Sequenz – Alignment Teil 2

14.11.03

Vorlesung Bioinformatik 1

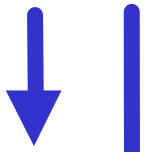
Molekulare Biotechnologie

Dr. Rainer König

Besonderen Dank an Mark van der Linden, Mechthilde Falkenhahn und der Husar Biocomputing Service Gruppe für die Unterstützung bei meinen Folien

Typen von Sequenzvergleichen

- Paarweises Alignment



- Multiples Alignment

- Datenbank-Suchen

Methoden des Sequenz-Alignments

- **Dynamische Programmieren**
- **Dotplot-Analyse**
- **Wort-Methoden für DB-Suchen**

Datenbanksuchen: BLAST

- Fenster-basiert, Wortgröße = 11 (Nukleotide), 3 (Aminosäuren)
- Wenn ein Wort über diesem Schwellenwert gefunden wurde, wird es nach Möglichkeit verlängert
- Verlängerung bricht ab, wenn der Score-Wert zu stark abfällt
- Neuere BLAST-Versionen können auch Gaps verarbeiten

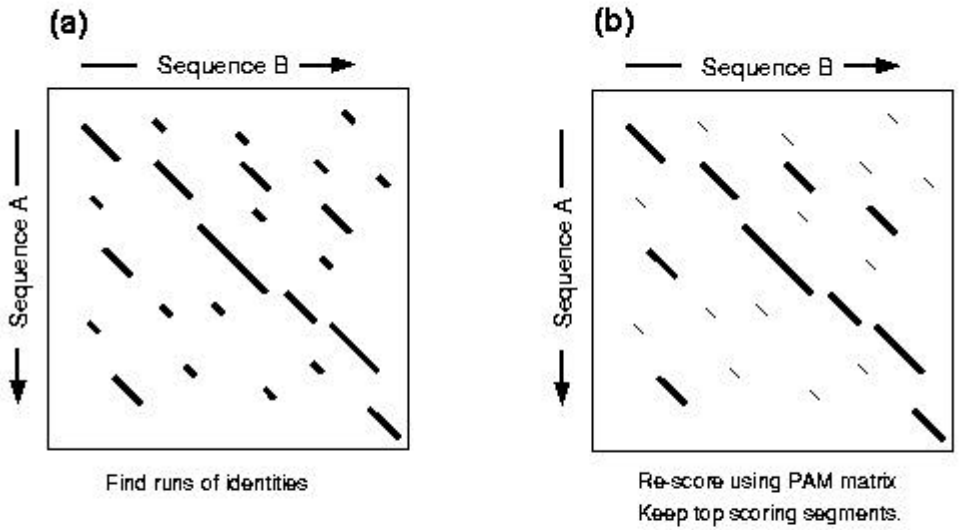
Heuristische Methode, ohne dynamisches Programmieren, schnell aber nicht so sensitiv

Datenbanksuchen: FastA

- Fenster-basierend, Wortgröße 4-6 (Nukleotide), 1-2 (Aminosäuren)
- berechnet Scorewerte für Diagonale, Diagonale mit den besten Scores werden behalten
- Diagonalen werden verlängert
- Diagonalen, die schlecht zu einem Alignment mit der besten Diagonalen passen, werden verworfen
- vollständiges Dynamisches Programmieren innerhalb einer Zone/Band, in dem die verbliebenen Diagonalen sind (rechenintensiv)

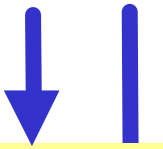
Heuristische Methode, enthält dynamisches Programmieren. Ist langsamer als BLAST, aber dafür sensitiver.

FASTA Algorithm



Typen von Sequenzvergleichen

- Paarweises Alignment



- **Multiples Alignment**

- Datenbank-Suchen

Multiples Sequenz-Alignment:

Begriff: Verfahren, um drei oder mehr Sequenzen zu alignieren

```
Seq. A    N  _  F  L  S
Seq. B    N  _  F  _  S
Seq. C    N  K  Y  L  S
Seq. D    N  _  Y  L  S
```

Motivation

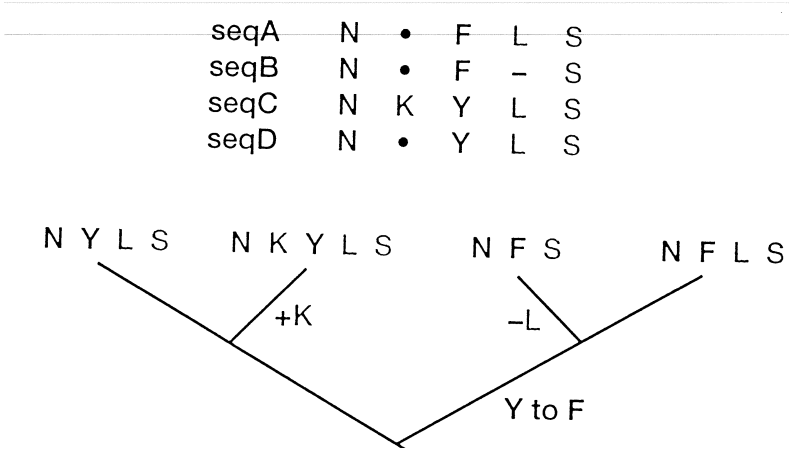
- Entdecken evolutionärer Beziehungen, phylogenetischer Bäume

- Genom-Sequenzierung

- Finden von konservierten Regionen und Domänen, damit:
    - ähnliche Promotoren => gemeinsam regulierte Gene
    - Entdecken von strukturellen Ähnlichkeiten => Funktionelle Ähnlichkeit

Sowohl gleiche, als auch verschiedene Aminosäuren können hilfreich sein:

Konservierte und nicht-konservierte Aminosäuren



- konservierte AS sind wichtig zur Strukturerhaltung des Proteins (in Fig: N,S) derartige AS können die Struktur des Proteins und damit seine Funktion ändern und meistens zerstören, wenn sie mutieren

  => diese AS sind nützlich, um das Alignment zu berechnen!

- weniger konservierte AS beeinflussen Struktur und Funktion in geringerem Maße

  => nützlich, um evolutionäre Verhältnisse abzuleiten!

Bemerkung zur phylogenetischen Analyse

wähle Gene aus, die

- gut konserviert und damit ähnlich in allen zu untersuchenden Organismen sind
- keinem evolutionärem Druck unterlagen
- aber dennoch etwas genetische Variabilität aufweisen
- Beispiel: rRNA

Bemerkung

Multiples-Sequenz-Alignment ist

- leicht, wenn Sequenzen ähnlich,
- schwer, wenn Sequenzen weiter voneinander entfernt sind (viele Möglichkeiten, Gaps zu setzen, Sequenzen gegeneinander zu verschieben,...)

Methoden zum Multiplen-Sequenz-Alignment

M-S-A ist rechentechnisch komplex => Approximierungen nötig:

- Multidimensionales dynamisches Programmieren
  (MSA, Lipman 1988, DCA, Jens Stoye)

- Progressive Alignments
  (Clustalw, Higgins 1996; PileUp, Genetics Computer Group (GCG))

- Lokale Alignments
  (e.g. DiAlign, Morgenstern 1996; viele Andere)

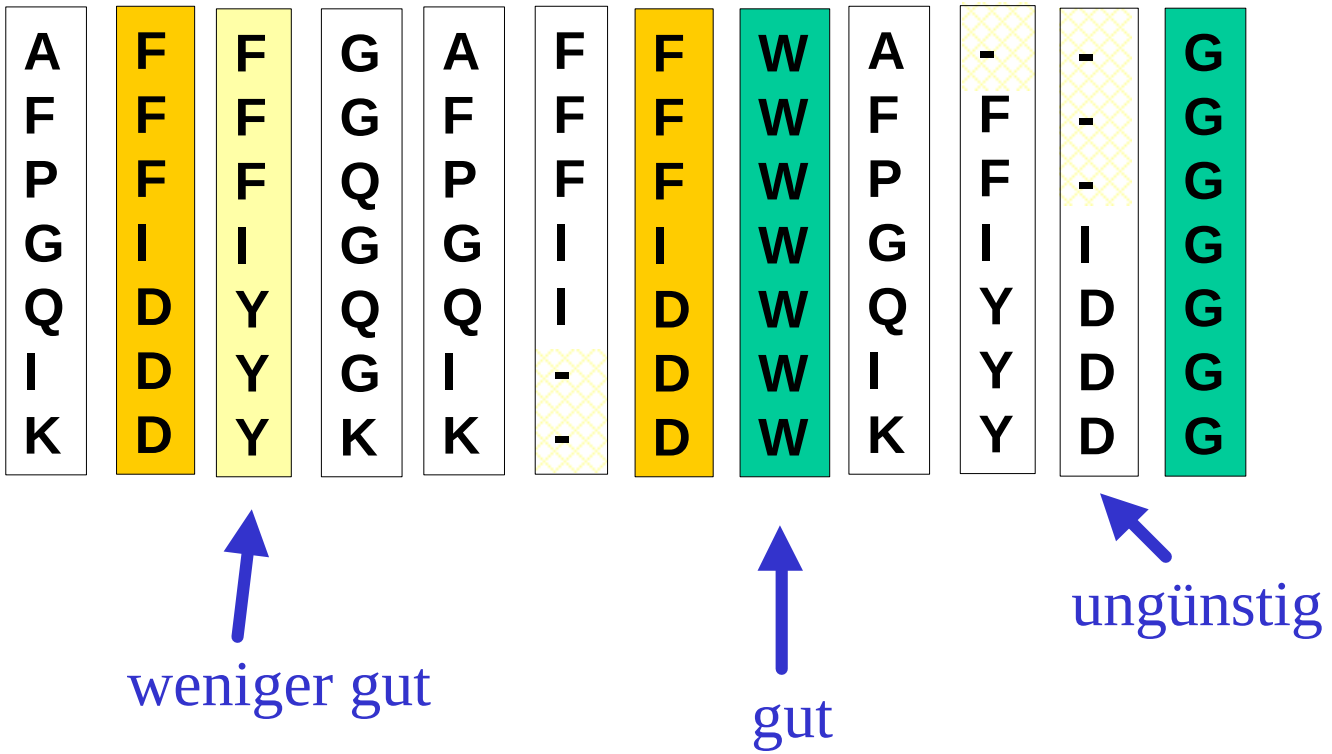
- Iterative Methoden (wird hier nicht behandelt)
  (e.g. PRRP, Gotoh 1996)

- Statistische Methoden (extra Vorlesung über HMM)
  z.B. Bayes'sche Hidden-Markov-Modelle

