Bioinformatik

Lokale Alignierung

Gapkosten

Silke Trißl / Ulf Leser

Wissensmanagement in der Bioinformatik

Inhalt dieser Vorlesung

- Ähnlichkeit
- Lokales und globales Alignment
- Gapped Alignment

Ähnlichkeit

- Welche Frage will man eigentlich beantworten?
  - Wie weit entfernt sind diese beiden Sequenzen?
  - Wie ähnlich sind sich zwei Sequenzen?
- Editabstand berechnet den (einen) Abstand
  - Großer Abstand – geringe Ähnlichkeit
- Wir können die Sequenzähnlichkeit auch direkt berechnen
  - Intuitives Maß – je ähnlicher, desto höher ist die Wahrscheinlichkeit für ähnliche Funktion
- Dabei werden wir langsam differenzierter
  - Ähnlichkeit einzelner Basen / Aminosäuren
  - Ähnliche Moleküle – positive Werte
  - Unähnliche Moleküle – negative Werte
  - Individuelle Kosten für InsDel einer Base

Formal

- Definition
  Gegeben Alphabet $\Sigma' = \Sigma \cup$ '_', Strings A,B über Σ' mit |A|=|B|=n
  - *Eine Scoringfunktion ist eine Funktion s: $\Sigma' x \Sigma' \rightarrow$ Integer*
    - *Anderer Name: Substitutionsmatrix*
  - *Die Ähnlichkeit von A,B bzgl. der Scoringfunktion s ist*

$$sim(A,B) = \sum_{i=1}^{n} s\left(A[i], B[i]\right)$$

- Bemerkung
  - Zur Anwendung für zwei Sequenzen A,B wählen wir ein Alignment und berechnen dann die Ähnlichkeit der beiden Zeilen zueinander
    - Also mit den eingefügten InsDels
  - Wir suchen natürlich wieder das/die Alignment(s) mit dem höchsten Ähnlichkeitswert

Beispiel

$\sum' = \{A,C,G,T,\_\}$

	A	C	G	T	_
A	4	-2	-2	-2	-2
C		4	-2	-2	0
G			4	-1	0
T				4	-2
_					0

```
AC_GTC
AGGT_C
```
= **3**

```
ACGTC
AGGTC
```
= **14**

```
A_CGTC
AG_GTC
```
= **16**

Berechnung

- Gleiches Prinzip wie zur Berechnung des Editabstands
- Nur kleine Veränderungen

$$d(i,0) = \sum_{k=1}^{i} s(A[k], \_) \qquad d(0,j) = \sum_{k=1}^{j} s(\_, B[k])$$

$$d(i,j) = \max \begin{Bmatrix} d(i,j-1) + s(\_, B[j]) \\ d(i-1,j) + s(A[i], \_) \\ d(i-1,j-1) + s(A[i], B[j]) \end{Bmatrix}$$

Beispiel

Abstand
(Ins/del/Repl: 1,
Match: 0)

