

Sequenzen-Alignierung in der Bioinformatik

VO Algorithm Engineering

Professor Dr. Petra Mutzel

Lehrstuhl für Algorithm Engineering, LS11

22. VO

23.01.2006

1

Literatur für diese VO

- Volker Heun: Skriptum zur Vorlesung Algorithmische Bioinformatik I/II, SS05 und WS05/06, LMU München, Kap. 5 (und 8)
- Neil C. Jones und Pavel A. Pevzner: An Introduction to Bioinformatics Algorithms, MIT Press, Cambridge, 2004
- Folien(-teil): Danke an Gabriele Koller, TU Wien

2

Überblick

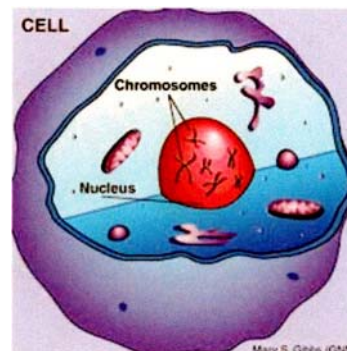
Einführung in die Molekularbiologie

Vergleich von Sequenzen

Paarweise Sequenzen Alignierung

3

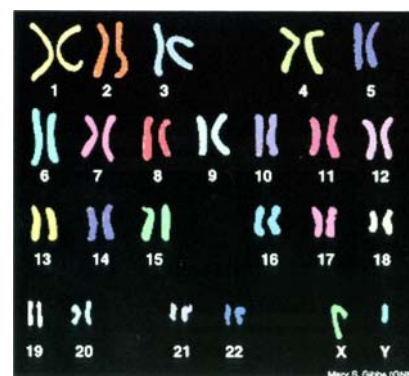
Einführung in die Molekularbiologie



4

Organism	Anzahl an Chromosomen
pea plant	14
sun flower	34
cat	38
puffer fish	42
human	46
dog	78
Goldfische	94

5



6



Chromosome

DNA

Gene

DNA: Doppelhelix mit Nucleotiden aus Zucker, Phosphat und Basen (Adenin, Cytosin, Guanin, Thymin)

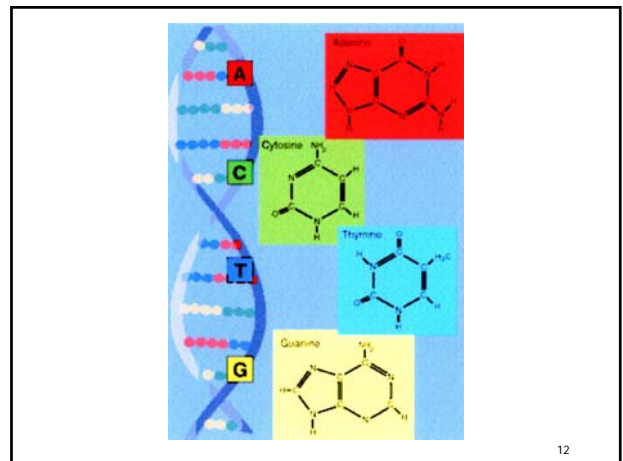
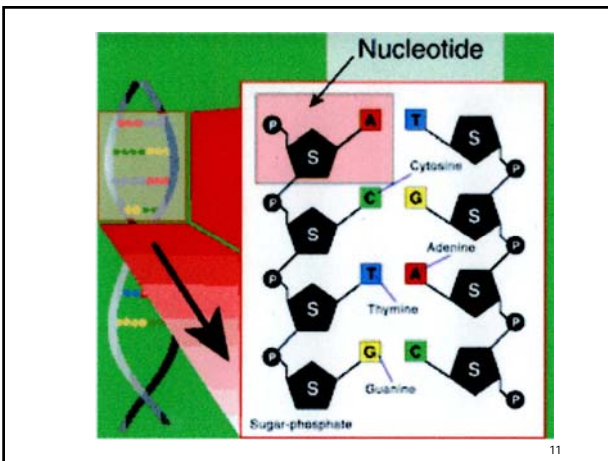
8