Scoring

Für alle Methoden wichtig: Bewertung eines

multiplen Alignments, berechnen eines Scorewertes dafür



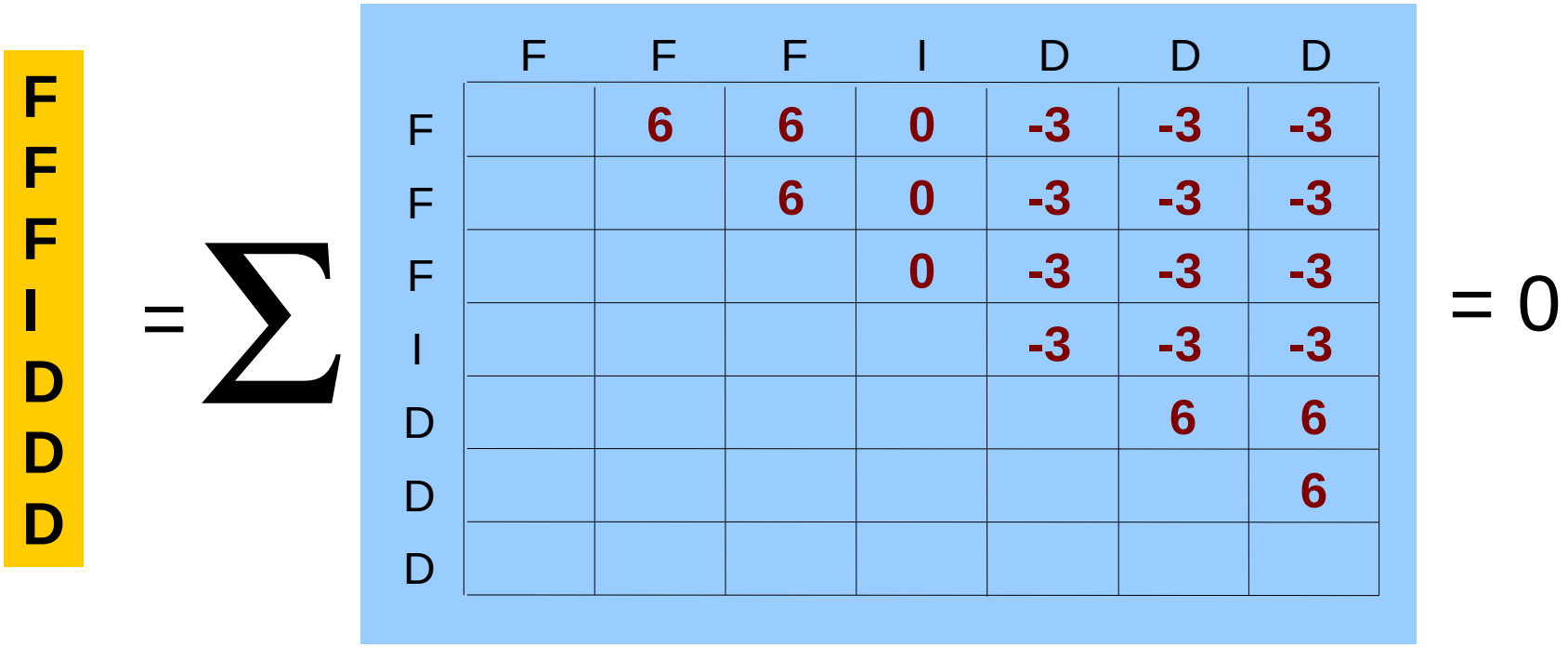
BLOSUM62

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	
C	**9**																				**C**
S	-1	**4**																			**S**
T	-1	1	**5**																		**T**
P	-3	-1	-1	**7**																	**P**
A	0	1	0	-1	**4**																**A**
G	-3	0	-2	-2	0	**6**															**G**
N	-3	1	0	-2	-2	0	**6**														**N**
D	-3	0	-1	-1	-2	-1	1	**6**													**D**
E	-4	0	-1	-1	-1	-2	0	2	**5**												**E**
Q	-3	0	-1	-1	-1	-2	0	0	2	**5**											**Q**
H	-3	-1	-2	-2	-2	-2	1	-1	0	0	**8**										**H**
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	**5**									**R**
K	-3	0	-1	-1	-1	-2	0	-1	1	1	-1	2	**5**								**K**
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	**5**							**M**
I	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	**4**						**I**
L	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	**4**					**L**
V	-1	-2	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	**4**				**V**
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	**6**			**F**
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	**7**		**Y**
W	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	**11**	**W**
	C	**S**	**T**	**P**	**A**	**G**	**N**	**D**	**E**	**Q**	**H**	**R**	**K**	**M**	**I**	**L**	**V**	**F**	**Y**	**W**	

Methode des Sum-of-Pairs (SP) Scoring

= Summe über jedes mögliche Paar in einer Spalte

Beispiel:



F: Phe, I: Iso, D: Asp

Methode des Sum-of-Pairs (SP) Scoring

weiteres Beispiel, besserer Score, weil Tyr näher an Phe als Asp ...

F
F
F
I
Y
Y
Y

$=\sum$

	F	F	F	I	Y	Y	Y
F		6	6	0	3	3	3
F			6	0	3	3	3
F				0	3	3	3
I					-1	-1	-1
Y						7	7
Y							7
Y							

= 63

F: Phe

I: Iso

Y: Tyr

Sum-of-Pairs (SP) Scoring, Formel

FFFIDDD

Die Einträge der Score-Matrizen werden gebraucht (z.B von Blosum62)

eine Spalte:

$$S(m_i) = \sum_{k<l} s(m_i^k, m_i^l)$$

k-te Sequenz, i-te Spalte

Gesamtscore = Summe der Scores aller Spalten:

$$S(m) = \sum_i S(m_i)$$

Problem bei der Sum-of-Pairs-Methode

score = 60 >> score = 24

=> eine einzige falsche AS in einer Spalte zieht den Score schon sehr stark nach unten

=> Lösungen dafür gibt es, sind aber in der Standardsoftware noch nicht implementiert

Methoden zum Multiplen-Sequenz-Alignment

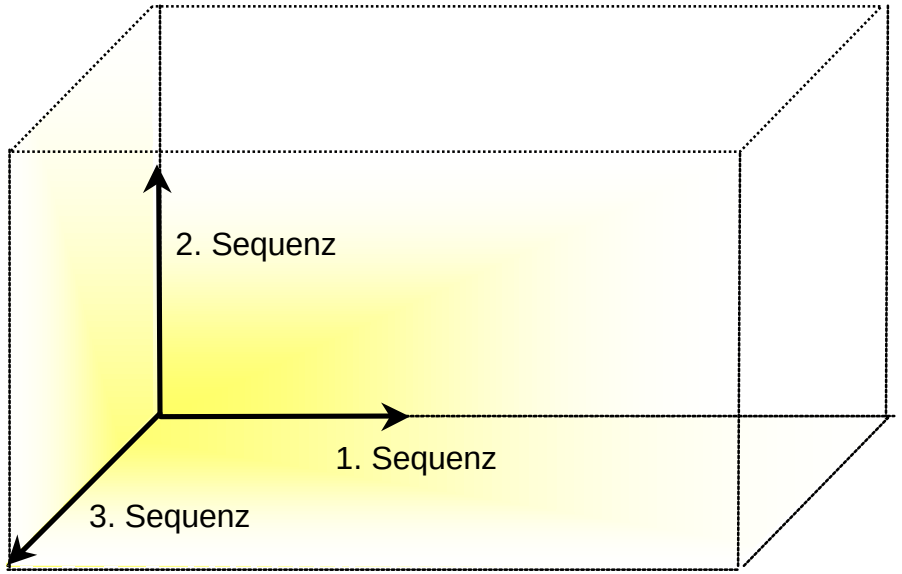
- **Multidimensionales dynamisches Programmieren**
  (MSA, Lipman 1988, DCA, Jens Stoye)

- Progressive Alignments
  (Clustalw, Higgins 1996; PileUp, Genetics Computer Group (GCG))

- Lokale Alignments
  (e.g. DiAlign, Morgenstern 1996; viele Andere)

Multidimensionales dynamisches Programmieren mit drei Sequenzen

Dynamisches Programmieren mit 3 Sequenzen ergibt eine dreidimensionale Alignment-Pfad-Matrix:



aber:

Rechenzeit und

Speicherbedarf!

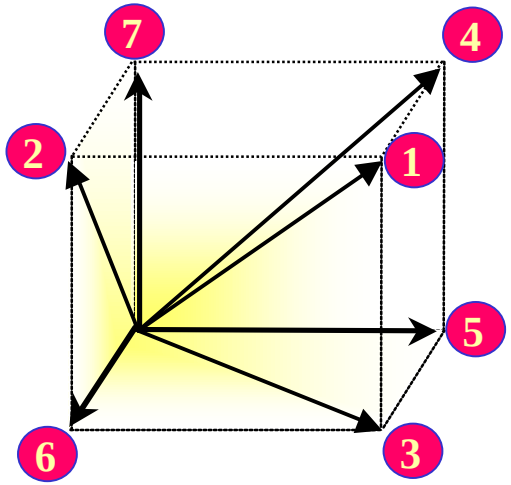
Drei dimensionale Alignment-Matrix

- Bei zwei Sequenzen gibt es immer drei Möglichkeiten, ein Alignment fortzusetzen (vgl. paarweises Alignment von heute morgen):

H	_	H
_	P	P

- Bei 3 Sequenzen gibt es 7 Möglichkeiten:

H	_	H	H	H	_	_
P	P	_	P	_	_	P
E	E	E	_	_	E	_



- Bei N Sequenzen gibt es 2^N-1 Möglichkeiten ...

Hohe Rechenzeit

- Beim vollständigen multidimensionalen dynamischen Programmieren gibt es 2^N-1 Möglichkeiten, ein Alignment forzusetzen, zu verlängern
- Diese Verlängerung wird ja für jede Stelle in den Sequenzen berechnet (also überall in dem $L_1*L_2*L_3$ ....* L_N -großen n-dimensionalen "Würfel"). Wenn alle Sequenzen ungefähr die gleiche Länge haben, kommt man auf ungefähr $L^N(2^N-1)$ zu berechnende Verlängerungsmöglichkeiten

F: Wenn es 1 Sekunde dauert, um 2 Sequenzen mit Länge 50 zu alignieren, wie viele Sequenzen können dann in 5 Mrd. Jahren (= ~10^{17} s.) berechnet werden?

A: => Dreisatz:

$50^2(2^2-1)*C = 1$ s.

$50^N(2^N-1)*C = 10^{17}$ s.

$C = 1 \text{ sec} / 7500 = \sim 10^{-5}$ s.

$(2*50)^N = 10^{17} / 10^{-5}$

$10^{2N} = 10^{22}$

$N = 11$

... also braucht man Näherungen ...

Multiple Alignments

Ansätze dazu:

- Multidimensionales dynamisches Programmieren in einem reduzierten Suchraum

MSA (Lipman, Altschul and Kececioglu, 1989)
(MSA kann ein vernünftiges Alignment von 5-7 Sequenzen mit 200-300AS Länge berechnen)

DCA (Jens Stoye)

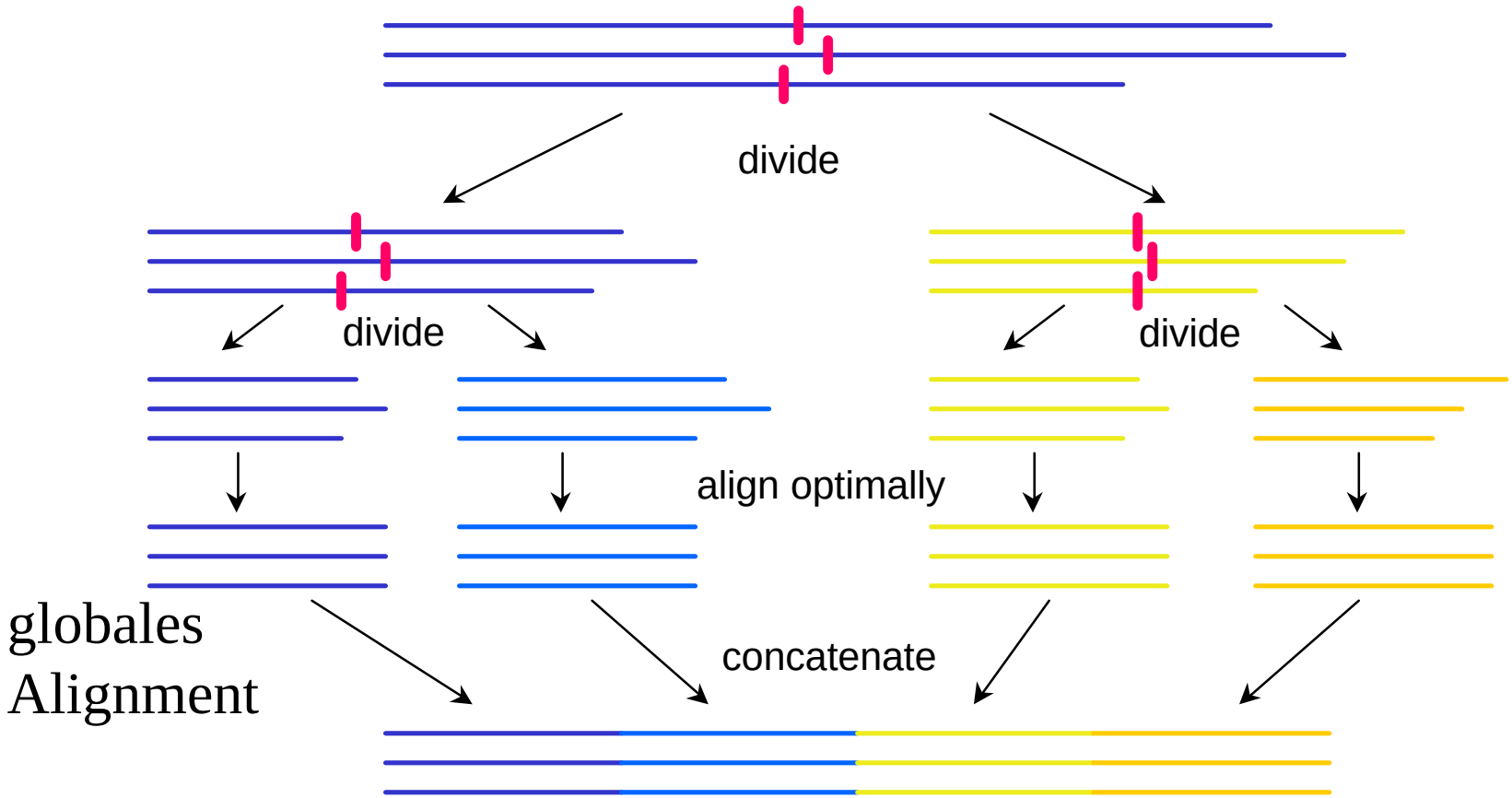
beide Ansätze verkleinern den Suchraum, DCA wird beispielhaft erläutert