		A	G	G	T	C
	0	1	2	3	4	5
A	1	0	1	2	3	4
G	2	1	0	1	2	3
T	3	2	1	1	1	2
C	4	3	2	2	2	1
C	5	4	3	3	3	2

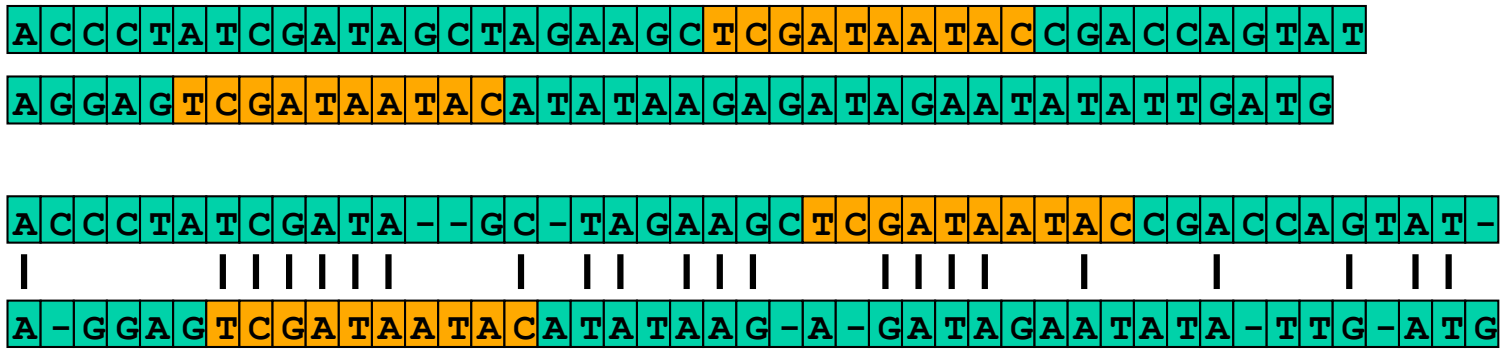
Ähnlichkeit

	A	G	T	C
A	4	-1	-1	-1
G		4	-1	-1
T			4	-1
C				4
_	-3	-3	-3	-3

		A	G	G	T	C
	0	-3	-6	-9	-12	-15
A	-3	4	1	-2	-5	-8
G	-6	1	8	5		
T	-9					
C	-12					
C	-15					

Lokales und globales Alignment

- Bisher: Globale Alignments
  - Beide Sequenzen werden komplett betrachtet
- Das entspricht oft nicht der biologischen Frage
  - Funktion wird nur durch manche Sequenzblöcke bestimmt (Gene, Exons, Proteindomänen, ...)
- Wir suchen meistens ähnliche Teilsequenzen
  - Funktionstragende Sequenzblöcke
  - „Lokale" Alignments

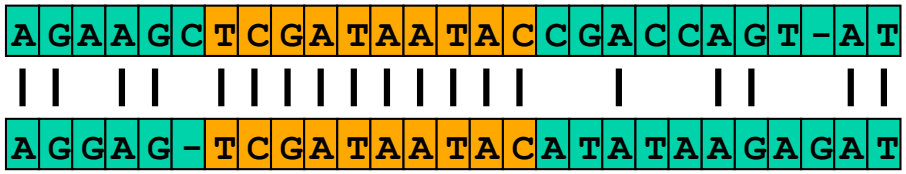


Lokale Alignments

- Definition. *Gegeben zwei Strings A, B.*
  - *Seien a,b Substrings mit a⊆A, b⊆B so dass*

$$sim(a,b) = \max_{\forall a' \in A, b' \in B} \left(sim(a',b') \right)$$

  - *Das vom (globalen) Alignment von a und b induzierte Alignment von A und B heißt lokales Alignment von A und B*
  - *Der lokale Ähnlichkeitsscore* $dist_{local}(A,B) = sim(a,b)$
- Bemerkung
  - Unempfindlich gegen unterschiedliche lange Strings
- Beispiel
  - Lokales A. findet den identischen Substring