Genom: nur 5% Gene

Organism	Number of genes in the genome
<i>Mycoplasma genitalium</i>	517
<i>Saccharomyces cerevisiae</i>	6,275
<i>Arabidopsis thaliana</i>	~ 20, 000
<i>Caenorhabditis elegans</i>	19, 099
<i>Haemophilus influenzae</i>	1,743
<i>Drosophila melanogaster</i>	13, 601
<i>Neisseria meningitidis</i>	2, 158
<i>Homo sapiens</i>	~ 30, 000

10



Organismus	Basenpaare	Gene
Bäckerhefe	12 Mill.	6000
Fruchtfliege	137 Mill.	13.601
Mensch	3 Mrd.	~30.000

13

Proteine und Aminosäuren

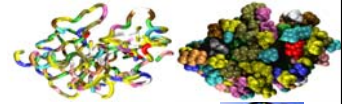
Proteine sind aus Aminosäuren aufgebaut

20 Aminosäuren: durch je Nukleotid-Triplets kodiert

Beispiel: GCA, GCC, GCG → Alanin

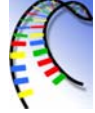
Primär- Sekundär- Tertiärstruktur

ADMVİKAPAG
AKVTKAPVAF
SHKGHASMDC



Proteinsynthese

- Transkription: DNA → mRNA
- Translation: mRNA → Aminosäuren → Protein



14

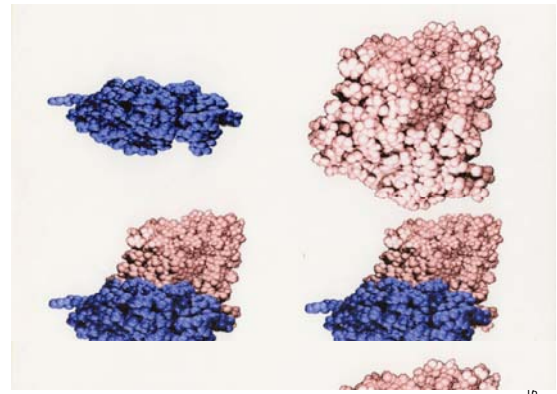
Genforschung

Ziele: besseres Verständnis des Genoms,
z.B. zur Diagnose bzw. Vorbeugung von genetisch
beeinflussten Krankheiten

Aufgaben:

- Sequenzierung eines Gens
- Lokalisierung der Gene innerhalb des Genoms
- Analyse der Funktion eines Gens
- Analyse der DNA außerhalb der Gene
- Beziehungen zwischen Genen
- Vergleich von Genomen

15



16

Aufgaben der Bioinformatik

Unterstützung der Biologen in der Bearbeitung,
Analyse und Interpretation großer Datenmengen

- Biologische Datenbanken
- Methoden zum Vergleich und zur Funktionsvorhersage von Sequenzen
- Auffinden neuer Zusammenhänge in den Daten
- Simulation biologischer Prozesse und von Experimenten an biologischen Systemen

17

2. Vergleich von Sequenzen

Zweck:

- Suche in Datenbanken
- Neue Sequenzen mit bekannten vergleichen
- Basis für viele andere Berechnungen in der Bioinformatik (z.B. phylogenetische Bäume)