Divide-and-Conquer Alignment (DCA)

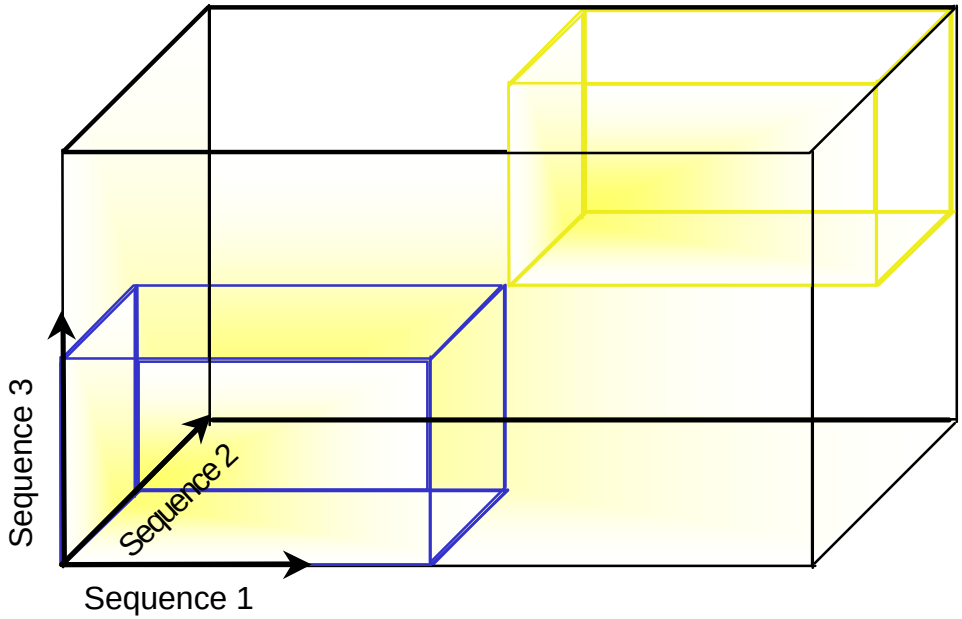
- Verkleinern des Suchraums (=> weniger Rechenzeit):
  - die Sequenzen werden ungefähr in der Mitte zerschnitten, es entstehen 2 Gruppen von Sequenzen kleinerer Länge.
  - diese werden wieder zerschnitten, diese wieder, usw., solange, bis die Sequenzen so klein sind
  - multidimensionales dynamisches Programmieren (alignieren)
  - Die alignierten Sequenzen werden wieder zusammengefügt und der Gesamtalignmentscore (Sum-of-Pairs-Score) berechnet

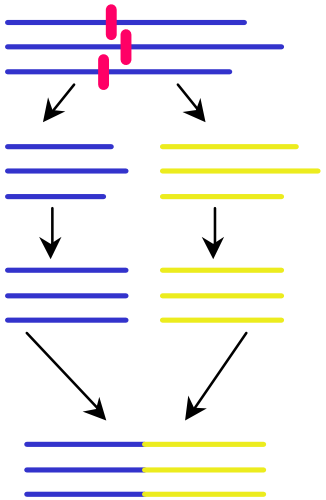
=> entscheidend sind die Zerschneidepunkte...

Divide-and-Conquer Alignment

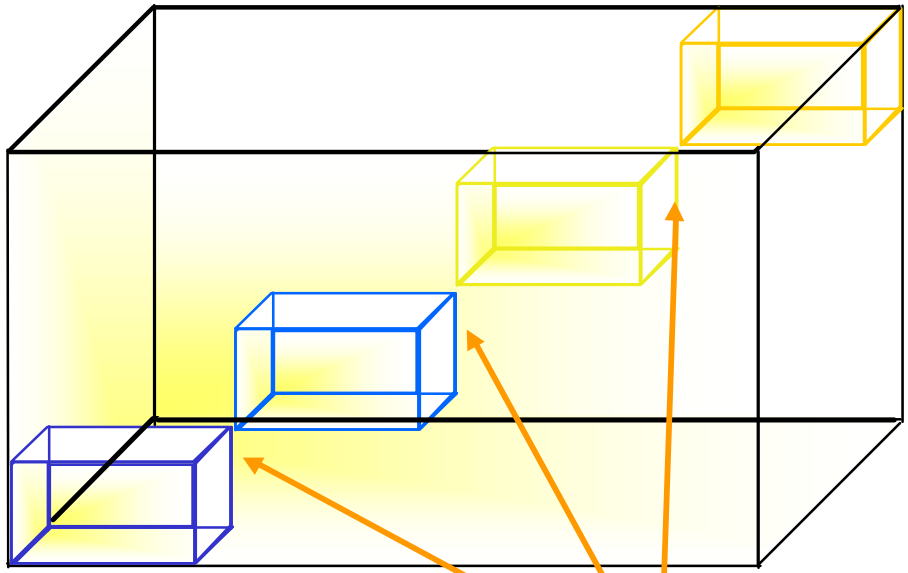


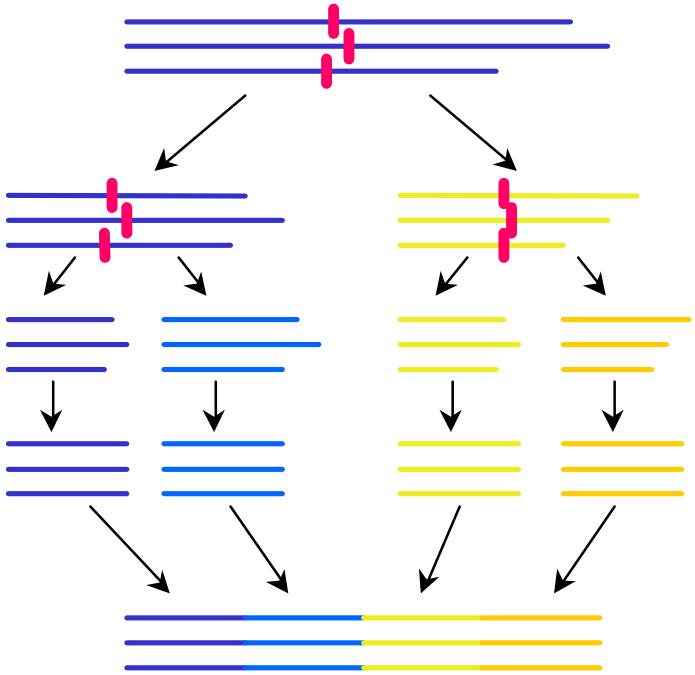
Reduzierung des Suchraums





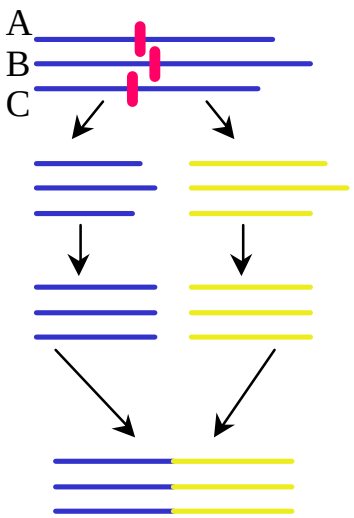
Reduzierung des Suchraums





Zerschneidestellen

Auffinden geeigneter **Schneidestellen**



- Sequenz A wird an c1 zerschnitten
- Sequenz B an c2
- Prefixe und Sufixe werden jeweils paarweise aligniert und der jeweilige Score berechnet (S_{Prefix}, S_{Sufix})
- Score der unzerschnittenen Gesamtsequenz wird berechnet ($S_{Complete}$)

Kostenfunktion eines Paares:

$$\mathbf{C(c1,c2) := S_{Prefix} + S_{Sufix} - S_{Complete}}$$

Kostenfunktion klein => Alignment der Stückchen ≈ Alignment der ganzen Sequenzen

Auffinden geeigneter Schneidestellen

- Die Gesamt-Kostenfunktion ergibt sich zu

$$C(c1,c2,c3) := C(c1,c2) + C(c1,c3) + C(c2,c3)$$

- wird minimiert durch wiederholtes Verändern der Schneidestellen

- wenn die Schneidestellen gefunden sind, führe globales dynamisches multiples Alignment mit den Stückchen durch und füge die multiple-alignierten Vorder- und End-Teile zusammen

Methoden zum Multiplen-Sequenz-Alignment

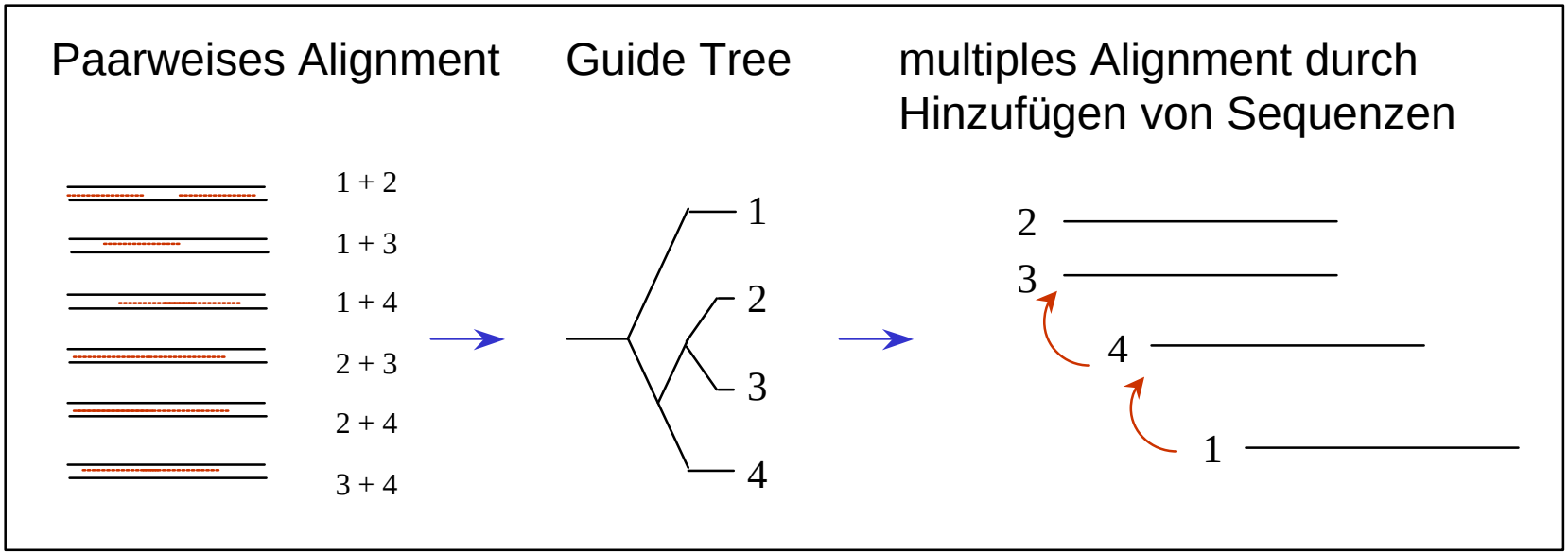
- Multidimensionales dynamisches Programmieren
  (MSA, Lipman 1988, DCA, Jens Stoye)