Smith-Waterman Algorithmus

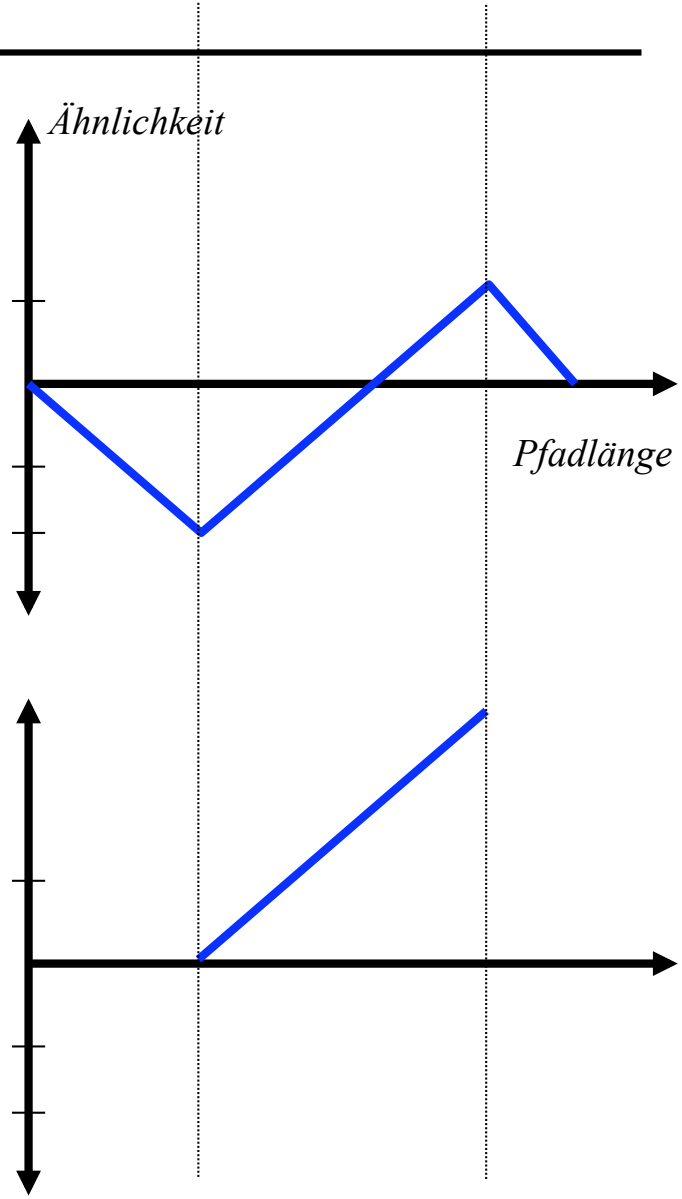
- Smith, Waterman: „Identification of common molecular subsequences“, J. Mol. Bio 147, 1981
- Grundidee
  - Geringfügige Veränderung des Algorithmus für globales Alignment
    - Funktioniert nur mit Ähnlichkeit, nicht mit Abstand
    - Scoring Funktion mit positiven Werten für Matches und negativen Werten für alles andere
  - Eine Reihe von Matches beim Vergleich erzeugt also sukzessive größere Ähnlichkeitswerte
  - Bei Mismatch oder InsDel wird der Wert wieder kleiner
  - Um Regionen mit hoher Ähnlichkeit zu finden, können wir alle Substringmatches mit negativem Gesamtähnlichkeit vergessen
  - Also: Statt in negative Werte zu rutschen, darf der Algorithmus auch bei 0 anfangen

Match: +1
I/R/D: -1

Beispiel

		A	T	G	T	G	G
	0	-1	-2	-3	-4	-5	-6
G				-1			
T					0		
G						1	
A							0

		A	T	G	T	G	G
	0	0	0	-3	-4	-5	-6
G				1			
T					2		
G						3	
A							0



Optimales lokales Alignment

- Theorem
  Gegeben Strings A,B. Mit der folgenden Funktion v(i,j), mit 0≤i≤n und 0≤j≤m

$$v(i,j) = \max \begin{Bmatrix} 0 \\ v(i,j-1) + s(\_, B[i]) \\ v(i-1,j) + s(A[i], \_) \\ v(i-1,j-1) + s(A[i], B[j]) \end{Bmatrix}$$

  - *Gilt:* $dist_{local}(A,B) = \max_{i,j}(v(i,j))$

- Traceback
  - Starte beim maximalen Wert in der Matrix
  - Verfolge beliebigen Pfad bis zu einer Zelle mit Wert 0

Beispiel

Match: +1
I/R/D: -1

		A	T	G	T	C	G
	0	-1	-2	-3	-4	-5	-6
A	-1	1	0	-1	-2	-3	-4
T	-2	0	2	1	0	-1	-2
G	-3	-1	1	3	2	1	0

```
ATGTCG
ATG___

ATGTCG
AT___G

ATGTCG
A__T_G
```

➢ Drei Lösungen, alle mit gleicher Güte

		A	T	G	T	C	G
	0	0	0	0	0	0	0
A	0	1	0	0	0	0	0
T	0	0	2	1	1	0	0
G	0	0	1	3	2	1	0

```
ATGTCG
ATG___
```

➢ Eine Lösung – das lokale Alignment

Wann lokales, wann globales Alignment

- Globales Alignment
  - Genauer Vergleich ähnlicher Sequenzen
  - Z.B. Charakterisierung von Protein-Familien
  - Z.B. Bestimmung einer Consensus-Sequenz für MSA
- Lokales Alignment
  - Finden der interessanten Regionen in unbekannten Sequenzen
    - Z.B. unterschiedliche Spezies, unterschiedliche Gene, ...
  - Finden konservierter (=funktionaler) Subsequenzen
    - Erster Schritt zur Proteinfamilie; danach g. A. innerhalb der Familie
  - Finden konservierter Bereiche
    - Untersuchung von evolutionären Prozessen

