

Algorithmische Grundlagen des Sequenz – Alignments Teil 2b

24.01.05

Vorlesung Bioinformatik 1

Molekulare Biotechnologie

Prof. Dr. Roland Eils

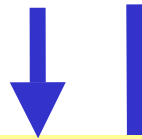
Besonderen Dank an Dr. R. König, Dr. M. van der Linden, M.

Falkenhahn und der Husar Biocomputing Service Gruppe für die

Unterstützung bei der Erstellung der Folien

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

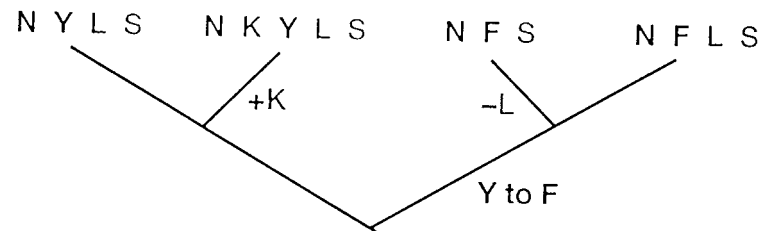
Motivation

- 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

seqA	N	•	F	L	S
seqB	N	•	F	-	S
seqC	N	K	Y	L	S
seqD	N	•	Y	L	S



- konservierte AS sind wichtig zur Strukturhaltung 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

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
F
F
I
D
D
D

= Σ

	F	F	F	I	D	D	D
F		6	6	0	-3	-3	-3
F			6	0	-3	-3	-3
F				0	-3	-3	-3
I					-3	-3	-3
D						6	6
D							6
D							

= 0

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

= Σ

	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

F
F
F
I
D
D
D

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

N
N
N
N
N

score = 60

>>

N
N
N
N
L

score = 24

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

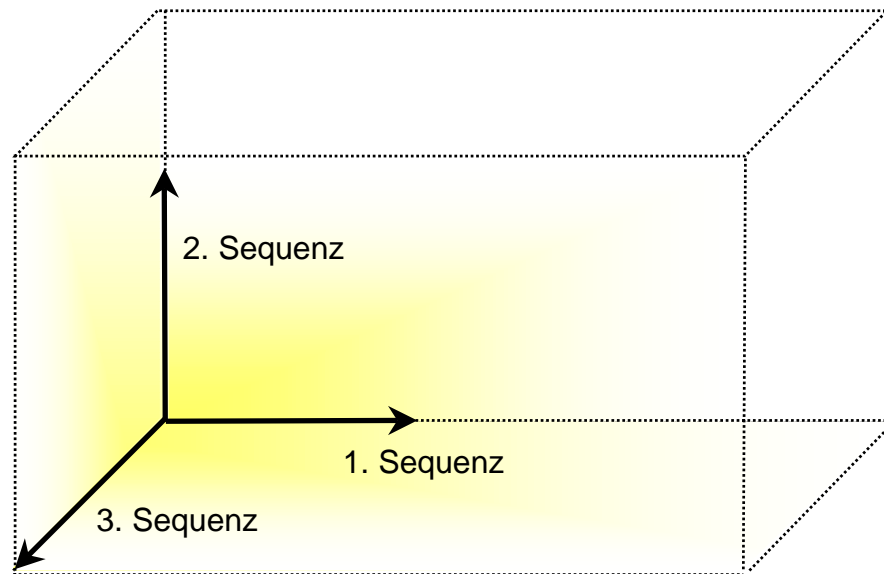
=> Lösungen dafür gibt es, sind aber in der Standardsoftware noch nicht implementiert

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

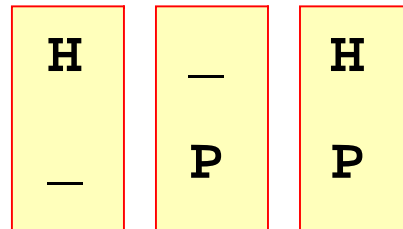
Dynamisches Programmieren mit 3 Sequenzen ergibt eine dreidimensionale Alignment-Pfad-Matrix:



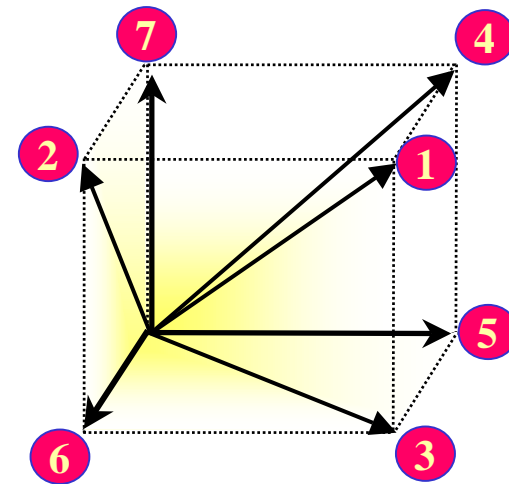
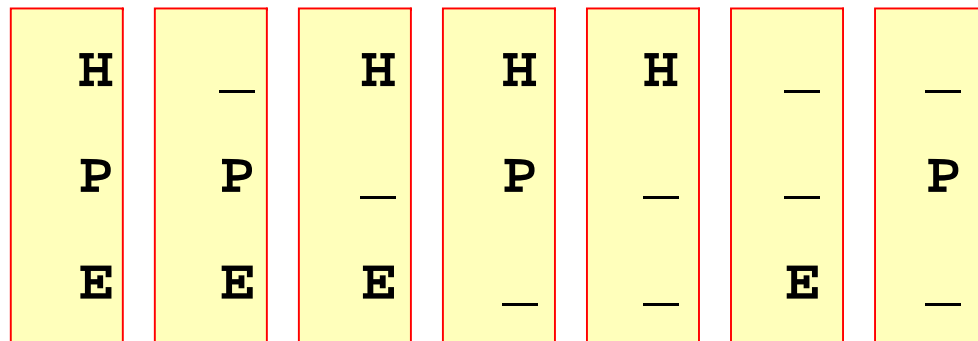
aber:
Rechenzeit und
Speicherbedarf!

Drei dimensionale Alignment-Matrix

- Bei zwei Sequenzen gibt es immer drei Möglichkeiten, ein Alignment fortzusetzen (vgl. paarweises Alignment letzte Woche):



- Bei 3 Sequenzen gibt es 7 Möglichkeiten:



- 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 für jede Stelle in den Sequenzen berechnet (also überall in dem $L_1 * L_2 * L_3 \dots * L_N$ -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 ($= \sim 10^{17}$ s.) berechnet werden?

A: => Dreisatz:

$$50^2(2^2-1)*C = 1 \text{ s.}$$

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

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

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

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

$$N = 11$$

... 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)

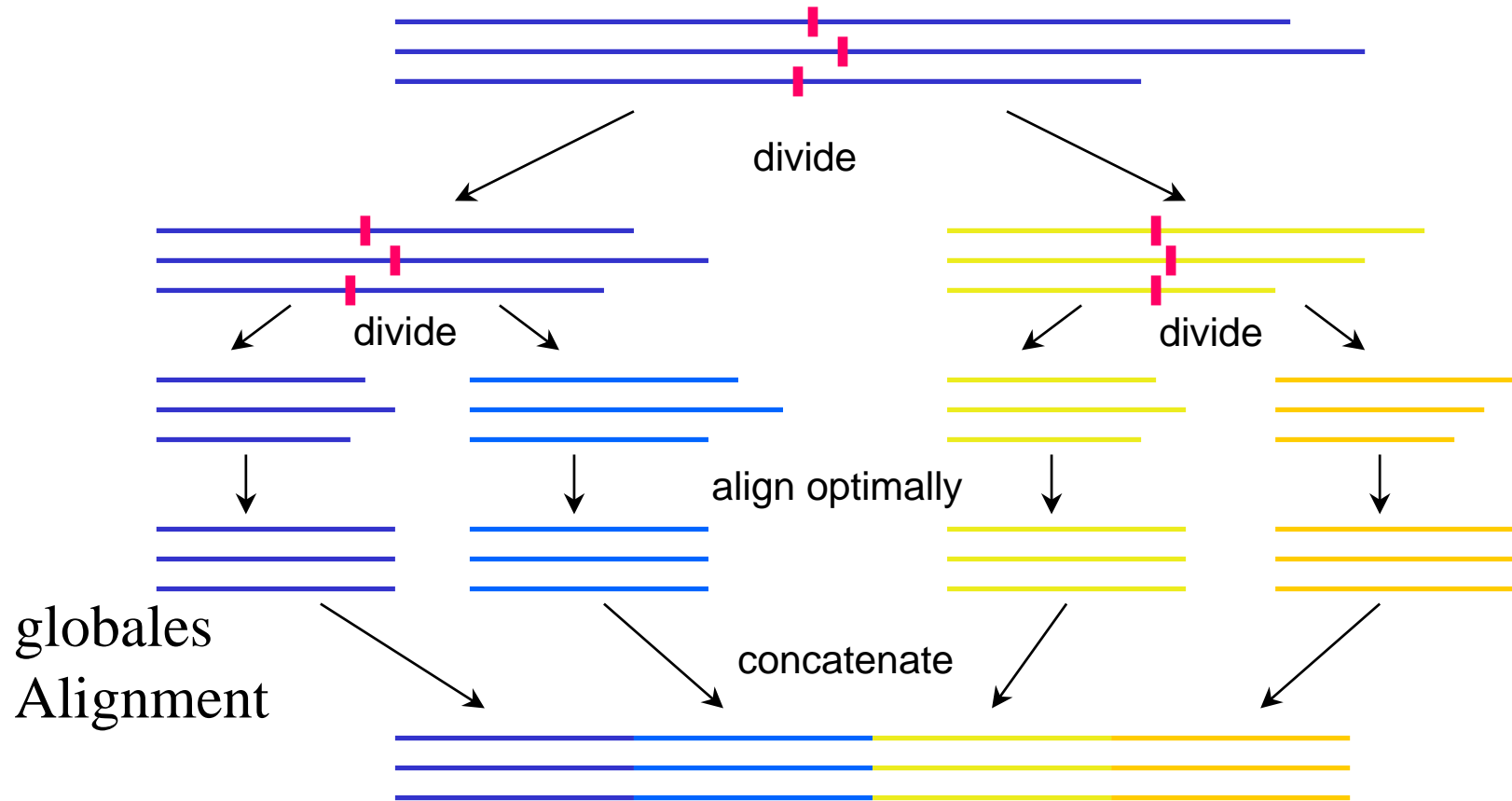
DCA (Jens Stoye)