- **Progressive Alignments**
  (Clustalw, Higgins 1996; PileUp, Genetics Computer Group (GCG))

- Lokale Alignments
  (e.g. DiAlign, Morgenstern 1996; viele Andere)

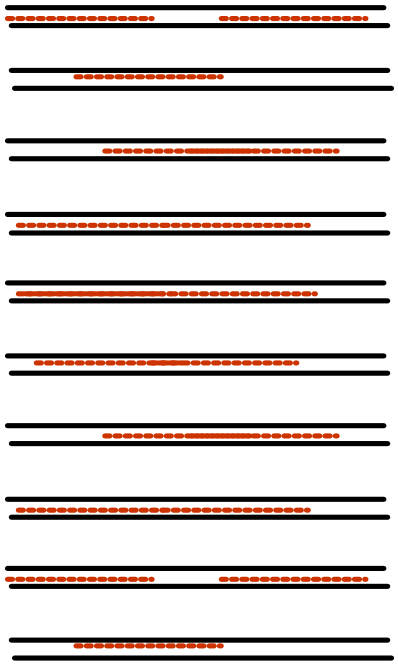
Progressives Alignment (z.B. ClustalW)

Prinzip:



1 → **2** → **3**

Pairweiser Vergleich aller Sequenzen



1 : 2
1 : 3
1 : 4
1 : 5
2 : 3
2 : 4
2 : 5
3 : 4
3 : 5
4 : 5

Ähnlichkeits-
Score von
jedem Paar,
der "**Score**"
(vorherige Vorlesung)

Distanz-Score
von jedem Paar

Berechnen des Distanz-Scores zweier Sequenzen

einfachste Methode: "Hamming Distanz"

Zählen der Mismatches:

```
Seq A = T  A  T  T  C  G
        |        |
Seq B = T  G  C  T  G  T,
```

ergibt Distanz-Score = 4

Berechnen des Distanz-Scores zweier Sequenzen

etwas komplexer: Ähnlichkeistsscores (der "normale" Scorewert) werden in Distanz-Werte umgerechnet, z.B. so (Feng & Doolittle 1996):

normalisieren:

S_{real} = normaler Score-Wert von A und B

S_{ident} = Mittelwert der Scores von A mit sich selbst und B mit sich selbst

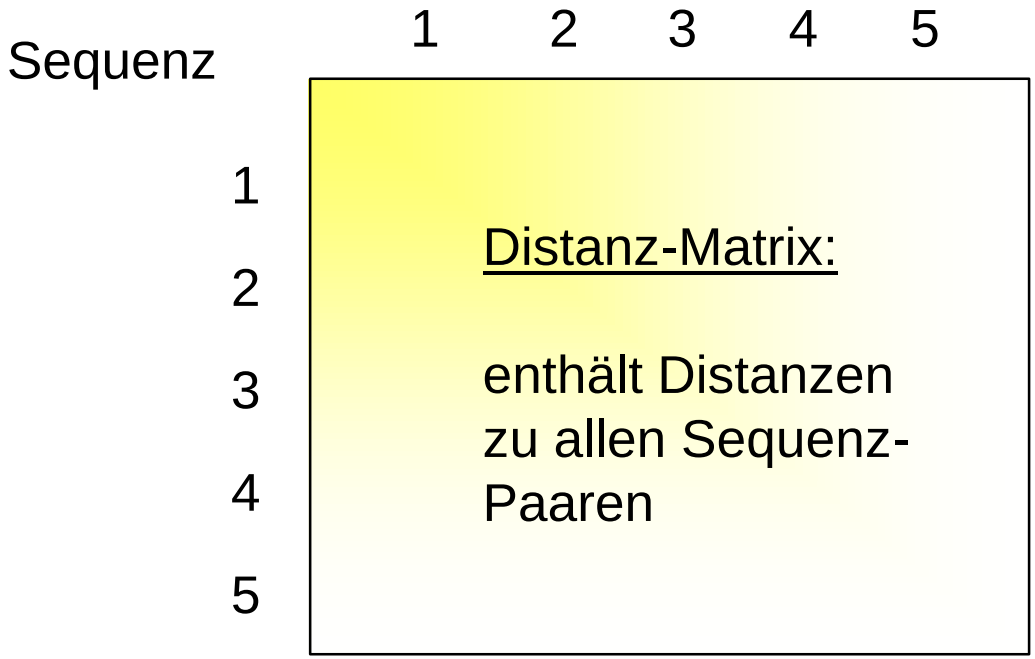
S_{rand} = Mittelwert der Scores von ~1000 geschüttelten Sequenzen A und B

$$\Rightarrow S_{norm} = \frac{S_{real} - S_{rand}}{S_{ident} - S_{rand}} \Rightarrow Distanz_{AB} = -\log S_{norm}$$

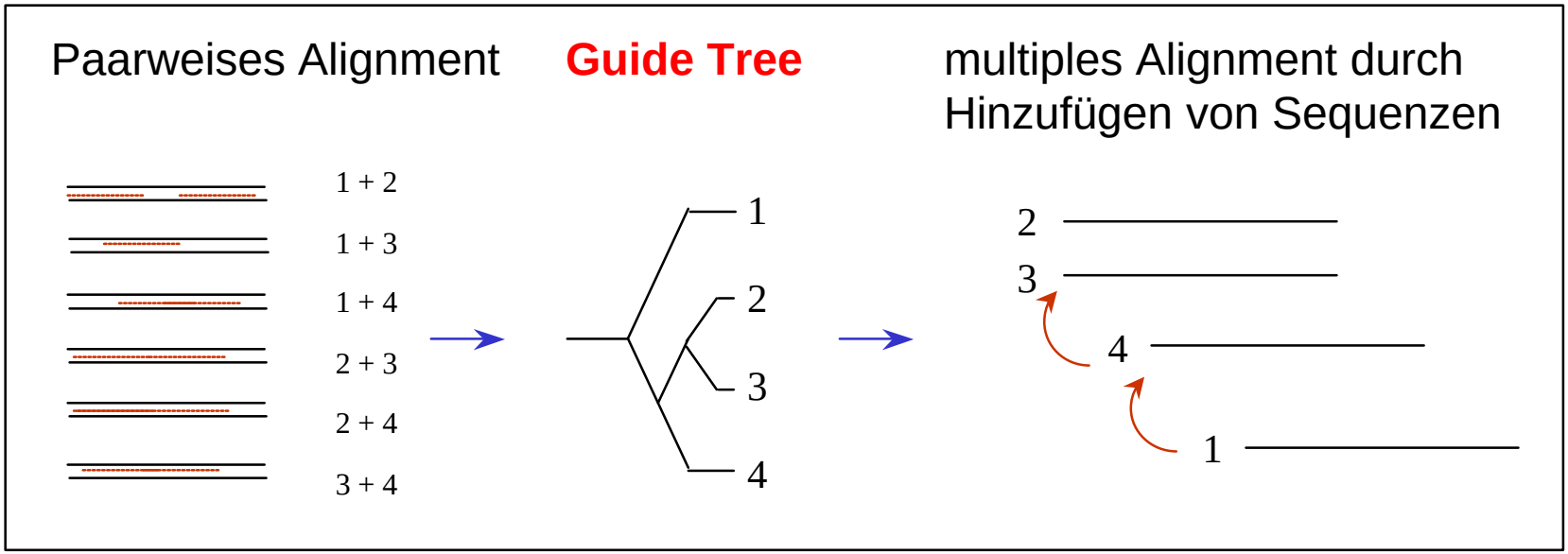
keine Ähnlichkeit: $S_{norm} = 0$ => Distanz = ∞

identisch: $S_{norm} = 1$ => Distanz = 0

Distanz-Matrix



Progressives Alignment



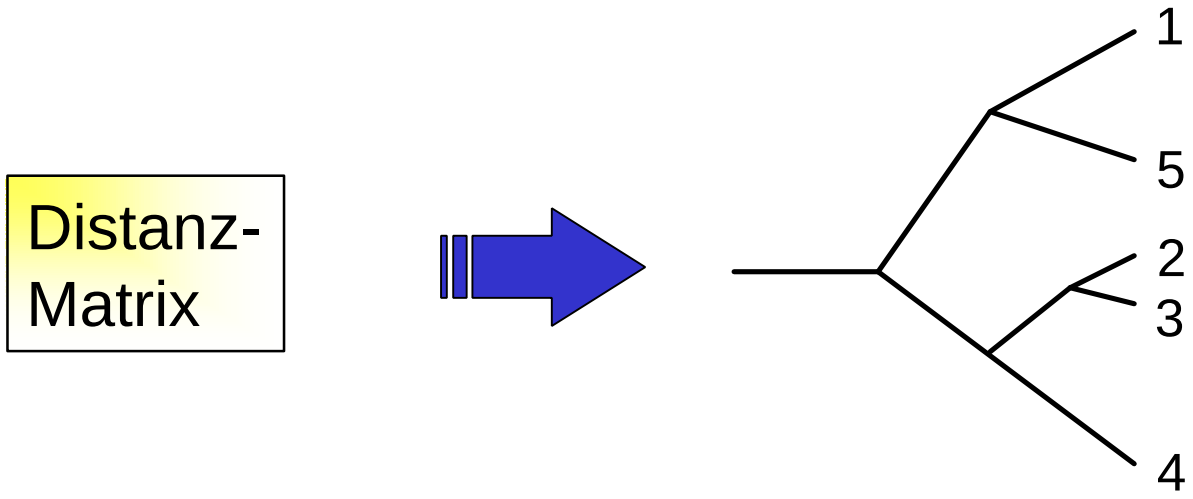
1 → **2** → **3**

Konstruktion des Guide-Trees

kein phylogenetischer Baum !

Guide Tree

Distanz-Matrix



es wird geclustert:

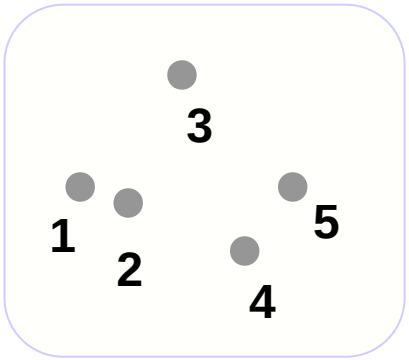
UPGMA (unweighted pair group method of arithmetic averages),

anderes VY: Tyrerfahren: Neighbour-Joining

Hierarchisches Clustern mit UPGMA

unweigthed pairwise group method of arithmetric averages

UPGMA



d_{ij}	1	2	3	4	5
1	0	2	6	9	7
2	2	0	5	7	7
3	6	5	0	5	4
4	9	7	5	0	3
5	7	7	4	3	0

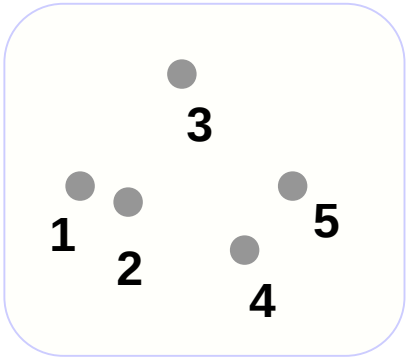
UPGMA

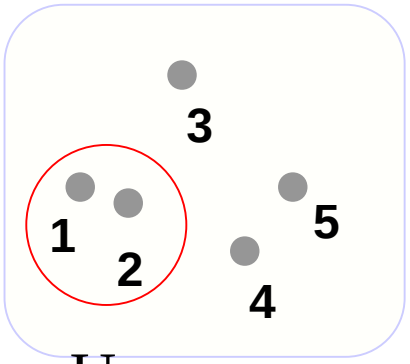
d_{ij}	1	2	3	4	5
1	0	2	6	9	7
2	2	0	5	7	7
3	6	5	0	5	4
4	9	7	5	0	3
5	7	7	4	3	0