Inhalt dieser Vorlesung

- Ähnlichkeit
- Lokales und globales Alignment
- Gapped Alignment
  - Gap = Loch = Zusammenhänge Folge von Spaces = Insertions bzw. Deletions
  - Bisher zählt jedes Space einzeln
  - Ist das immer das richtige Modell?

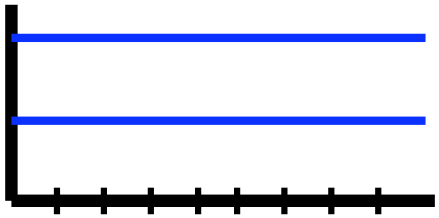
Gaps

- Evolution besteht nicht nur aus Punktmutationen
- Oft werden ganze Blöcke verschoben
  - Beispiele
    - Crossing-Over während Meiose (geschlechtliche Zellteilung)
    - „Versehentliche" Duplikation von Sequenzen
    - Transposable Elements
- Dazwischen: lange Reihen von Inserts / Deletions
- Nur die „guten" Blöcke sind relevant, die Zwischenräume nicht
  - Exons und Introns unterliegen unterschiedlichem Selektionsdruck
  - Proteine setzen sich aus „active sites" und variablen Zwischensequenzen zusammen

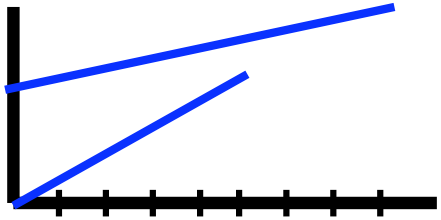
Bewertung von Gaps

- Gesucht: Gapscorefunktionen, die Gaps unterschiedlicher Länge bei der Berechnung der Sequenzähnlichkeit flexibel bewerten können
- Sei w(k) der Score eines Gaps der Länge k
- Folgende Funktionsklassen
  - Konstanter Gapscore c
    - w(k) = c
    - Score unabhängig von Gaplänge
  - Linearer Gapscore mit Kosten für Gapbeginn w_s und Gapfortsetzung w_f
    - $w(k) = w_s + w_f * k$
    - Bisheriges Modell nimmt $w_s=0$ und $w_f=1$ an
  - Konvexer Gapscore - Score wird nie kleiner mit wachsendem k, aber immer langsamer größer
    - $w(k) = f(k)$; mit $f'(k)>0$ und $f''(k)\leq 0$
    - Beispiel: w = log(k)
  - Beliebiger Gapscore - Score kann wachsen, fallen, oszillieren, ...
    - w(k) = f(k); mit f(k) beliebig

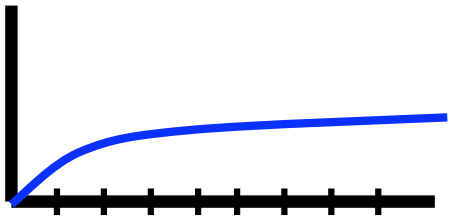
Klassen von Gapscorefunktionen



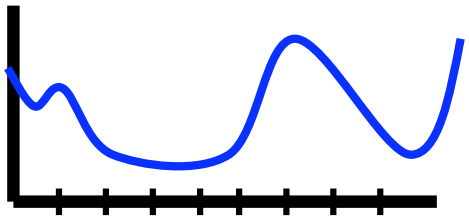
- Konstanter Gapscore



- Linearer Gapscore



- Konvexer Gapscore



- Beliebiger Gapscore

Auswirkungen auf Berechnung

- Je flexibler die Scorefunktion, desto komplexer die Berechnung
  - Konstant oder linear: $O(nm)$
  - Konvex: $O(nm \log m)$
  - Beliebig: $O(nm^2 + n^2m)$