

Bioinformatik

Profilalignment und PSI Blast
Center-Star Verfahren
Progressives MSA – CLUSTAL W



Ulf Leser

Wissensmanagement in der
Bioinformatik



Definition

- Bisher
 - Immer Vergleich zweier Strings
- Jetzt
 - Multipler Stringvergleich: Vergleich von $k > 2$ Strings
- Definition
 - *Ein **multiple Sequenzalignment (MSA)** von k Strings S_i , $1 \leq i \leq k$, ist eine Tabelle mit k Zeilen und l Spalten, so dass*
 - *In Zeile i steht String S_i , mit beliebig eingefügten Leerzeichen*
 - *Jedes Zeichen jedes S_i steht in exakt einer Spalte*
 - *In keiner Spalte stehen nur Leerzeichen*
- Bemerkungen
 - Direkte Generalisierung des Alignment zweier Strings
 - Es folgt, dass $l = |\text{MSA}| \leq \sum(|S_i|)$
 - Warum?

Motivation II

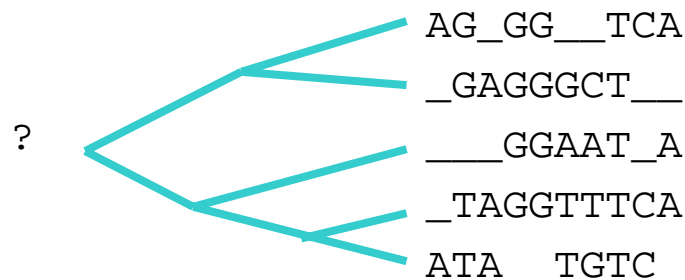
- Proteine (und damit auch DNA) setzen sich aus funktionalen Blöcken und „Zwischenraum“ zusammen
 - Die Blöcke findet man nicht, wenn man Sequenzen nur paarweise vergleicht
 - Bzw. man kann sie nicht vom Rauschen unterscheiden
 - Wenn aber mehr Sequenzen vorhanden sind
 - Findet ein MSA die (wenigen) Gemeinsamkeiten
 - Sind diese auch statistisch signifikant
- Sprich: Trennung des eventuell zufällig Gemeinsamen (Alignment) vom bedeutungsvoll Gemeinsamen (MSA)

```
AAC__GTG__AT__T__GAC__  
_TCGAGTGC_TTTACA_GT
```

```
AAC__GTG__AT__T__GAC__  
_TCGAGTGC_TTTACA_GT  
GCCG__TGC__TA__GTCG_  
TTC__AGTGGACGTG__GTA  
G____GTGCA__TGACC__
```

MSA Zielfunktion

- **Zielfunktion** beim einfachen Alignment war klar
 - Möglichst wenig I,R,D
 - Eventuell mit Substitutionsmatrix
 - Eventuell mit spezieller Behandlung von Gaps
- Zielfunktion für MSA ist nicht so klar
 - Score einer Spalte mit 2 T, zwei G und einem Leerzeichen?
 - Angabe einer Substitutionsmatrix für k Sequenzen über Alphabet Σ würde $O(|\Sigma|^{k+1})$ Werte erfordern
 - Nicht machbar und biologisch nicht begründbar



Definitionen

- Definition

- Gegeben ein MSA M für Sequenzen S_1, \dots, S_k . Das *durch M induziertes Alignment für zwei Sequenzen S_i und S_j ($i \neq j$)* ist
 - Entferne aus M alle Zeilen außer i und j
 - Entferne alle Spalten, die in i und j ein Leerzeichen enthalten
- Gegeben ein MSA M für Sequenzen S_1, \dots, S_k . Der *Sum-Of-Pairs Score für M (SP-Score)* ist die Summe aller Alignmentsscores der durch M induzierten paarweisen Alignments
- Das *SP-Alignment Problem für Sequenzen S_1, \dots, S_k* sucht das MSA M , dessen SP-Score minimal ist

