Pairwise Alignment

Steffen Forkmann

Proseminar: BioInformatik

Wintersemester 2004/2005

Inhaltsverzeichnis

1 Problemstellungen

In der Informatik gibt es viele Problemstellungen, die fehlertolerante Mustervergleiche erfordern. Hierunter fallen die Spracherkennung und der Vergleich von Fingerabdrücken.

1.1 Rechtschreibkorrektur

Eines der bekanntesten Beispiele für solche Problemstellungen ist die Rechtschreibkorrektur wie man sie heute in modernen Textverarbeitungsprogrammen wie Microsoft Word findet. Hier wird z.B. für DNA als Wort DANN vorgeschlagen.

1.2 DNA- und Aminosäure-Sequenzen

Für Biologen sind besonders Sequenzuntersuchungen ein interessantes Problemfeld. So lassen sich Protein- oder DNA-Sequenzen, die im Labor entdeckt wurden, hinsichtlich ihrer Sequenzähnlichkeit katalogisieren und geben so Aufschluß über ihre Funktionalität.

2 Was ist ein Alignment?

Abstrahiert man eine biologische Sequenz als einen eindimensionalen String, so kann der fehlertolerante Mustervergleich in der Molekularbiologie als zeichenweiser Vergleich von zwei Strings verstanden werden. Ziel dieses Vergleiches ist eine Aussage über den Grad der Abweichung der beiden Strings.

2.1 Globales Alignment

In einem globalen Alignment werden die Sequenzen so durch Einfügen von "-" verlängert, dass möglichst viele Sequenzteile die übereinstimmen in beiden Verlängerungen an der selben Position stehen. Es werden dabei immer die beiden kompletten Sequenzen betrachtet.
Will man Beispielsweise die Sequenzen „ANNA" und „HANA" vergleichen, ist unter vielen anderen auch folgendes Alignment möglich:

```
-ANNA
HAN-A
```

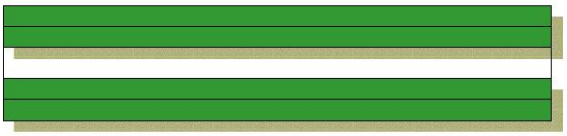

Abbildung 1: Globales Alignment nach [Mül04]

2.1.1 Definition

- Ein globales Alignment von A und H sind zwei Strings A' und H' mit
  - E ... endliches Alphabet, E' = E ∪ "-"
  - $A = a_1 + a_2 + \ldots + a_n$ mit $a_i \in E$
  - $H = h_1 + h_2 + \ldots + h_m$ mit $h_i \in E$
  - $A' = a'_1 + a'_2 + \ldots + a'_k$ mit $a'_i \in E'$
  - $H' = h'_1 + h'_2 + \ldots + h'_k$ mit $h'_i \in E'$
  - $n, m \leq k \leq n + m$
  - Entfernt man alle "-" aus H', A' erhält man H, A

Als Folge dieser Definition ergibt sich, dass es viele Möglichkeiten gibt zwei beliebige Sequenzen zu alignieren. Um eine Aussage über die Güte der gerade untersuchten Anordnung zu geben, müssen die Sequenz-Alignments bewertet werden. Dies geschieht wie weiter unten beschrieben durch Einführung einer „Scoring-Funktion".

2.2 Lokales Alignment

Abbildung 2: Lokales Alignment nach [Mül04]

Proteine sind oftmals aus einem Repertoire aus ähnlichnen Untereinheiten aufgebaut, obwohl sie global wenig Ähnlichkeit zeigen. Die Suche nach gemeinsamen Untereinheiten kann mittels lokalem Alignment erfolgen.
Ein optimales lokales Alignment ist also ein optimales globales Alignment über die Substrings von A und H. Das heißt es gibt kein Paar von Substrings von A und H das einen höheren Alignment-Score hat.

3 Transformationsoperationen

Um die beiden gegebenen Sequenzen an einander auszurichten, bedarf es einiger Operationen. Hierbei werden die drei Operationen Deletion, d.h das Entfernen eines Zeichens aus A, Insertion - das Einfügen eines Zeichens in H und die Substitution - das Ersetzten eines Zeichens verwendet.

