzhalter, eingefügt werden müssen. Das jeweilige Alignment orientiert sich dabei an einer Bewertung der Übereinstimmungen und Lücken, um die verschiedenen Möglichkeiten abzuwägen. Diese Bewertung erfolgt mittels einer Scoring-Funktion, welche für jede Übereinstimmung zwischen den Buchstabenpaaren der Sequenzen eine positive Punktzahl vergibt und das Einfügen von Lücken Punktabzug bestraft. Das Alignment mit der höchsten Punktzahl ist dann ein optimales Alignment. Dieses zu finden ist ein Optimierungsproblem, welches beim paarweisen Alignment mit der Methode der dynamischen Programmierung relativ effizient gelöst werden kann.

2.2.1 Globales Alignment

Bei einem globalen Alignment zwischen zwei Sequenzen werden alle Symbole berücksichtigt. Globale Alignments werden hauptsächlich verwendet, wenn die zu untersuchenden Sequenzen ähnlich lang sind und starke Sequenzhomologien erwartet werden. Sequenzhomologie bedeutet hierbei, dass die Ähnlichkeiten von homologen Merkmalen evolutionär auf ein und dieselbe Struktur bei einem gemeinsamem Vorfahren zurückgehen, sich aber auseinander entwickelt haben können und daher nicht (mehr) die gleichen Funktionen erfüllen müssen. Sie sind dann bezüglich ihrer Herkunft äquivalent.

2.2.2 Lokales Alignment

Die Berechnung des optimalen lokalen Alignment hat eine andere Anwendung als die Berechnung des optimalen globalen Alignment. Methoden zum Finden

von lokalen Alignments werden verwendet, wenn zwei Sequenzen auf Homologien untersucht werden sollen, jedoch keine Übereinstimmung auf der gesamten Länge der Sequenz zu erwarten ist. Das heißt ein lokales Alignment ist auf Teilbereiche der Sequenz beschränkt. Beispiele sind hierbei die Suche nach gleichen Sequenzmotiven bei Proteinen

3 Dynamische Programmierung

Unter dem Optimalitätsprinzip von Bellman versteht man ein Optimierungsproblem, das aus vielen gleichartigen Teilproblemen besteht, und eine optimale Lösung des Problems sich aus optimalen Lösungen der Teilprobleme zusammensetzt. Das Paradigma der dynamischen Programmierung bietet eine algorithmische Lösung von Optimierungsproblemen. Das Verfahren der dynamischen Programmierung besteht darin, zuerst die optimalen Lösungen der kleinsten Teilprobleme direkt zu berechnen, und diese dann geeignet zu einer Lösung eines nächstgrößeren Teilproblems zusammenzusetzen, solange bis eine Gesammtlösung entsteht. Man versucht Rekursion durch zwischenspeichern von Teillösungen in einer Tabelle, um später auf sie zurückgreifen zu können, zu vermeiden.

Anhand von Abbildung 1 wird versucht die dynamische Programmierung schematisch zu erklären. Nehmen wir an, es wird zum maximalem Wert eines Vorgängers 1 hinzugerechnet und mögliche Vorgänger befinden sich links, oben oder oben links.

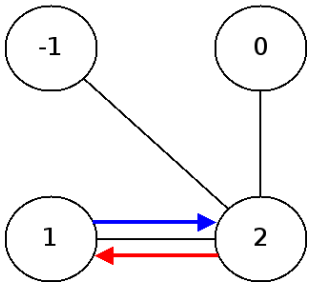

Abbildung 1: Beispiel

In den Kreisen sind Teillösungen eines Teilproblems dargestellt. ein weiteres Teilproblem besteht darin sich den Weg zu merken wie man auf die nächste Teillösung gekommen ist. Mögliche Wege sind als schwarze Kanten dargestellt, der Weg von dem auf den nächsten Punkt gestoßen wurde in blau. Nun speichert man sich die Werte zwischen um später den roten Weg zurückberechen zu können.