d_{ij}	u	3	4	5
u	0	5.5	8	7
3	5.5	0	5	4
4	8	5	0	3
5	7	4	3	0

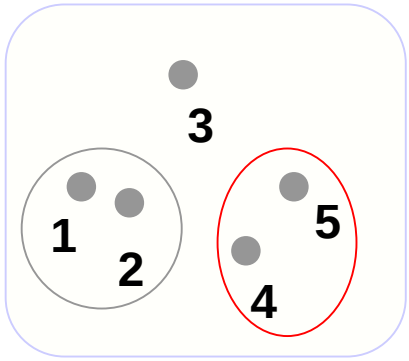
d_{ij}	u	3	v
u	0	5.5	7.5
3	5.5	0	4.5
v	7.5	4.5	0

d_{ij}	u	w
u	0	6.5
w	6.5	0

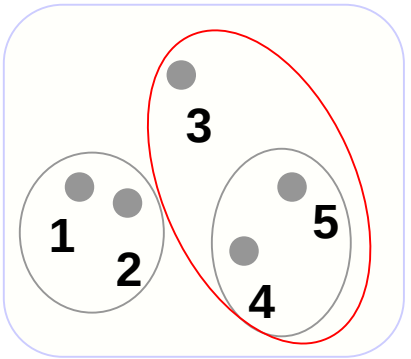




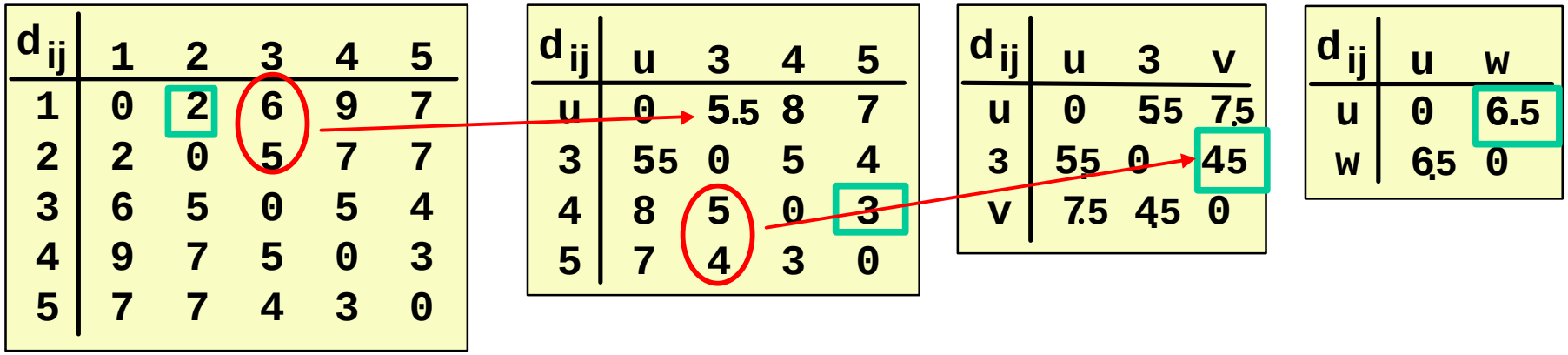
U



V



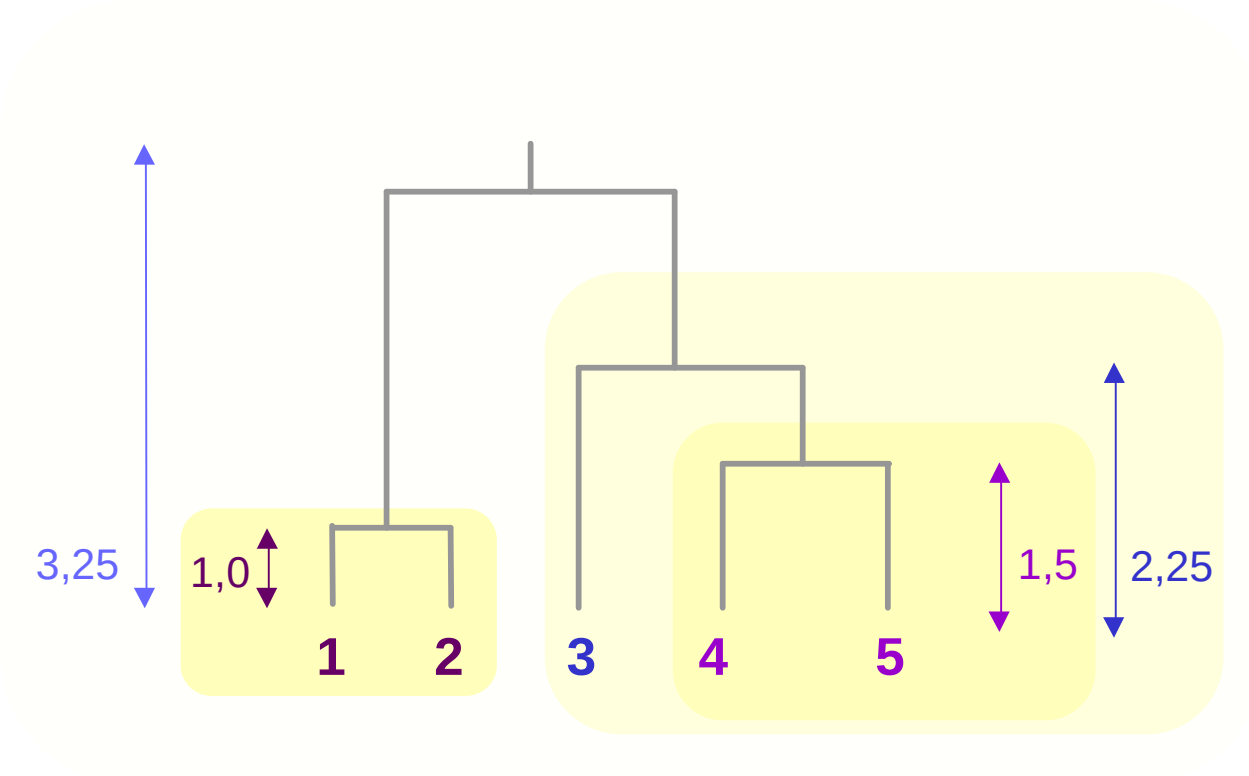
UPGMA



Average linkage clustering:

$$d_{neu,k} = \frac{d_{ik} + d_{jk}}{2}$$

Der entstandene Guide-Tree



Hierarchisches Clustern mit UPGMA

Prinzip:

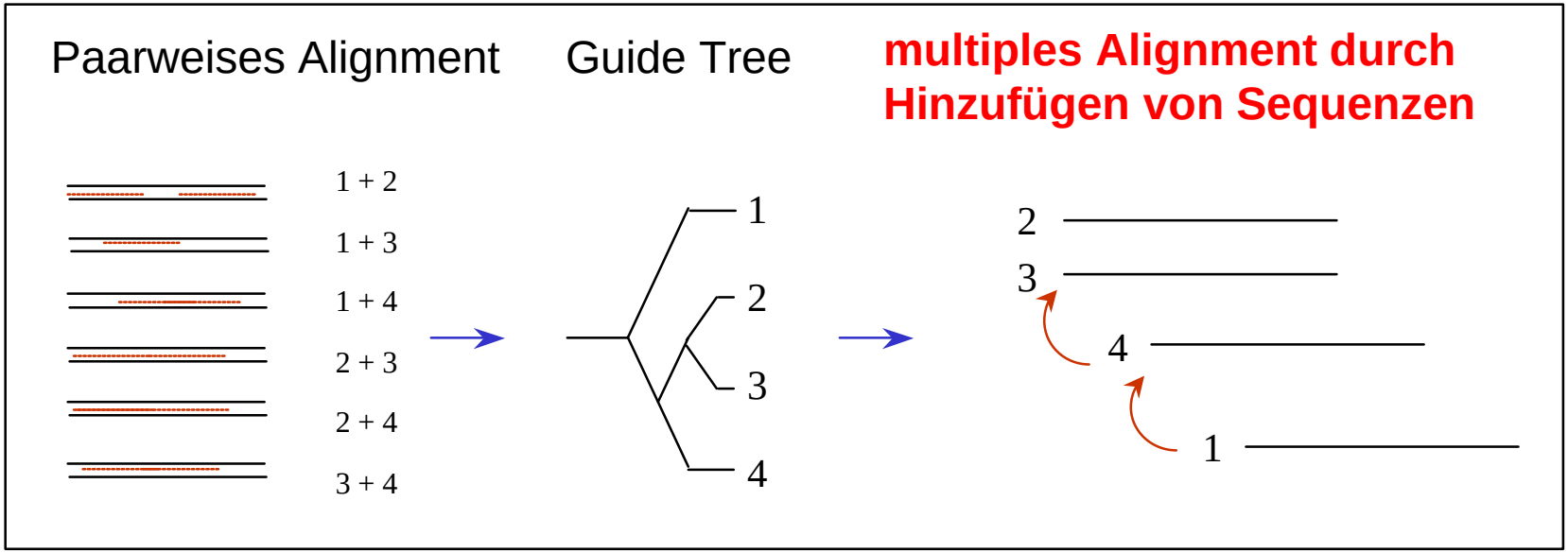
- ermittle kleinsten Wert in der Distanz-Matrix
- Bilde einen Cluster (Gruppe) der entsprechenden Einträge (hier, u = {1,2}).
- berechne den Abstand dieses Clusters zu den restlichen Einträgen, durch Mitteln der Abstände der beiden geclusterten Einträge in dem Cluster zu den restlichen Einträgen

d_{ij}	1	2	3	4	5
1	0	2	6	9	7
2	2	0	5	7	7
3	6	5	0	5	4
4	9	7	5	0	3
5	7	7	4	3	0

d_{ij}	u	3	4	5
u	0	5.5	8	7
3	5.5	0	5	4
4	8	5	0	3
5	7	4	3	0

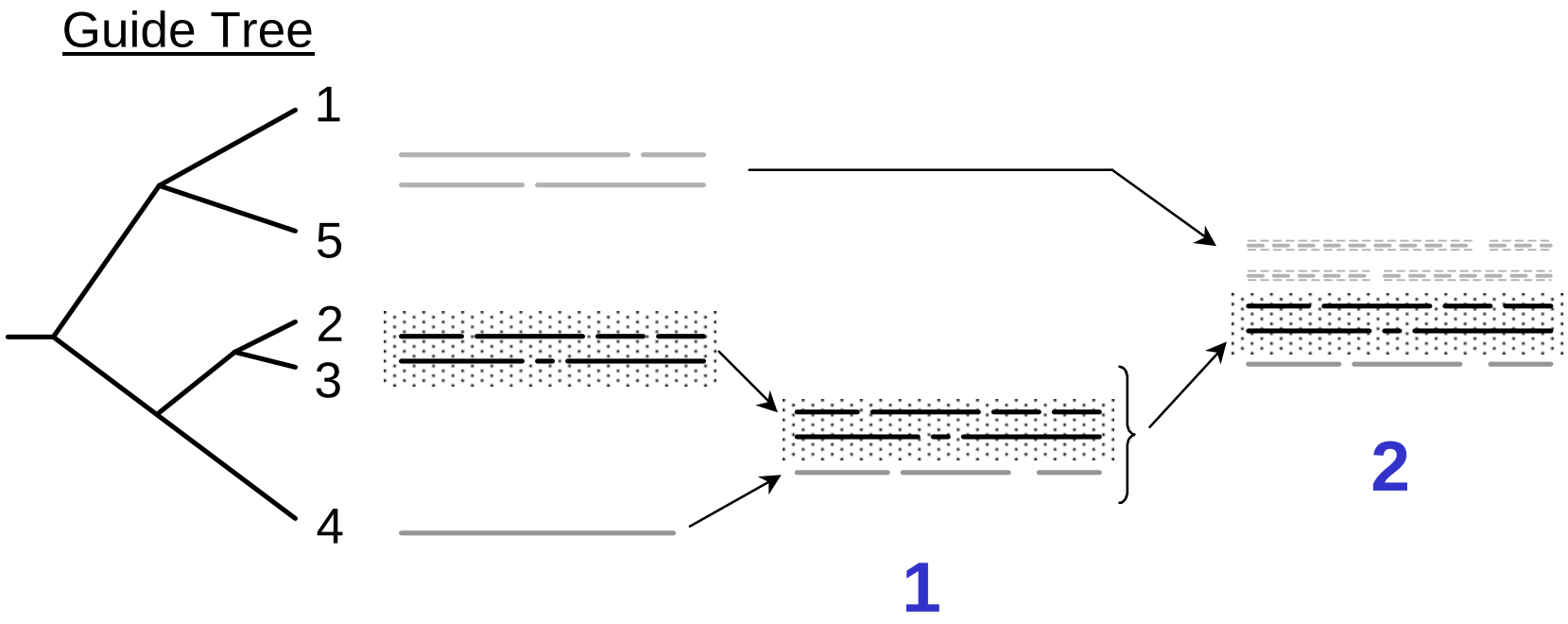
- Wiederhole das solange, bis nur noch zwei Cluster übrig sind