1. Vergleich zweier Sequenzen
2. Vergleich mehrerer Sequenzen

18

AAATAAGG G EQGSNME PW NL SQYDY OMGGDY GEG ... QMGAA YF
 VAAATRAQT G EQGSNME PW NL SQYDY OMGGDY GEG ... QMGAA YF
 VOLVSAQR G SQGGGOT PA LW SIWOM ODSEFY GBT ... ENK WS
 AATAAGR G EQGSNME PW NL SQYDY OMGGDY GEG ... QMGAA YF
 AATAAGR G EQGSNME PW NL SQYDY OMGGDY GEG ... QMGAA YF
 SETVESR G AP NL SQYDY OSTDAY GTO ... QSGP BS
 RGSAA G G RQAGDAL PG GL SSWOM OTTYDY GIG ... QSQ DG
 AGFAAAG G QP HF SEFY OTTDAY GDO ... QSGP BS
 AGFAAAG G QP HV SEFY OTTDAY GDO ... QSGP BS
 RGSAA G G RQAGDAL PG GL SSWOM OTTADY GDO ... QSQ DG
 RGSAA G G RQAGDAL PG GL SSWOM OTTADY GDO ... QSQ DG
 YOVAIAQ G RQAGKL PW NL SQWOM OSTDEY SPD HW QSN K K
 EQ G RQAGKL PW NL SQYDY OSSDY SP S KH QSN K K

19

Vergleich zweier Sequenzen

Geg. DNA-Sequenzen: **GACCGATTAG** **GATCGGAATAG**

Mögliche Alignments:

```

GACCGATTAG-
GATCGGAATAG
  
```

```

GA-CGGATTAG
GATCGGAATAG
  
```

Gesucht: bestmögliches Alignment

- Globales Alignment: ganze Sequenzen
- (Lokales und semiglobales Alignment)

20

Ähnlichkeit zweier Sequenzen

Geg. Alignment:

```

GA-CGGATTAG
GATCGGAATAG
  
```

Einfache Bewertung:

Match: 1 z.B. **G** mit **G**
 Mismatch: -1 z.B. **T** mit **A**
 Space: -2 z.B. **-** mit **T**

Gesamtwert: $9 \cdot 1 + 1 \cdot (-1) + 1 \cdot (-2) = 6$
 Ähnlichkeit (Similarity): Wert des besten Alignments

21

Lösungsansatz mittels DP

Geg. Sequenzen s (mit Länge m) und t (mit Länge n) über ein Alphabet Σ

Idee: Berechne alle Ähnlichkeitwerte zwischen Präfixen der beiden Sequenzen \rightarrow Matrix

Es gibt nur 3 Arten, für i, j ein Alignment zu erhalten:
 (zwei Leerzeichen in einer Spalte sind nicht erlaubt)

- ...- mit ... $t[j]$ \leftrightarrow $s[1..i]$ mit $t[1..j-1]$ oder
- ... $s[i]$ mit ... $t[j]$ \leftrightarrow $s[1..i-1]$ mit $t[1..j-1]$ oder
- ... $s[i]$ mit ...- \leftrightarrow $s[1..i-1]$ mit $t[1..j]$

22

Algorithmus Similarity (Needleman-Wunsch)

Input : Sequenzen s und t
Output : Ähnlichkeit zwischen s und t

$m \leftarrow |s|; n \leftarrow |t|;$
 for $i \leftarrow 0$ to m do
 $a[i,0] \leftarrow i \cdot g;$
 for $j \leftarrow 0$ to n do
 $a[0,j] \leftarrow j \cdot g;$
 for $i \leftarrow 1$ to m do
 for $j \leftarrow 1$ to n do
 $a[i,j] \leftarrow \max($
 $\quad a[i-1,j] + g,$
 $\quad a[i-1,j-1] + p(i,j),$
 $\quad a[i,j-1] + g);$
 return $a[m,n];$

Match/Mismatch:
 $p(i,j) = 1$ falls $s[i]=t[j]$
 -1 falls $s[i] \neq t[j]$

Space: $g = -2$

23

Beispiel für DP

Match/Mismatch:
 $p(i,j) = 1$ falls $s[i]=t[j]$
 -1 falls $s[i] \neq t[j]$

Space: $g = -2$

$S: AAAC$
 $t: AGC$
 $m=4$
 $n=3$

Alignments

- AAAC
AG-C
- AAAC
-AGC
- AAAC
A-GC

Edit-Graph

24

Algorithmus $Align(i,j)$

Eindache Rückverfolgung: Überprüfung auf Gleichheit!