- Bemerkung

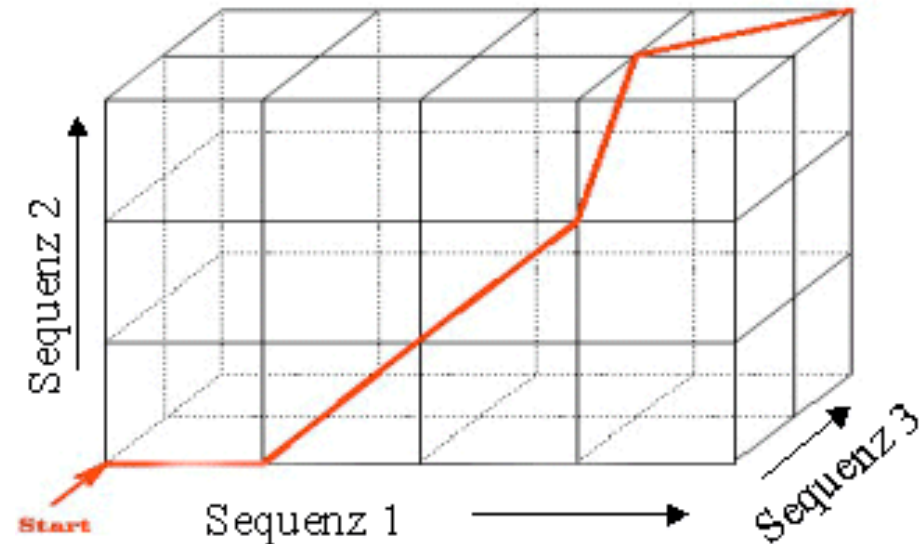
- Vergleich aller Sequenzen mit allen anderen Sequenzen
- Für die paarweisen Alignments kann man beliebige Substitutionsmatrizen benutzen

Dynamische Programmierung in k Dimensionen

- $k = 2$
 - 2-dimensionale Matrix

		0	1	2	3	4	5	6	7
			w	r	i	t	e	r	s
0		0	1	2	3	4	5	6	7
1	v	1	1	2	3	4	5	6	7
2	i	2	2	2	2	3	4	5	7
3	n	3	3	3	3	3	4	5	6
4	t	4	4	4	4	3	4	5	6
5	n	5	5	5	5	4	4	5	6
6	e	6	6	6	6	5	4	5	6
7	r	7	7	6	7	6	5	4	5

- $k = 3$
 - 3-dimensionale Matrix



Formaler

- Wir nehmen ein einfaches Kostenmodell (I/D/R=1, M=0)
- Theorem
 - Gegeben Sequenzen S_1, S_2, S_3 .
 - Sei $d(i,j,k)$ der Score des SP-optimalen Alignments der Strings $S_1[1..i], S_2[1..j], S_3[1..k]$
 - Sei $c_{ij} = 0$, wenn $S_1(i) = S_2(j)$, sonst 1
 - Sei $c_{ik} = 0$, wenn $S_1(i) = S_3(k)$, sonst 1
 - Sei $c_{jk} = 0$, wenn $S_2(j) = S_3(k)$, sonst 1
 - Dann berechnet sich $d(i,j,k)$ als

$$d(i, j, k) = \min \left\{ \begin{array}{llll} d(i-1, j-1, k-1) + c_{ij} & + c_{ik} & & + c_{jk} \\ d(i-1, j-1, k) & + c_{ij} & + 2 & \\ d(i-1, j, k-1) & + c_{ik} & + 2 & \\ d(i, j-1, k-1) & + c_{jk} & + 2 & \\ d(i-1, j, k) & & + 2 & \\ d(i, j-1, k) & & + 2 & \\ d(i, j, k-1) & & + 2 & \end{array} \right.$$

Komplexität

- Für drei Sequenzen der Länge n
 - Würfel hat n^3 Zellen
 - Für jede Zelle sind 7 Berechnungen notwendig
 - Zusammen $O(7 * n^3)$
- Allgemeiner Fall: k Sequenzen der Länge n
 - Hyperwürfel hat n^k Zellen
 - Für jede Zelle sind $2^k - 1$ Berechnungen notwendig
 - Alle Ecken eines k -dimensionalen Würfels minus eins
(Das ist die Ecke die gerade berechnet wird)
 - Zusammen $O(2^k * n^k)$
- Tatsächlich gilt
 - *Das SP-Alignment Problem ist NP vollständig*