Progressives Alignment



1 → **2** → **3**

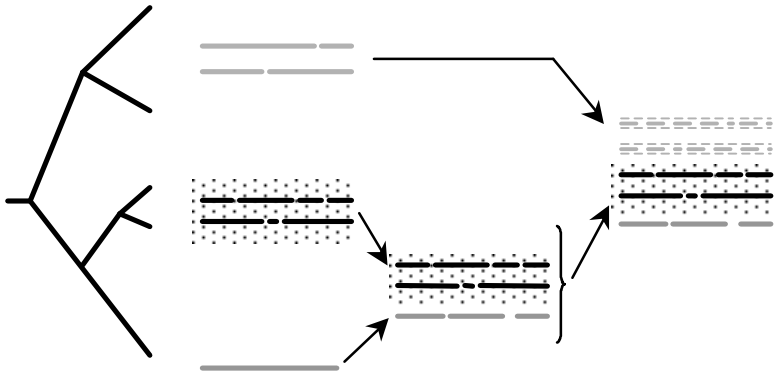
Multiples Alignment



Multiples Alignment

Prinzip:

- Starte mit den 2 Sequenzen, die im Guide-Tree am engsten benachbart sind
- lass sie paarweise global alignieren (z.B. Needleman-Wunsch)
- berechne ein Profil (= "Mischsequenz", s.u.) für diese Sequenzen,
- setze dieses Profil an den gemeinsamen Knoten
- nehme die nächsten zwei Sequenzen oder Profile, die jetzt am engsten zueinander benachbart sind
- berechne ihr Profil
- ende, wenn alle Sequenzen in einem Profil sind (das ist das fertige MSA)



Aber wie bekomme ich ein Profil?

Alignment-Profile

Ein Profil besteht aus einer Liste von Wahrscheinlichkeiten, zu jeder Aminosäure:

F
F
F
I
D
D
D

Profil dieser Spalte:

p_A= 0.00, p_C= 0.00, p_D= 0.43, p_E= 0.00,
p_F= 0.43, p_G= 0.00, p_H= 0.00, p_I= 0.14,
p_K= 0.00,...,p_Y= 0.00

0.00
0.00
0.43
0.00
0.43
0.00
0.00
0.14
0.00
0.00
0.00
0.00
0.00
0.00
0.00
0.00
0.00
0.00
0.00
0.00

Alignment-Profile

Wie ergeben sich diese Wahrscheinlichkeiten:
p_{AS} ~ Vorkommen in der Spalte

für das gesamte Profil ergibt sich dann die folgende Formel:

F
F
F
I
D
D
D

N = 7

$c_F = 3$

$c_I = 1$

$c_D = 3$

$$P_i(a) = \frac{c_{ia}}{\sum_{a'} c_{ia'}} = \frac{c_{ia}}{N}$$

Wahrscheinlichkeit für Aminosäure a in der i-ten Spalte

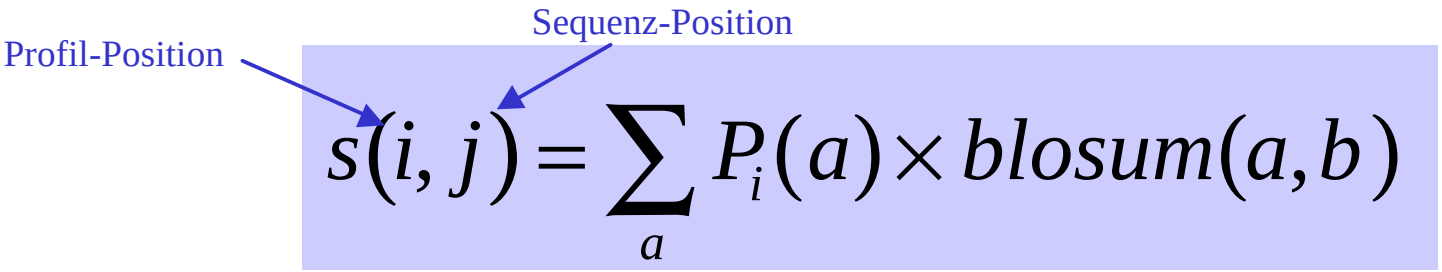
Anzahl der Zeilen

Anzahl der Aminosäure a in der i-ten Spalte

Wie wird eine Sequenz mit einem Profil aligniert?

=> dynamisches Programmieren mit Sequenz und Profil ("Mischsequenz")

(1) Berechne Score-Matrix s(i,j) für diese Sequenz mit diesem Profil:



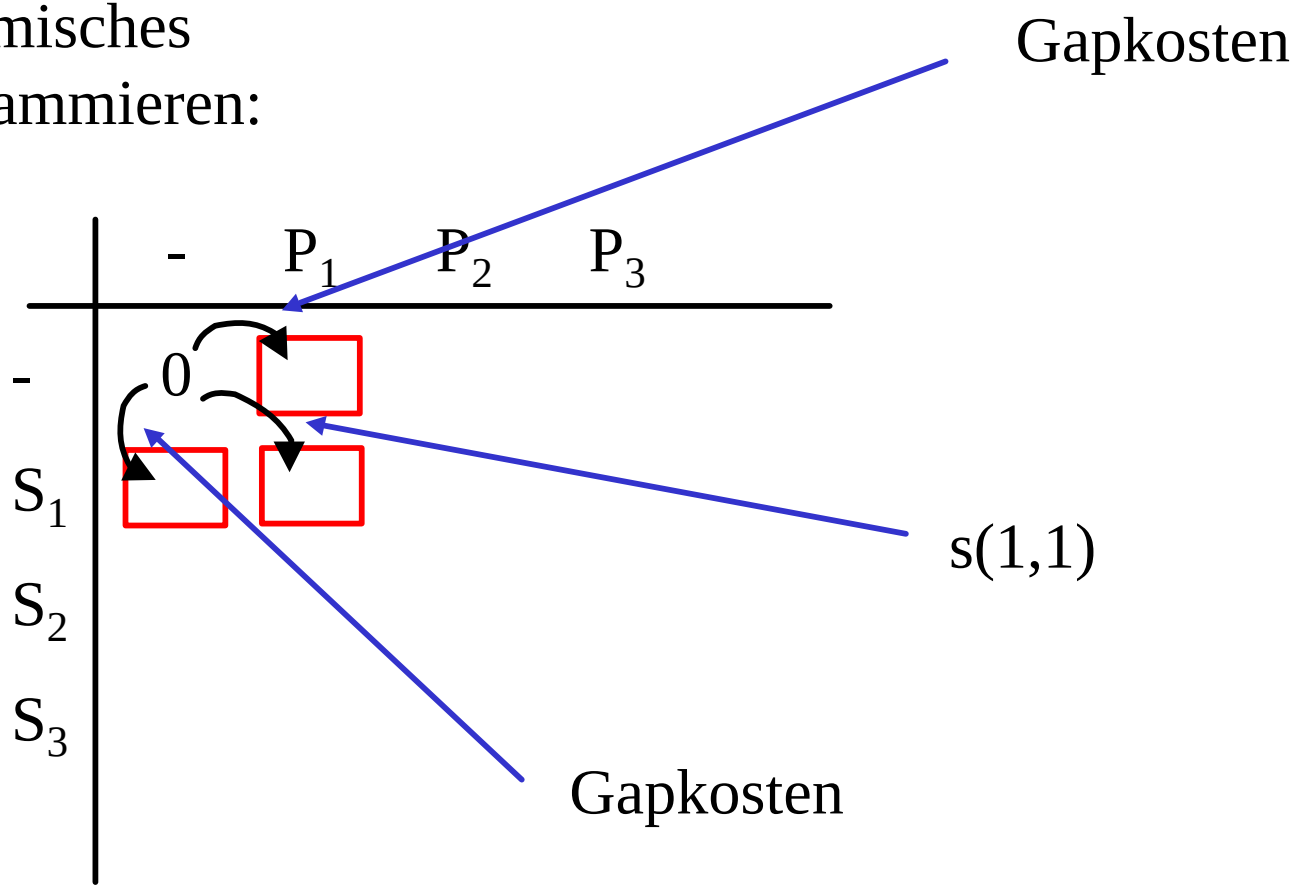
$P_i(a)$ ist die Wahrscheinlichkeit, dass Aminosäure a in der i-ten Spalte auftaucht (im Profil).

b ist die Aminosäure, die in der Sequenz an der j-ten Stelle kommt

(2) dynamisches Programmieren, Sequenz gegen Mischsequenz, mit Score-Matrix aus (1) und normalen Gapkosten

Wie wird eine Sequenz mit einem Profil aligniert?

Dynamisches Programmieren:



Wie wird ein Profil mit einem Profil aligniert?

=> dynamisches Programmieren mit zwei Profilen

(1) berechnen der Score-Matrix s(i,j):

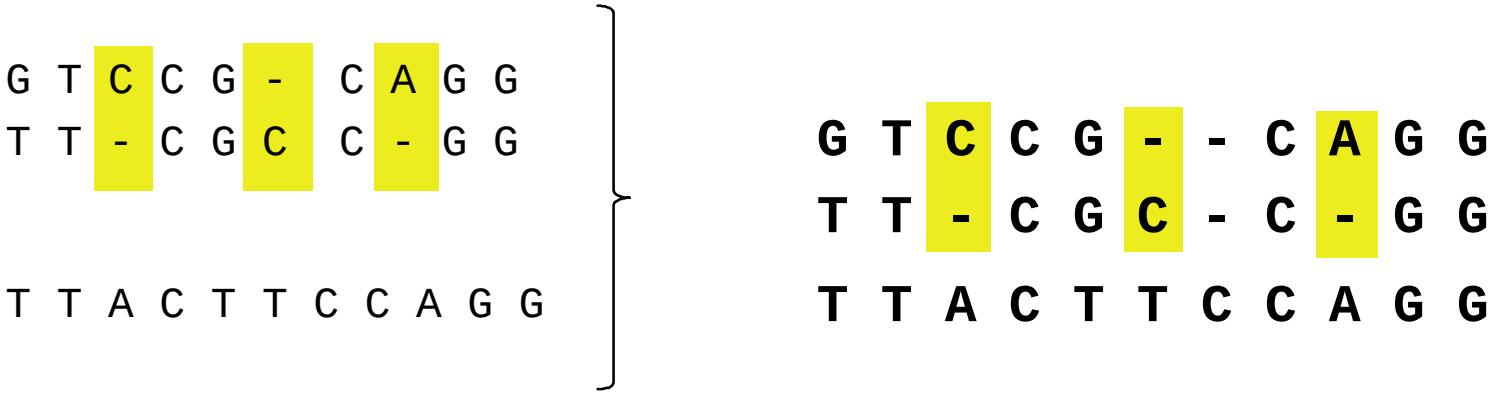
$$s(i,j) = \sum_a \left[P_i(a) \sum_b \left[P_j(b) \times blosum(a,b) \right] \right]$$

$P_i(a)$ ist die Wahrscheinlichkeit, dass Aminosäure a in der i-ten Spalte im 2. Profil auftaucht,