beide Ansätze verkleinern den Suchraum, DCA wird
beispielhaft erläutert

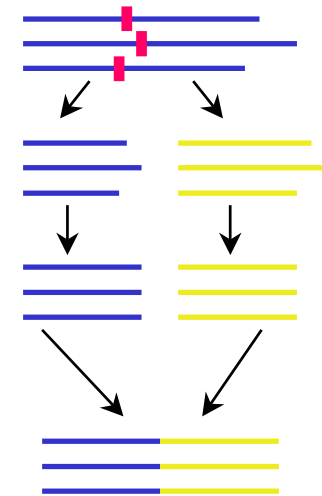
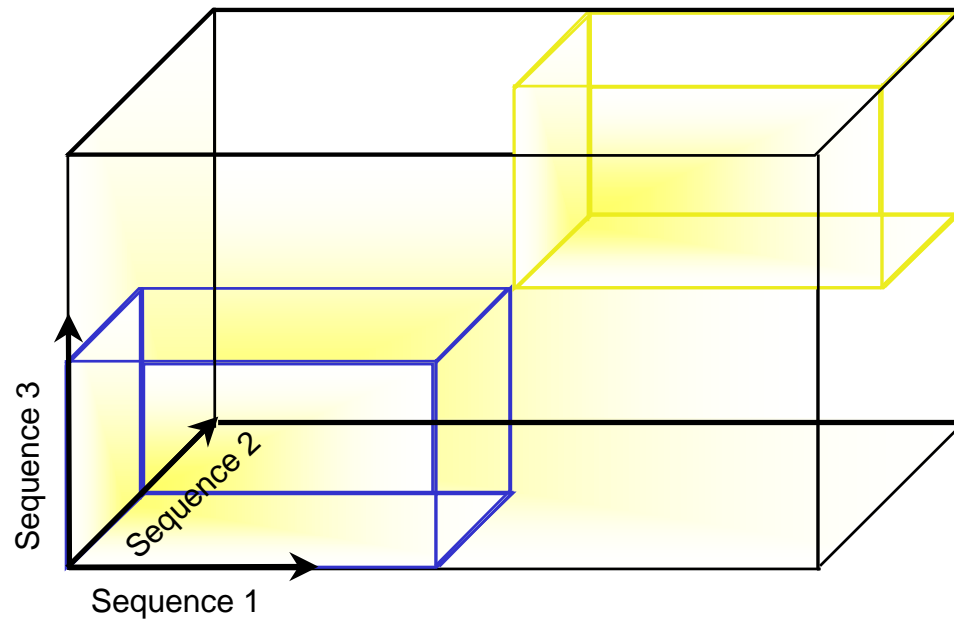
Divide-and-Conquer Alignment (DCA)

- Verkleinern des Suchraums (\Rightarrow 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 genügend klein sind
 - multidimensionales dynamisches Programmieren (alignieren)
 - Die alignierten Sequenzen werden wieder zusammengefügt und der Gesamtscore (Sum-of-Pairs-Score) berechnet
- \Rightarrow entscheidend sind die Zerschneidepunkte...