4 Needleman-Wunsch-Verfahren

Der Needleman-Wunsch Algorithmus ist ein Verfahren der Bioinformatik. Mittels dynamischer Programmierung wird mit diesem Verfahren ein optimales globales Alignment zweier Sequenzen gefunden. Das Verfahren benötigt zwei Sequenzen a und b, eine Matrix D der größe $D(n+1, m+1)$, wobei $n = len(a)$ und

$m = len(b)$ ist. Weiterhin wird eine Bewertungsfuntktion $w(a_i, b_j)$ benötigt, die zwei Nukleotide vergleicht und bewertet. Die Matrix wird mit einer Rekursionsgleichung erstellt, die die Teilalignments und das globale Alignment berechnet.

4.1 Mathematische Definition

Die einzelnenen Teilprobleme mit der zugehörigen Teillösung werden im folgenden datailhiert betrachtet. Dem Algorithmus werden zuerst zwei zu vergleichende Sequenzen übergeben, a und b.

Abbildung 2: Sequenzen A & B

$$\begin{array}{lllllll} Sequenz\ a: & A & C & G & T & C & E \\ Sequenz\ b: & A & G & T & C & D & E \end{array}$$

Sequenz a und b besitzen die Größe 6. Nun wird die Matrix D zum speichern der Teillösungen angelegt. Dafür wird zur Länge eine Zeile und Spalte hinzugefügt um eine Gap zu ermöglichen. Daraus resultiert eine Matrix $D(7, 7)$, Abbildung 3.

Abbildung 3: Matrix

$$D_{init} = \begin{bmatrix} & - & A & G & T & C & D & E \\ - & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ G & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ T & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ E & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Eine Angabe der Sequenz inklusive der Gaps macht mathematisch keinen Sinn, soll hier jedoch der Übersichtlichkeit dienen.

Nun erstellen wir eine Bewertungsfunktion $w(a_i, b_j)$, die im Falle eines Alignments, also wenn a_i und b_j übereinstimmen, eine 1 zurückgibt und im Falle keiner Übereinkunft dies mit einer -1 bestraft.

Abbildung 4: Bewertungsfunktion

$$w(a_i, b_j) = \begin{pmatrix} 1 & falls\ a_i = b_j \\ -1 & sonst \end{pmatrix}$$

Die Berechnung der einzelnen Zellern der Matrix erfolgt anhand folgender Rekursionsfunktion, dabei wird der maximale Wert des jeweiligen Vorgängers zur Bewertung der beiden Proteine hinzu addiert und in der Matrix gespeichert.

Abbildung 5: Rekursionsfunktion

$$D_{i,j} = max \begin{pmatrix} D_{i-1,j-1} + w(a_i, b_j) \\ D_{i-1,j} + w(a_i, -) \\ D_{i,j-1} + w(-, b_j) \end{pmatrix}$$

Hierbei ist zu beachten, das $D(0,0)$ mit 0 initialisiert und in Spalte und Zeile 0 nur gültige, also keine außerhalb der Matrix, Werte zur berechnung herangezogen werden. In Abbildung 6 wurde die Matrix mittels der Rekursionsgleichung aus Abbildung 5 berechnet.

Abbildung 6: berechnete Matrix

$$D_{align} = \begin{bmatrix} & - & A & G & T & C & D & E \\ - & 0 & -1 & -2 & -3 & -4 & -5 & -6 \\ A & -1 & 1 & 0 & -1 & -2 & -3 & -4 \\ C & -2 & 0 & 0 & -1 & 0 & -1 & -2 \\ G & -3 & -1 & 1 & 0 & -1 & -1 & -2 \\ T & -4 & -2 & 0 & 2 & 1 & 0 & -1 \\ C & -5 & -3 & -1 & 1 & 3 & 2 & 1 \\ E & -6 & -4 & -2 & 0 & 2 & 2 & 3 \end{bmatrix}$$