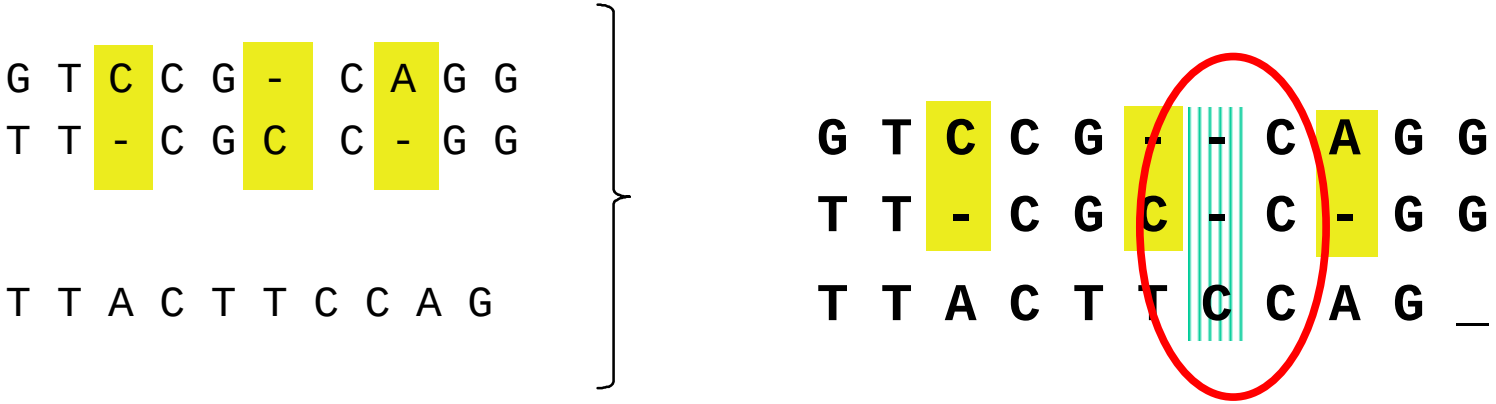
$P_j(b)$ ist die Wahrscheinlichkeit, dass Aminosäure b in der j-ten Spalte im 2. Profil auftaucht

(2) dynamisches Programmieren mit dieser Score-Matrix

Spalten: einmal aligniert => nie mehr geändert

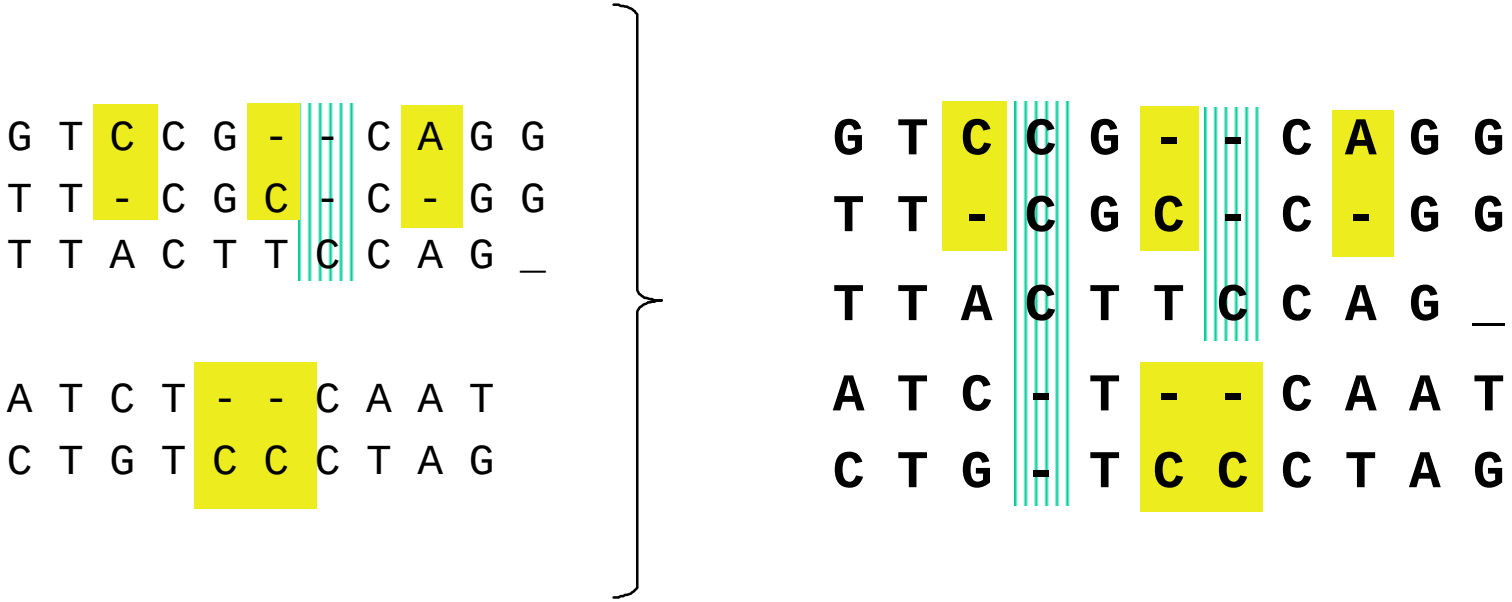


Spalten: einmal aligniert => nie mehr geändert



. . . . und immer wieder neue Gaps dazu ...

Spalten: einmal aligniert => nie mehr geändert



Gap-Kosten

• Problem: wenn man Gaps wie im paarweisen Alignment behandeln würde, würden es zuviele!

=> Lösung in Clustalw (Thompson, Higgins & Gibson 1994):

1. Sowohl Gap-Öffnen, als auch Gap-Verlängern verteuert sich, wenn in der Spalte selbst keine Gaps sind, dafür aber in den Spalten daneben => zwingt Gaps in den gleichen Spalten aufzutreten
2. Gap-Kosten werden mit einem Modifizierer multipliziert, z.B. ergeben sich damit höhere Kosten in Spalten mit hydrophoben Residuen (im Protein geborgen, dürfen sich nicht groß ändern), als in Spalten mit hydrophilen oder flexiblen Residuen.
3. Gap-Öffnen-Kosten sind an Stellen reduziert, die von fünf oder mehr hydrophilen Residuen umgeben sind
4. Gaps treten bevorzugt neben einigen bestimmten Aminosäuren auf, begünstige diese Gaps

Bemerkung. 3. und 4. hängt mit der Beobachtung zusammen, dass Gaps vermehrt zwischen Sekundärstruktur auftritt

Sequenz- Gewichtung

- Homologe Sequenzen sind nicht unabhängig. Sie stammen in unterschiedlichem Maße von gleichen Vorfahren oder voneinander ab.

  => einige Paare sind stärker miteinander verwandt als andere

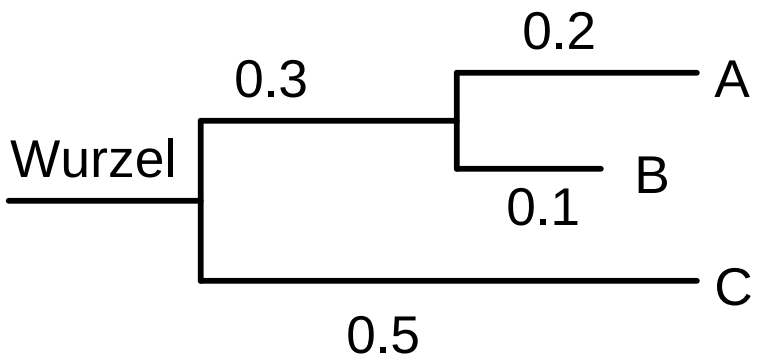
- Hat man viele stark miteinander verwandte Sequenzen und ein paar, die weniger mit diesen verwandt sind, kann ein Ungleichgewicht beim Erstellen des Profils entstehen

  => vermindere den Einfluss der stark Verwandten!

Sequenz- Gewichtung

Beispiel für 3 Sequenzen A, B, C:

- ungewichtet: für alle Sequenzen ist Gewicht w gleich (z.B 1 oder 1/3)
- gewichtet:



$w_A = 0.2 + 0.3/2 = 0.35$

$w_B = 0.1 + 0.3/2 = 0.25$

$w_C = 0.5$

w = Distanz von der Wurzel, die sich geteilt wird, wenn Nachbarn auftauchen

-nach Normalisieren (Summe = 1): **$w_A = 0.33$**
$w_B = 0.22$
$w_C = 0.45$

$w_A = 0.33$
anstatt $w_B = 0.33$
$w_C = 0.33$

Einige Bemerkunegn zur Software

Clustalw

Clustal: 1988, Higgins & Sharp

Clustalw: verbesserte Version, Sequenzen können gewichtet werden

Clustalx: wie Clustalw, enthält zusätzlich grafische Oberfläche

paarweises Alignment mit dynamischem Programmieren (verbesserte Mmethod, von Myers & Miller 1988)

Guide-Tree: Neighbour-Joining

Pileup

im GCG Paket enthalten

paarweises Alignment: Needleman-Wunsch

Guide-Tree: UPGMA, Average-Linkage

Probleme mit Methoden des progressiven Alignments

- hängen stark vom Erfolg des paarweisen Alignments
  und der startenden zwei Sequenzen ab

=> brauchen zwei nah verwandten Sequenzen zum Start (sehr verwandt)

=**> alle Sequenze-Paare müssen paarweise alignierbar sein!! (verwandt)**

- wenn das nicht der Fall ist: versuche lokale Alignment-Methoden oder statistische Ansätze (z.B. HMM)

Methoden zum Multiplen-Sequenz-Alignment

- Multidimensionales dynamisches Programmieren
  (MSA, Lipman 1988, DCA, Jens Stoye)

- Progressive Alignments
  (Clustalw, Higgins 1996; PileUp, Genetics Computer Group (GCG))

- **Lokale Alignments**
  (e.g. DiAlign, Morgenstern 1996; viele Andere)

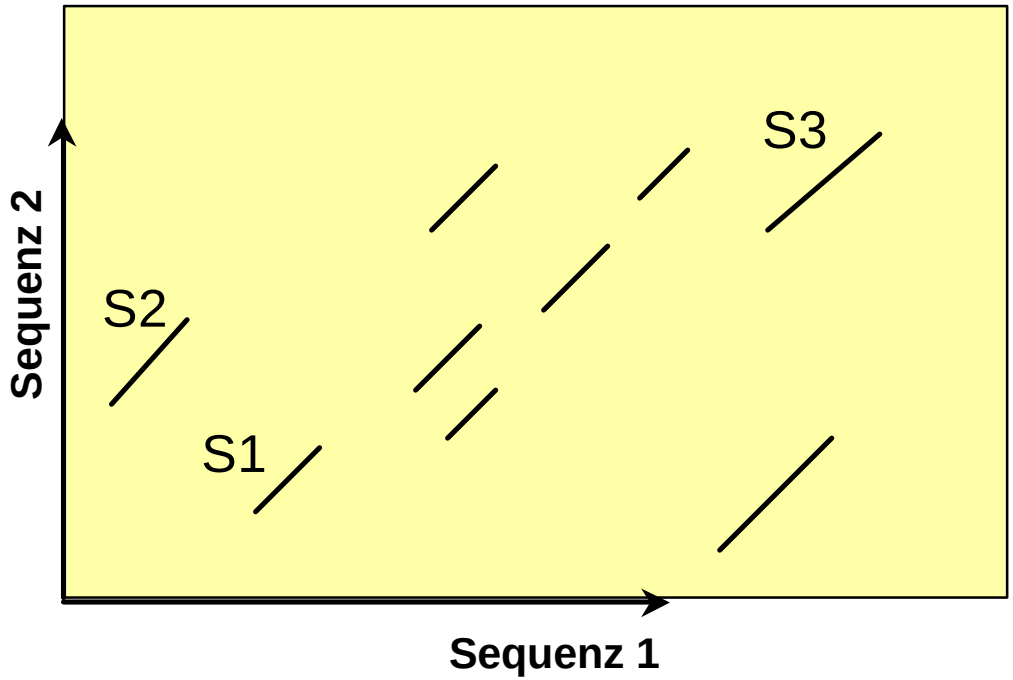
DiAlign
(Diagonal Alignment, von B. Morgenstern)

≈ Erweiterung von Dotplot auf mehrere Sequenzen

- Lokales Alignment
- Segment-zu-Segment Vergleich ohne Gaps
  - also keine Vergleiche einzelner Residuen
- Gaps werden nicht explizit betrachtet (auch keine Gap-Kosten)
  - Gaps representieren die Teile in den Sequenzen,