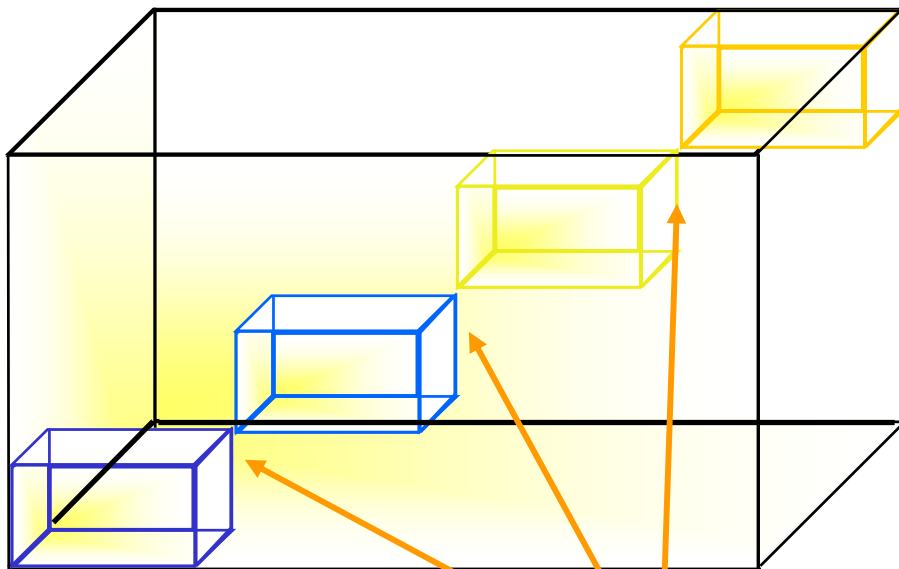
Divide-and-Conquer Alignment



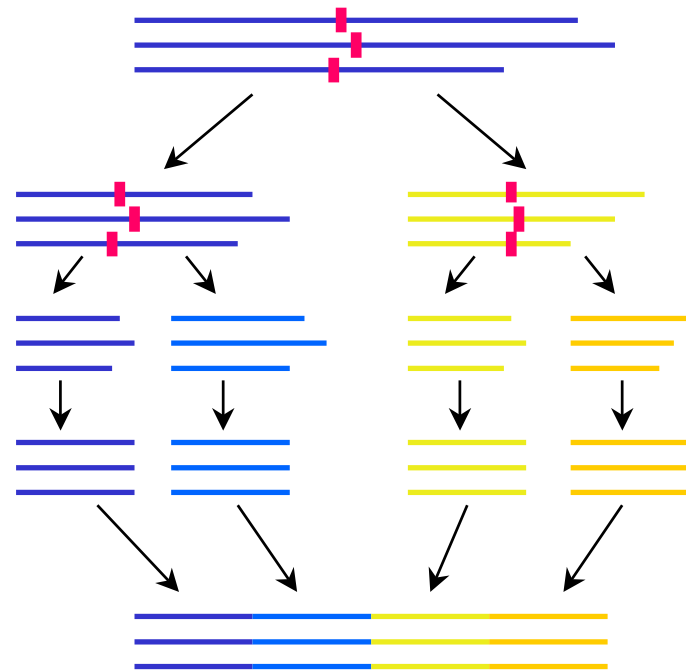
Reduzierung des Suchraums



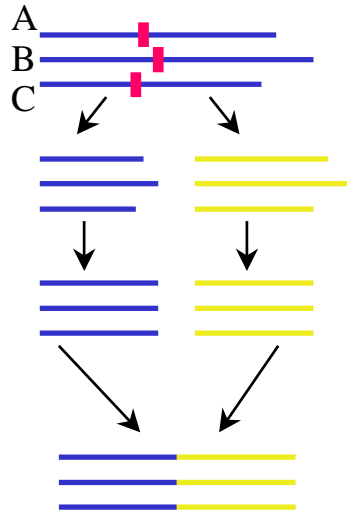
Reduzierung des Suchraums



Zerschneidestellen



Auffinden geeigneter Schneidestellen



- Sequenz A wird an c_1 zerschnitten
- Sequenz B an c_2
- Prefixe und Suffixe werden jeweils paarweise aligniert und der jeweilige Score berechnet (S_{Prefix} , S_{Suffix})
- Score der unzerschnittenen Gesamtsequenz wird berechnet (S_{Complete})

Kostenfunktion
eines Paares:

$$C(c_1, c_2) := S_{\text{Prefix}} + S_{\text{Suffix}} - S_{\text{Complete}}$$

Kostenfunktion klein \Rightarrow Alignment der Stückchen \approx 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