Deletion	Insertion	Substitution
Löschung von a_i in A → Einfügen von "-" in H'	Einfügen eines Buchstaben an h_i → Einfügen von "-" in A'	Ersetzen von h_j durch a_i

Tabelle 1: Transformationsoperationen

Das folgende Beispiel veranschaulicht die drei Operationen noch einmal grafisch:

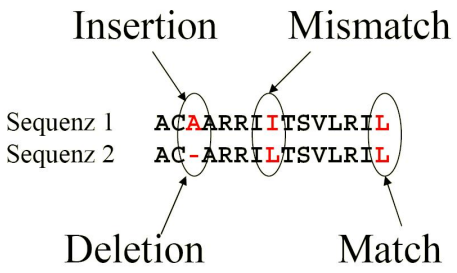

Abbildung 3: Transformation nach [Mül04]

4 Scoring-Funktion

Um unterschiedliche Alignments bewerten zu können, wird eine Scoring-Funktion eingeführt. Diese kann für verschiedene Problemstellungen sehr unterschiedlich aussehen.
Für Aminosäuren gibt es z.B. die Möglichkeit die Bewertungsfunktion als 20 x 20 Matrix aufzufassen. Dazu werden folgende Modellannahmen getroffen:

- Alle Aminosäure-Substitutionen sind symmetrisch.
- Die Aminosäure-Häufigkeiten ändern sich nicht über die evolutionäre Zeit.
- Die Aminosäuren eines Proteins evolvieren unabhängig.
- Die einzelnen Aminosäuren-Positionen haben kein Gedächtnis.
  ($X \to Y \to Z \to X$ ist möglich)
- Der Konservation von seltenen Aminosäure wird höher bewertet, als der Erhalt von häufig vorkommenden.
- Der Austausch von zwei ähnlichen Aminosäuren wird höher bewertet als der von strukturverschiedenen. Das bedeutet es werden die physikalisch-chemischen Eigenschaften der Aminosäuren berücksichtigt.

4.1 BLOSUM - BLOcks SUbstitution Matrix

Eine der wichtigsten Scoring-Funktionen für Aminosäuren sind die 1992 von Henikoff und Henikoff eingeführten BLOSUM-Matrizen. Diese bauen auf die BLOCKS-Datenbank auf, die ca. 3000 multiple Alignments von konservierten Sequenzen (Blocks) aus ca. 800 Proteinfamilien enthält. In dieser Datenbank sind nur Substitutionen, jedoch keine Insertionen und Deletionen enthalten. Demnach stellt die BLOCKS-Datenbank eine gute Datenquelle zur Bestimmung der Austauschwahrscheinlichkeit zwischen Aminosäuren dar.

Als Beispiel sei hier die BLOSUM62 angegeben:

	Cystein	Serin	Threonin	Prolin	Alanin	Glycin	Asparagin	Aspartat	Glutamat	Glutamin	Histidin	Arginin	Lysin	Methionin	Isoleucin	Leucin	Valin	Phenylalanin	Tyrosin	Tryptophan
Cystein	9																			
Serin	-1	4																		
Threonin	-1	1	5																	
Prolin	-3	-1	-1	7																
Alanin	0	1	0	-1	4															
Glycin	-3	0	-2	-2	0	6														
Asparagin	-3	1	0	-2	-2	0	6													
Aspartat	-3	0	-1	-1	-2	-1	1	6												
Glutamat	-4	0	-1	-1	-1	-2	0	2	5											
Glutamin	-3	0	-1	-1	-1	-2	0	0	2	5										
Histidin	-3	-1	-2	-2	-2	-2	1	-1	0	0	8									
Arginin	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	5								
Lysin	-3	0	-1	-1	-1	-2	0	-1	1	1	-1	2	5							
Methionin	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	5						
Isoleucin	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	4					
Leucin	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	4				
Valin	-1	-2	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	4			
Phenylalanin	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	6		
Tyrosin	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	7	
Tryptophan	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	11

Abbildung 4: Beispiel für BLOSUM62

4.2 Gap-Penalties

Um die Scoring-Funktion zur für Alignments zu vervollständigen, muss man noch die mit „-“markierten Stellen (Gaps) bewerten.