Inhalt dieser Vorlesung

- Profilalignment
 - Suche mit einem MSA
- Andere Scoringfunktionen
 - Center-Star Methode
- CLUSTAL W
 - Heuristisches, progressives Alignment

Suche mit MSA

- Erinnerung: Erzeugung von Proteinfamilien
 - Starte mit Proteinen gleicher/ähnlicher Funktion
 - Finde das Gemeinsame durch MSA
 - Suche „damit“ nach weiteren Vertretern
 - Modifiziere Familie entsprechend
 - Iteriere, bis Zufriedenheit eintritt
- Wie sucht man mit einem MSA?
 - Wir müssen entscheiden, wie gut eine (neue) Sequenz S zu einem MSA M passt
 - Profiles

Profile

- Definition

Gegeben ein MSA M mit n Spalten, $\Sigma' = \Sigma \cup \{-\}$

- Das *Profil P* zu M ist eine Tabelle der Größe $n * |\Sigma'|$
- In der Zelle (i,j) steht die *relative Häufigkeit des Zeichens j in der Spalte i*

- Beispiel

S ₁	A	G	C	-	A
S ₂	A	G	A	G	A
S ₃	A	C	C	G	-
S ₄	C	G	-	G	C
A	0.75	0	0.25	0	0.50
G	0	0.75	0	0.75	0
C	0.25	0.25	0.50	0	0.25
-	0	0	0.25	0.25	0.25

Profile und Sequenzen

- Mit Hilfe von Profilen bewerten wir Sequenzen
 - Messen der **Ähnlichkeit der Sequenz zum MSA**
- Das setzt wieder ein Alignment voraus
 - Welche Zeichen der Sequenz sollen mit welchen Spalten des MSA verglichen werden?
- Gegeben Sequenz S und Profil P. Wir brauchen also zwei Dinge
 - Methode zur Bewertung eines konkreten Alignments von S und P (bzw. S und dem MSA)
 - Methode zum Finden des besten Alignments gegeben die Bewertungsfunktion

Bewertung eines Alignments

- Definition

Gegeben ein Profil P mit n Spalten, eine Sequenz S und eine Substitutionsmatrix m . Seien c_j die j Zeichen des Alphabets Σ' .

- *Sei n^* der String bestehend aus der Konkatenation der Zahlen $1\dots k$*
- *Ein **Alignment** A von S gegen P ist ein Untereinanderschreiben von n^* und S mit beliebigen Leerzeichen*
- *Seien k' der obere String (n^*) und S' der untere String in A . Wir erzeugen ein P' aus P , in dem wir an allen Stellen, an denen k' ein Leerzeichen enthält, eine leere Spalte in P einfügen*
- *Der **Score** $s(A)$ von A berechnet sich als*

$$s(A) = \sum_{i=1}^{|A|} \sum_{j=1}^{|\Sigma'|} \begin{cases} (P'[j, i] * m[c_j, S'[i]]), & \text{wenn } k' \neq \text{"_"} \\ m[_, S'[i]], & \text{sonst} \end{cases}$$

Beispiel

Profil P

	1	2	3	4	5
A	0.75		0.25		0.50
G		0.75		0.75	
C	0.25	0.25	0.50		0.25
-			0.25	0.25	0.25

Ähnlichkeitsmatrix m

	A	G	C	-
A	2	-1	-3	-1
G		2	-1	-1
C			2	-1

Sequenz S: AAGGC

Alignment A

1	-	2	3	4	5
A	A	G	-	G	C

$$\begin{aligned}
 s(A) &= (2 \cdot 0.75 + -1 \cdot 0 + -3 \cdot 0.25 + -1 \cdot 0) + \\
 &\quad (-1) + \\
 &\quad (-1 \cdot 0 + 2 \cdot 0.75 + -1 \cdot 0.25 + -1 \cdot 0) + \\
 &\quad (-1 \cdot 0.25 + -1 \cdot 0 + -1 \cdot 0.50 + -1 \cdot 0.25) + \\
 &\quad (-1 \cdot 0 + 2 \cdot 0.75 + -1 \cdot 0 + -1 \cdot 0.25) + \\
 &\quad (-3 \cdot 0.50 + -2 \cdot 0 + 2 \cdot 0.25 + -1 \cdot 0.25) \\
 &= 0.75 - 1 + 1.25 - 1 + 1.25 - 1.25 \\
 &= 0
 \end{aligned}$$