Input: Gefüllte Matrix a , Indizes i, j

Output: Optimales Alignment zwischen s und t in sa und ta

if $i = 0$ and $j = 0$ then

$len \leftarrow 0$;

else if $i > 0$ and $a[i,j] = a[i-1,j] + g$ then

$Align(i-1,j,len)$;

$len \leftarrow len + 1$; $sa[len] \leftarrow s[j]$; $ta[len] \leftarrow -$;

else if $i > 0$ and $j > 0$ and $a[i,j] = a[i-1,j-1] + p(i,j)$ then

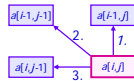
$Align(i-1,j-1,len)$;

$len \leftarrow len + 1$; $sa[len] \leftarrow s[j]$; $ta[len] \leftarrow t[j]$;

else

$Align(i,j-1,len)$;

$len \leftarrow len + 1$; $sa[len] \leftarrow -$; $ta[len] \leftarrow t[j]$;



25

Algorithmenanalyse

Similarity berechnen:

• Laufzeitanalyse:

– Initialisierung 1. Spalte: $O(m)$

– Initialisierung 1. Zeile: $O(n)$

– Restmatrix füllen: $O(mn)$

⇒ Gesamtaufwand: $O(mn)$

• Speicherplatz: $O(mn)$

Optimales Alignment erzeugen:

• für Alignment der Länge len : $O(len) = O(m+n)$

Gibt es schnellere / speichersparendere Algorithmen?

26

Alignierung mittels Kürzester Wege

• Edit Graph (DP): $G=(V,E)$

– V : Matrixeinträge,

– E : Rekursionseinträge

– Kosten auf Kanten:

Kosten der DP Rekursion

• Optimales Alignment:

Kürzester Weg von $s=(0,0)$

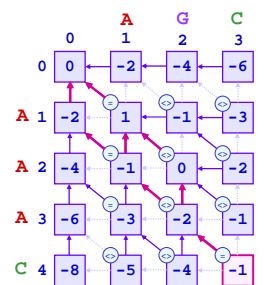
nach $t=(m,n)$ in G

• Beob.: G ist ein DAG

• Dijkstra: $O(|E| + |V| \log |V|)$

• →quadr. Zeit + Platz

• A*-Suche (Goal Directed Unidirectional Search)



Achtung: Ähnlichkeitsmaß ≥ 0

27

A* Algorithmus

Idee: Beschränke (s,t) -Weg Suche auf den relevanten Teil

• Sei $l(u,t)$ untere Schranke eines kürzesten (u,t) -Weges

• Redefiniere Kosten: $c'(u,v) = c(u,v) + (l(v,t) - l(u,t))$

• Wir fordern dabei, dass $l(\dots)$ die Konsistenzbedingung einhält: $c(u,v) \geq l(u,t) - l(v,t)$ für alle $(u,v) \in E$ (weil Gewichte nicht-negativ sein müssen)

• Beh.: Redefinition der Kosten stört nicht, d.h. die Menge der optimalen Wege ändert sich nicht

• Eine einfache untere Schranke: $l((i,j),(m,n)) = ((m-i) - (n-j))g$

• Je besser die Schranke, desto schneller die Suche: wenn l exakt ist, dann werden Kanten genau entlang der kürzesten Wege extrahiert; hingegen bei $l=0$ wird wie bei Dijkstra durchsucht.

28

Beispiellauf A* Algorithmus

$s = \text{MYMISSISAHIPPIE}$

$t = \text{IMISSISSISSIPPI}$

Needleman-Wunsch: $16 \cdot 17 = 272$ Knoten werden besucht

Dijkstra: 165 Knoten werden in PQ eingefügt,
132 Knoten werden extrahiert.

A* (GDUS): 106 Knoten werden eingefügt,
76 Knoten werden extrahiert.

29