4.2.1 Lineare Gap-Penalties

Unter Annahme, dass Gaps in der Natur unwahrscheinlicher sind je länger sie sind kann man lineare Gap-Penalties einführen. Diese haben immer die Form:

- $W = d * k$
  - k . . .Länge des Gaps
  - d . . . Gap opening penalty

4.2.2 Affine Gap-Penalties

Der Nachteil der linearen Gaps ist, dass lange Inserts oder Deletionen überbewertet werden. Als Abhilfe kann man affine Gap-Penalties einführen:

- $W = d + (k * e)$
  - e . . . Gap extension penalty ($e < d$)

Diese sind jedoch wesentlich schwerer zu berechnen, so dass in der Praxis oftmals nur mit linearen Gap-Penalties gerechnet wird.

4.3 Alignment-Matrix

Das optimale globale Aligment kann man relativ leicht aus der folgendermaßen rekursiv definierten Matrix ableiten:

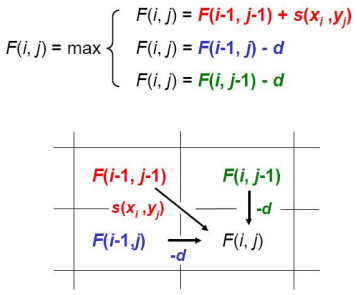

Abbildung 5: Alignment-Matrix

Implementiert man dies jedoch direkt auf diese Weise, so ist die Berechnung sehr aufwändig. Abhilfe schafft hier die dynamische Programmierung deren Idee es ist an einem Ende der Sequenz zu starten und dann solange einzelne Symbole zur aktuell besten Lösung hinzuzufügen bis alle Symbole verbraucht sind.

4.4 Needleman-Wunsch-Algorithmus - Globales Alignment

Ein spezielle Implementierung dieser Idee liefert der Needleman-Wunsch-Algorithmus.

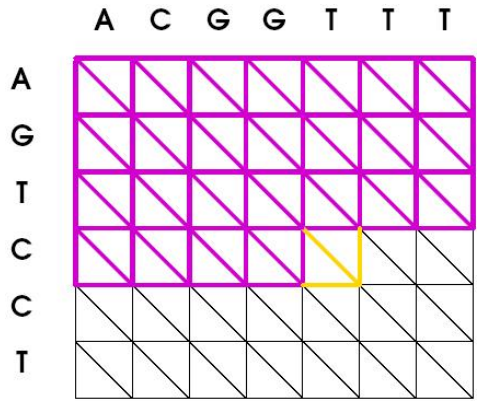

Abbildung 6: Needleman-Wunsch-Algorithmus nach [Met03]

- Man berechne F_{ij} Zeile für Zeile von links nach rechts
- Ergebnisse zwischenspeichern
- $\rightarrow$ Es stehen die benötigten Werte immer rechtzeitig zur Verfügung
- $\Rightarrow$ Laufzeit nur noch $O(m * n)$

4.5 Traceback

Am Ende der Zellenberechnung kann man die optimalen Alignments durch ein sogenanntes Traceback aus der Alignment-Matrix ableiten. Dazu läuft man einfach vom Feld $F(i, j)$ durch Matrix und geht in das Feld zurück, dass das Maximum im Needleman-Wunsch-Algorithmus geliefert hat. Man findet so alle optimalen Alignments in linearer Laufzeit.