Optimale Profilalignments

- Wie finde ich das optimale Profilalignment?
 - Natürlich mit **dynamischer Programmierung**

- Theorem

Gegeben eine Substitutionsmatrix m , Profil P , Sequenz S

- *Sei $c(x,i)$ der Score für das Alignieren eines Zeichen x mit Spalte i in P , also*

$$c(x,i) = \sum_{j=1}^{|\Sigma|} P[j,i] * m[c_j, x]$$

- *Sei $v(i,j)$ der Score für das optimale Alignment von $S[1..j]$ mit den ersten i Spalten von P . v berechnet sich als*

$$v(i, j) = \max \left(\begin{array}{l} v(i-1, j) + c(S[i], _) \\ v(i, j-1) + c(_, j) \\ v(i-1, j-1) + c(S[i], j) \end{array} \right)$$

PSI-BLAST

- PSI Blast – Implementierung der iterativen Suchstrategie
 - Gegeben Suchsequenz S: Berechne Profil P
 - Durchsuche DB mit P
 - Tatsächlich wird eine lineare Variante ohne Gaps verwendet
 - Bilde ein multiples Alignment von allen Hits
 - Berechne daraus eine neues P
 - Iteriere, bis Stoppkriterien erfüllt
- **Erhöhung der Sensitivität** gegenüber einfachem BLAST
 - PSI-BLAST findet auch weiter entfernte Homologien
 - Zwitter zwischen Patternmatching / Homologiesuche

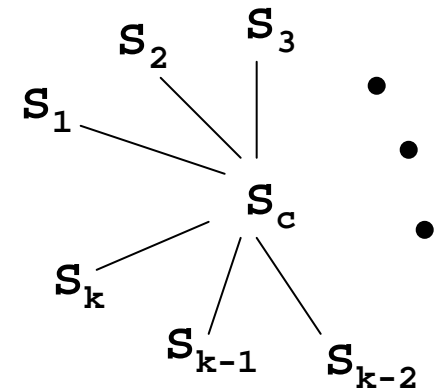
PSI-BLAST Probleme

- Multiples Sequenzalignment
 - Profile müssen aus MSA der gefundenen Sequenzen berechnet werden
 - MSA ist komplex – PSI-BLAST verwendet **Greedy-MSA**
- **Redundante Sequenzen**
 - n-fach vorkommende Sequenzen, z.B. ESTs
 - Gaukeln **falsche Positionsgewichte** vor
 - Bestimmte Base ist nicht evolutionär wichtig, sondern wurde nur sehr oft gemessen
 - Cut-Offs zur Entfernung fast-identischer Sequenzen
 - Reine Heuristik - auf Sequenzebene kann man nicht unterscheiden
 - Wirkliche redundante Sequenzen (identisches EST zweimal sequenziert)
 - Nicht redundante, sehr ähnliche Sequenzen (homologes Gen bei Maus und Mensch, Genduplikate)

-
- Andere Scoringfunktionen

MSA mit Consensussequenz

- SP minimiert Summe aller paarweisen Alignments
- Alternativ: Minimierung der **Summe der Alignments aller Sequenzen S_1, \dots, S_k mit einer Consensussequenz S_c**
 - S_c kann eine der S_i sein, muss aber nicht
 - Konstruktion von S_c z.B. durch Untereinanderschreiben und Wahl des häufigsten Buchstaben
- Center-Star Verfahren
 - Mit **guter Fehlergrenze**
- Braucht nur linear viele quadratische Alignments



Center-Star Verfahren

- Gegeben k Sequenzen der Länge n
- Wähle Sequenz S , die den **kleinsten durchschnittlichen Abstand** zu allen anderen Sequenzen hat
 - Wie komplex?
- Benutze S als Kern eines MSA M
- Iteriere
 - Wähle eine noch nicht alignierte Sequenz T
 - **Aligniere M und T**
 - Mit einer einfachen Methode, die wir hier nicht ausführen (Gusfield, p. 348)
 - Bis alle Sequenzen in M enthalten sind
- Beispiel für ein **progressives MSA** Verfahren
 - Sukzessive Sequenzen zu einem wachsenden MSA hinzufügen