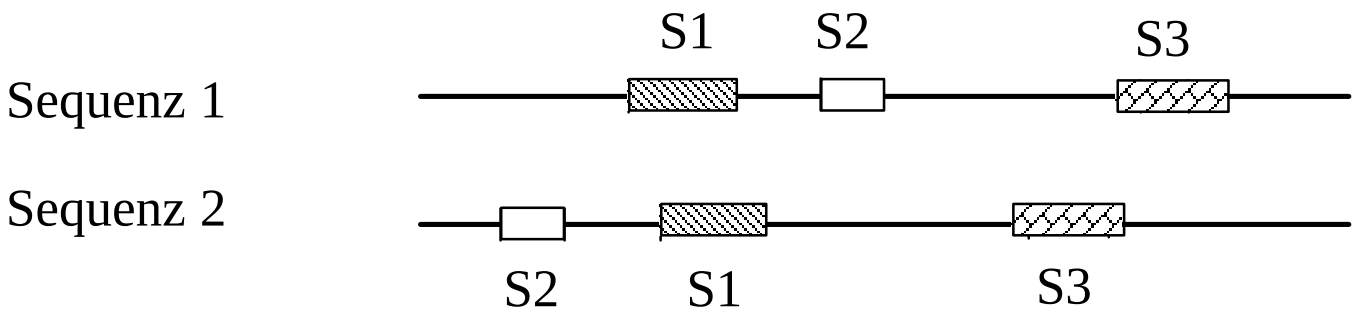
    die nicht aligniert werden

Beispiel: Alignment zweier Sequenzen



Begriffserläuterung:

Konsistente und Nicht-konsistente Diagonalen (in lokalen Alignments)



konsistent:	S1 + S3, S2 + S3
nicht-konsistent:	S1 + S2

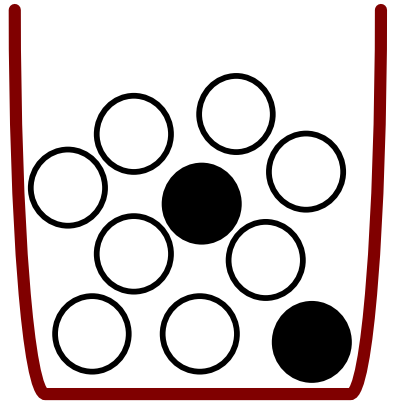
Einschub: Binomialverteilung

$$P(x=k) = \binom{n}{k} p_x^{\,k} (1-p_x)^{n-k}$$

P(x=k): Wahrscheinlichkeit, k Kugeln der Farbe x zu ziehen

p_x : Anteil der x-farbenen Kugeln im Topf

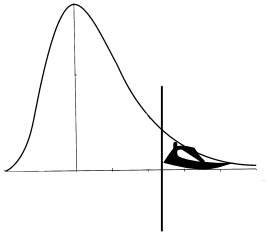
n: Stichprobenumfang, (wie oft wird gezogen)



$p_{schwarz} = 0.25$

Wahrscheinlichkeit eine Diagonale der Länge l mit ≥ m Matches zu bekommen

$$P(l,m)=\sum_{i=m}^{l}\binom{l}{i}p^{i}(1-p)^{l-i}$$



p : Wahrscheinlichkeit eines einzelnen Punktes in der Punkt-Matrix, einen Match zu repräsentieren

bei gleichmäßiger Verteilung: *p*=0,25 (DNA)

p=0,05 (Protein)

Scorewert einer Diagonalen

Negativer Logarithmus: $E(l, m) := -\lg(P(l, m))$

$$w(D) := \begin{cases} E(l, m), \text{ wenn } E(l, m) > T, \\ 0 \quad \text{sonst} \end{cases}$$

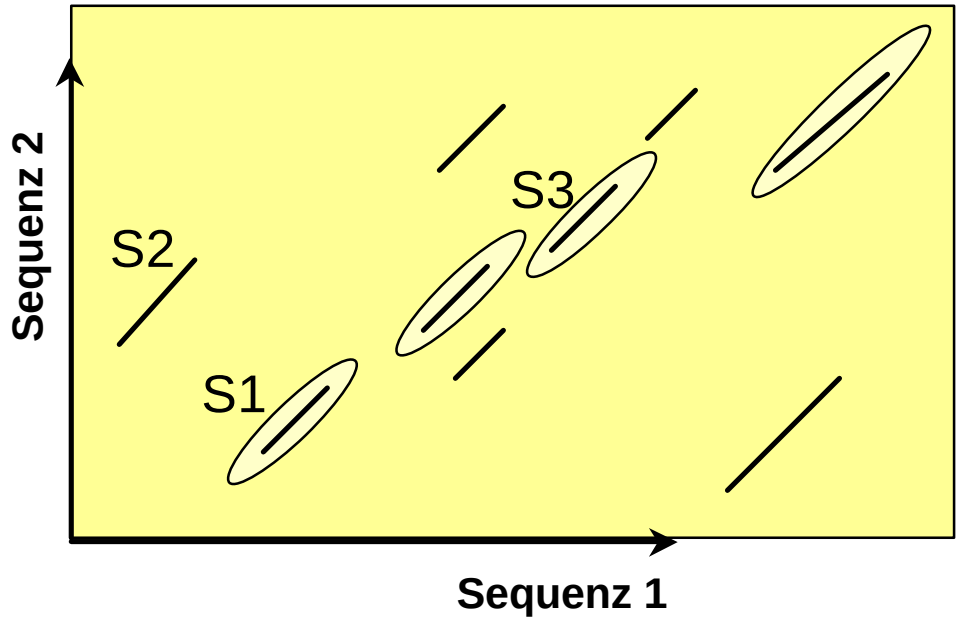
- T: Benutzer-definierter Schwellenwert (reduziert Rauschen)
- Gewicht w hoch => Diagonale selten, signifikant

Maximales Alignment

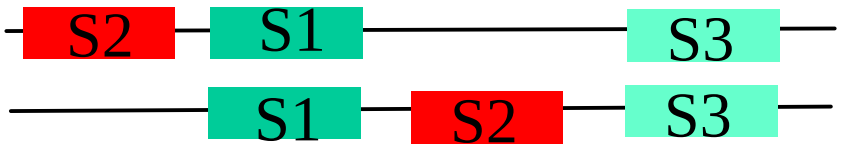
• Maximales Alignment := konsistente Sammlung von Diagonalen die die maximale Summe der Gewichte ergeben

Maximales Alignment

brauchbare Scores:

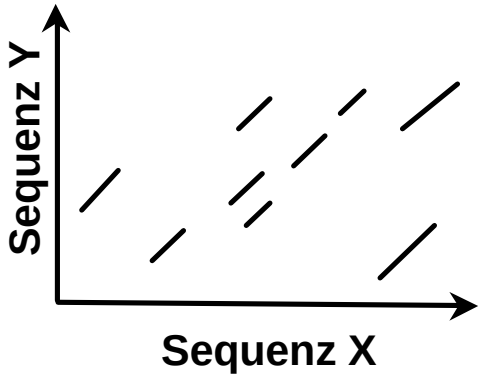


- Markierte Diagonale werden für das maximale Alignment genommen
- S2: nicht konsistent

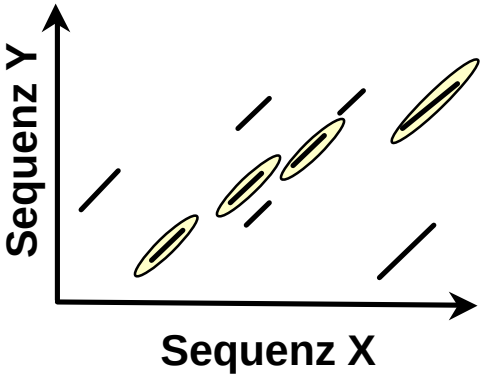


Schrittweises Vorgehen von Dialign

1. Paarweises Auftragen jedes Sequenz-Paares



2. Maximales Alignment (bestimmen der Diagonalen für jedes Sequenz-Paar)

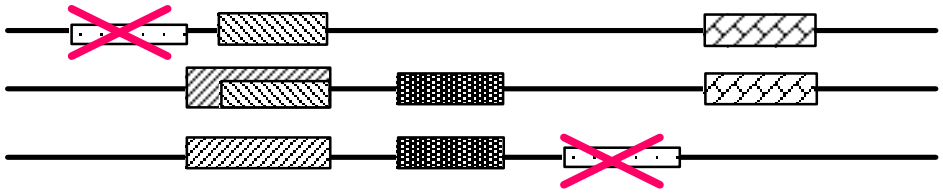


Schrittweises Vorgehen von Dialign

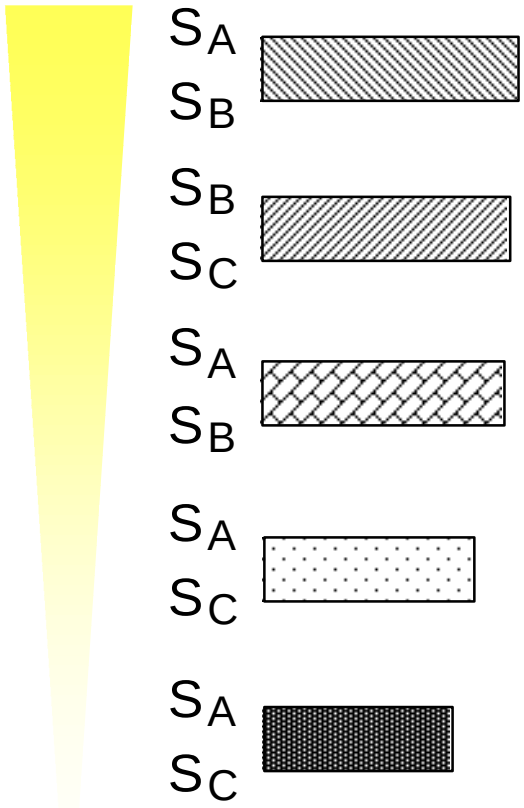
3. Die Diagonalen **<u>aller</u>** paarweisen maximalen Alignments werden nach ihrem maximalen Alignment-Score angeordnet und nacheinander in das multiple Alignment eingefügt,

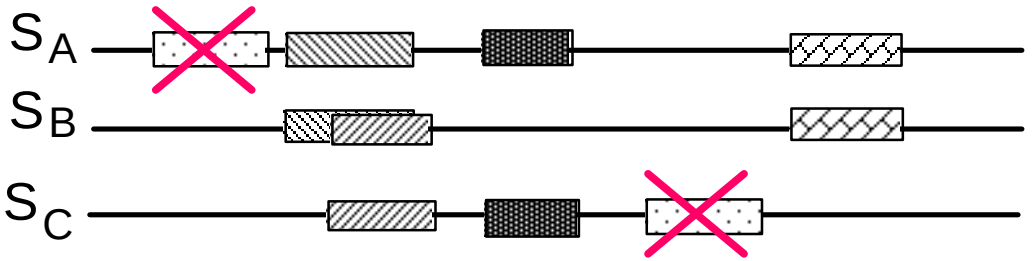
solange sie konsistent mit dem wachsenden multiplen Alignment sind

(nicht-konsistente Diagonalen werden wieder entnommen)



Die Erweiterung zum multiplen Alignment





Programme

Name	Source	Reference
Global alignments including progressive		
CLUSTALW or CLUSTALX (latter has graphical interface)	FTP to ftp.ebi.ac.uk/pub/software[a,d]	Thompson et al. (1994a, 1997); Higgin et al. (1996)
MSA	http://www.psc.edu/[b]	Lipman et al. (1989);
	http://www.ibc.wustl.edu/ibc/msa.html[c]	Gupta et al. (1995)
	FTP to fastlink.nih.gov/pub/msa	
PRALINE	http://mathbio.nimr.mrc.ac.uk/~jhering/praline	Heringa (1999)
Iterative and other methods		
DIALIGN segment alignment	http://www.gsf.de/biodv/dialign.html	Morgenstern et al. (1996)
MultAlin	http://protein.toulouse.inra.fr/multalin.html	Corpet (1988)
PRRP progressive global alignment (randomly or doubly nested)	ftp.genome.ad.jp/pub/genome/saitama-cc	Gotoh (1996)
SAGA genetic algorithm	http://igs-server.cnrs-mrs.fr/~cnotred/Projects_home_page/saga_home_page.html	Notredame and Higgins (1996)