4.2 Bildung des Alignments

Das Alignment der jeweiligen Sequenz läßt sich durch durch Backtracing ermitteln. Es wird mit dem maximalem Teilelignment begonnen, im Beispiel ist dies $D(n,m) = 3$, und mittels der Rekursionsfunktion zurückberechnet. Läßt sich $D(i-1, j-1)$ mittels der Rekursionsfunktion als Vorgänger von $D(i,j)$ feststellen wird in beiden Alignments a_i bzw. b_j hinzugefügt. Ist zusätzlich $D(i,j)$ kein Teilalignment[1] so entspricht dies einer Mutation der Sequenzen. Insertion und Deletion werden an der Position $i = 5, j = 5$ deutlich, Abbildung 7 zeigt die Rekursionsgleichung

Abbildung 7: $i = 5,\ j = 5$ Mutation

$$D_{5,5} = max \begin{pmatrix} D_{4,4} + w(T,C) & = 1 + -1 & = 0 & \\ D_{4,5} + w(T,-) & = 0 + -1 & = -1 & \\ D_{5,4} + w(-,D) & = 3 + -1 & = 2 & , Mutation \end{pmatrix}$$

[1] $a_i ! = b_j$

Da hier der Index i der Matrix nicht verändert wird und i als Zähler der Sequenz a dient, wird nun in das Alignment von a ein Gap eingefügt, dies entspricht einem Deletion. Abbildung 8 makiert diese Position mittels einer Raute in der Matrix.

Ein Deletion in a bedeutet ein Insertion im Alignment von b, da die Indexvariable j dekrementiert wird.

Abbildung 8: Sequent A Deletion

$$D_{align} = \begin{bmatrix} & - & A & G & T & C & D & E \\ - & 0 & -1 & -2 & -3 & -4 & -5 & -6 \\ A & -1 & 1 & 0 & -1 & -2 & -3 & -4 \\ C & -2 & 0 & 0 & -1 & 0 & -1 & -2 \\ G & -3 & -1 & 1 & 0 & -1 & -1 & -2 \\ T & -4 & -2 & 0 & 2 & 1 & 0 & -1 \\ C & -5 & -3 & -1 & 1 & 3\# & 2\# & 1 \\ E & -6 & -4 & -2 & 0 & 2 & 2 & 3 \end{bmatrix}$$

Abbildung 9 zeigt die beiden Alignments von a und b. Sequenz a wurde an der Stelle 2 ein Molekül hinzugefügt, dies entspricht dem Insertion und an Stelle 6 ein Molekül entnommen, Deletion. In der verglichenen Sequenz ist an diesen Stellen der gegenteilige Effekt eingetreten, da ein Isertion in einer Sequenz immer ein Deletion in der anderen Sequenz bedeutet.

Abbildung 9: Alignments

	1	2	3	4	5	6	7
$AlignmentA:$	A	C	G	T	C	$-$	E
$AlignmentB:$	A	$-$	G	T	C	D	E

4.3 Laufzeitverhalten

Der Needleman-Wunsch-Algorithmus hat eine Laufzeitkomplexität von $\mathrm{O}(nm)$. Diese Laufzeit entsteht, da der Algorithmus jedes der $n * m$ Felder mit einer Bewertung belegen muss. Die Berechnung der Bewertung funktioniert in der Regel in $\mathrm{O}(1)$, allerdings gibt es Bewertungsprobleme für spezielle biologische Problemstellungen die eine komplexere Bewertungfunktion benötigen. Die Rekursionsfunktion läuft ebenfalls in $\mathrm{O}(1)$. Fassen wir nun alle Aktivitäten zusammen. Die Matrix wird also in $\mathrm{O}(nm)$ Zeit berechnet.

Ein Alignment wird durch das Backtracing berechnet. Zum Backtracing werden nur zuvor zwischengespeicherte relevante Werte hinzugezogen und kann maximal der Länge der längsten Sequence wiederholt werden, da i Rekursionsgleichung in mindestens einer Sequenz den Index dekrementiert. Daraus ergibt sich dann folgende Gleichung:

$$\mathrm{O}(nm) = \mathrm{O}(1) * \mathrm{O}(1) * \mathrm{O}(nm) + \mathrm{O}(1) * \mathrm{O}(1) * max(n, m)$$

Die asymptotische Laufzeit beträgt also $\mathrm{O}(nm)$ und der Speicherbedarf Θnm, da die gesamte Matrix gespeichert wird.