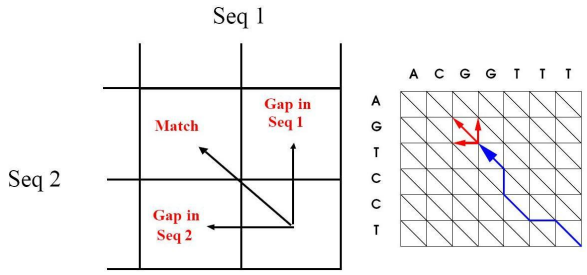

Abbildung 7: Traceback nach [Met03]

4.6 Bespiel globales Alignment

- Globales Alignment HEAGAWGHEE mit PAWHEAE
- Score-Matrix BLOSUM50
- Linearer Gap-Penalty ($d = 8$)

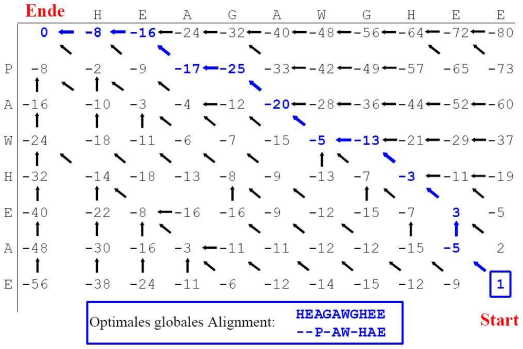

Abbildung 8: Beispiel globales Alignment nach [Dur98]

4.7 Smith-Waterman-Algorithmus - Lokales Alignment

Der Smith-Waterman-Algorithmus ist eine ähnliche Variante, jedoch diesmal zugeschnitten auf lokale Alignments. Er findet somit den längsten Bereich zweier Sequenzen mit der größten Ähnlichkeit. Die Berechnung erfolgt völlig analog zu Needleman-Wunsch. Jedoch mit der Besonderheit, dass wenn alle drei Alignmentmöglichkeiten einen negativen Score liefern, der Score-Wert wieder mit Null initialisert wird und die Kette abgebrochen wird. Als Folge daraus ergibt sich, dass Pfade direkt in der Matrix starten und enden können.

Es ergibt sich also folgende rekursive Definition:

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

4.8 Bespiel lokales Alignment

- Lokales Alignment HEAGAWGHEE mit PAWHEAE
- Score-Matrix BLOSUM50
- Linearer Gap-Penalty ($d = 8$)

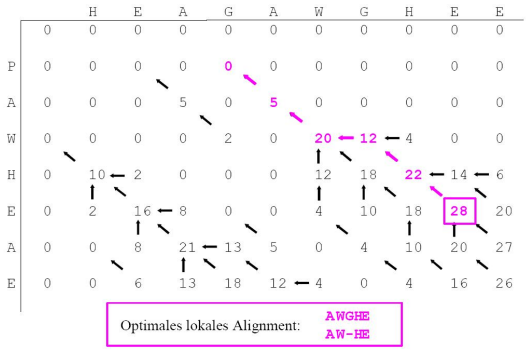

Abbildung 9: Beispiel lokales Alignment nach [Dur98]

5 Zusammenfassung

Mit Hilfe der dynamischen Programmierung lassen sich globales und lokales Alignment relativ einfach berechnen. Als Scoring-Modelle zur Bewertung der unterschiedliche Alignments empfehlen sich z.B. BLOSUM50 gekoppelt mit linearem Gap-Penalty. Das Alignment von zwei Sequenzen ist ohne Probleme möglich. Will man jedoch seine Sequenz mit einer großen Datenbank vergleichen so wird dies sehr rechenintensiv. Abhilfe schaffen hier heuristische Verfahren wie FastA und Blast.

Literatur

[Die98] Jens Peter Dietrich. Seminar bioinformatik. 1998.

[Dur98] Richard Durbin. *Biological Sequence Analysis*. Cambridge University Press, 1998.

[Heu02] Volker Heun. Skriptum zur vorlesung algorithmische bioinformatik i/ii. 2002.

[Kra04] Antje Krause. Scorematrizen - alignmentstatistik - datenbanksuche. 2004.

[Met03] Dirk Metzler. Alignment und phylogenieschätzung. 2003.

[Mül04] Dr. Tobias Müller. Theorie des sequenzalignments und sequenzdatenbanken. 2004.

[Vor98] Prof. Dr. Oliver Vornberger. Parallele algorithmen ss 98 (skript). 1998.