Beispiel

1. ATGGC
2. AGCC
3. TGCGAT
4. GCATG
5. TGCCTA
6. CAACTA

	S1	S2	S3	S4	S5	S6
S1	0	2	4	4	4	5
S2	2	0	4	4	3	4
S3	4	4	0	3	3	5
S4	4	4	3	0	3	4
S5	4	3	3	3	0	3
S6	5	4	5	4	3	0
Durchschnitt	3,8	3,4	3,8	3,6	3,2	4,2

Beispiel 2

- Kern des „MSA“: TGCCTA
- Wähle Sequenz: S3= TGCGAT
- Alignment: TGCC_TA
TGCGAT_
- Wähle Sequenz: S2= AGCC
- Alignment: TGCC_TA
TGCGAT_
AGCC_
- Wähle Sequenz: S1= ATGGC
- Alignment: _TGCC_TA
TGCGAT
AGCC
ATGGC

Beispiel 3

- Wähle Sequenz: S4 = GCATG
- Alignment:
 _TGCC_TA
 TGCGAT
 _AGCC____
 ATGGC____
 __GC_ATG
- Wähle Sequenz: S6 = CAACTA
- Alignment:
 _TGCC_TA
 TGCGAT
 _AGCC____
 ATGGC____
 __GC_ATG
 CAAC__TA

Fehlergrenze

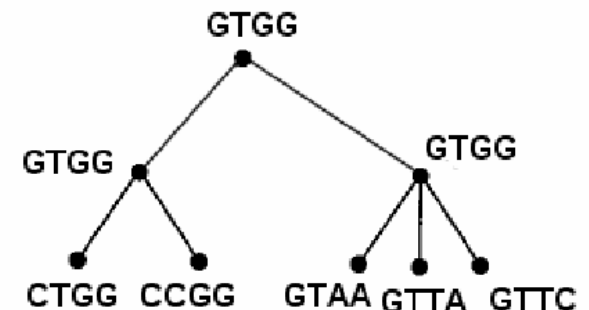
- Wenn für Zeichen i, j, k gilt: $d(i, j) + d(j, k) \geq d(i, k)$
 - Dreiecksungleichung
 - Gilt für einfaches Modell
 - Gilt **nicht für biologische Substitutionsmatrizen**
- Theorem
 - *Gegeben k Sequenzen, $k \geq 3$*
 - *Sei d der SP Score des durch den Center-Star Algorithmus berechneten MSA*
 - *Sei d^* der optimale SP Score*
 - *Dann gilt*

$$\frac{d}{d^*} \leq 2 - \frac{1}{k} < 2$$

- Beweis
 - Siehe Gusfield

MSA mit phylogenetischen Bäumen

- Grundidee
 - Annahme: **Sequenzen durch Evolution** aus Ursequenz entstanden
 - Wir ordnen Sequenzen Knoten im Baum zu
 - Sequenzen haben sich damit entlang der Pfade entwickelt
- Den echten phylogenetischen Baum kennen wir aber nicht
- Zielfunktion
 - Suche den Baum T so, dass die **Summe aller Alignmentscores von benachbarten Sequenzen in T minimiert** wird
 - Aus T kann man ein MSA ableiten
- Leider
 - Das **phylogenetische Alignmentproblem ist NP vollständig**
 - Wir werden ein ähnliches Problem noch besprechen:
Phylogenie mit Maximum Parsimony



Bemerkung

- Wir haben nur globales MSA untersucht
- Auch lokales MSA ist definiert
 - Aber nicht so populär wie lokales Alignment von Sequenzen
 - Warum? Weil man MSA i.d.R. nur mit ähnlichen Sequenzen macht
- Definition
 - Gegeben k Sequenzen S_1, \dots, S_k . Ein *lokales multiples Sequenzalignment* erhält man, in dem man
 - Für jedes S_i , wähle eine Teilzeichenkette $s_i \in S_i$
 - Aligniere alle s_i durch globales MSA
- Ziel ist natürlich wieder solche s_i zu finden, die möglichst hohe MSA Scores erzeugen
- Beispiel: DALIGN