5 Smith-Waterman-Algorithmus

Der Smith-Waterman-Algorithmus berechnet ein lokales Alignment, er basiert auf den Needleman-Wunsch-Algorithmus. Modifiziert wurde Rekursionsfunktion und Bewertungsfunktion. Beim Smith Waterman Algorithmus werden häufig mehrere lokale Alignments gefunden.

5.1 Mathematische Definition

Die beiden Sequenzen aus dem Neddleman Wunsch Beipiel werden nun mittels des Smith Waterman Algorithmuses alignniert

Abbildung 10: Sequenzen

$$\begin{array}{lllllll} SequenzA: & A & C & G & T & C & E \\ SequenzB: & A & G & T & C & D & E \end{array}$$

Die Matrix wird im Unterschied zum Needleman Wunsch Algorithmus komplett mit 0 initialisiert. Dadurch wird es möglcih auch Teilalignments, die sehr stark verschoben sind, zu finden.

Abbildung 11: Initialisierte Matrix

$$D_{init} = \begin{bmatrix} & - & A & G & T & C & D & E \\ - & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ G & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ T & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ E & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Der Rekursionsgleichung wurde um eine weitere Bedingung erweitert, mit dieser ist sichergestellt, das an jeder Position in einer der Sequenzen ein lokales Alignment beginnen kann.

Abbildung 12: modifizierte Rekursionsfunktion

$$D_{i,j} = max \begin{pmatrix} 0 \\ D_{i-1,j-1} + w(a_i, b_j) \\ D_{i-1,j} + w(a_i, -) \\ D_{i,j-1} + w(-, b_j) \end{pmatrix}$$

Wird nun kein Alignment an der Position $a_i,\ b_j$ gefunden, bestraft die neue Bewertungsfunktion diesen Missmatch mit -3. Somit werden Alignments mit mehreren Gaps erschwert, bzw. bei kleinen Sequenzen wie im Beispiel unmöglich.

Abbildung 13: modifizierte Bewertungsfunktion

$$w(a_i, b_j) = \begin{pmatrix} 1 & falls\ a_i = b_j \\ -3 & sonst \end{pmatrix}$$

5.2 Bildung des Alignments

Um Unterschied zum Needleman Wunsch Algorithmus wird beim Smith Waterman Algorithmus nicht bei $D(n, m)$ mit dem Backtracing begonnen, sondern am maximalem Teilalignment der Matrix. Im Beispiel an der Position $D(4, 6) = 3$ und dann mittels Backtracing ein Alignment gebildet bis eine Zelle mit dem Wert 0 erreicht wird. Beim Smith Waterman Algorithmus ist es möglich und erwüscht, das viele lokale Alignments gefunden werden.

Abbildung 14: berechnete Matrix

$$D_{align} = \begin{bmatrix} & - & A & G & T & C & D & E \\ - & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ G & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ T & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ C & 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ E & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Das gefundene lokale Alignment wird in der folgenden Abbildung dargestellt.

Abbildung 15: lokales Alignment

$$\begin{array}{lccc} SequenzA: & G & T & C \\ SequenzB: & G & T & C \end{array}$$

5.3 Laufzeitverhalten

Das Laufzeitverhalten gleicht dem des des Needleman Wunsch, jedoch gilt auch hier die Abweichung bei spezifischen biologischen Problemen

6 Quellen

- National Center for Biotechnology Information, U.S. National Library of Medicine, http://www.ncbi.nlm.nih.gov/
- Needleman, S.B. & Wunsch, C.D. 1971 “A general method applicable to the search for similarities in the amino acid sequence of two proteins”, Journal of Molecular Biology 48:443-453

- Temple F. Smith and Michael S. Waterman 1981, “Identification of Common Molecular Subsequences”, Journal of Molecular Biology, 147:195-197
- Altschul, Gish, Miller, et.al. 1990 Basic local alignment search tool. Journal of Molecular Biology 215. p. 403-410