CLUSTAL W

- Eine Art Greedy-Variante des Alignments mit einem phylogenetischen Baum
- Lange Zeit das **Standardprogramm** zum multiplen Sequenzalignment
 - Higgins, D. G. and Sharp, P. M. (1988). "CLUSTAL: a package for performing multiple sequence alignment on a microcomputer." *Gene* **73**(1): 237-44.
 - Thompson, J. D., Higgins, D. G. and Gibson, T. J. (1994). "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice." *Nucleic Acids Res* **22**(22): 4673-80.
- Heute ist die Situation unübersichtlicher
 - DAlign, T-Coffee, HMMT, PRRT, MULTALIGN, ...

Progressives Alignment

- Grundproblem des SP-MSA
 - Ständige Betrachtung aller Sequenzen
 - Das sind zu viele Möglichkeiten = Dimensionen
- Grundidee der progressiven Verfahren
 - Berechne zunächst **MSA für viele kleine Teilmengen** von Sequenzen
 - Baue das **Gesamt-MSA aus den Teil-MSA**
- Das wirft Fragen auf
 - Wie wähle ich die Teilmengen?
 - Wie „aligniert“ man zwei MSA?
 - Wir können MSA (Profil) und Sequenz, aber zwei MSA?
 - In welcher Reihenfolge verschmilzt man die Teil-MSA?
 - Erreicht man optimale Lösung?
 - Nicht die SP-optimale Lösung
 - Aber in der Praxis akzeptable Ergebnisse nach viel tunen

CLUSTAL W: Grundaufbau

- Gegeben k Sequenzen
- Drei Schritte
 - Berechne alle paarweisen Alignmentsscores
 - Konstruiere „Guide Tree“ durch hierarchisches Clustering
 - Berechne und verschmelze Teil-MSA gemäß dem Guide Tree
- Idee dahinter
 - Aligniere erst sehr ähnliche Sequenzen – **Signale werden verstärkt**
 - Werden z.B. zwei sehr verschiedene Cluster von Sequenzen betrachtet, berechnet CLUSTAL automatisch erst zwei (homogene) MSA und verschmilzt diese am Ende
 - Hohe Chance, dass **konservierte Blöcke** erhalten bleiben
 - **Außenseiter** kommen erst spät dazu und können die Blockstruktur nicht mehr stören
 - Orientierung an der „tatsächlichen“ Entstehungsgeschichte, dem **phylogenetischen Baum**

Schritt 1 und 2

- Berechnen der Ähnlichkeitsmatrix M
 - Berechne die $O(k^2)$ paarweisen Alignmentscores für die k Sequenzen
- Hierarchisches Clustering
 - Wähle Zelle (i,j) mit kleinstem Abstand aus Matrix M
 - Das ist das erste Paar
 - Erzeuge M': Lösche die Sequenzen i und j aus M und füge neue Spalte/Zeile (ij) ein
 - Für alle $k \neq ij$: $M'[ij,k] = (M[i,k] + M[j,k]) / 2$
 - Mittlerer Abstand zu i und j
 - Iteriere, bis Matrix nur noch 2x2 groß ist
- Tatsächlich benutzt CLUSTAL W heute Neighbour Joining
 - Details später: Ultrametrien und additive Bäume

Beispiel: Graphisch

A
B
C
D
E
F
G

ABCDEFG

A
B.
C..
D...
E....
F.....
G.....

(B,D) → a

A
B
C
D
E
F
G



ACEFGa

A
C.
E..
F...
G....
a.....

(E,F) → b

A
B
C
D
E
F
G

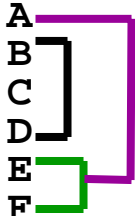


ACGab

A
C.
G..
a...
b....

(A,b) → c

A
B
C
D
E
F
G

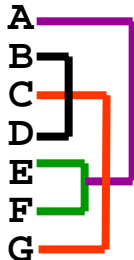


CGac

C
G.
a..
c...

(C,G) → d

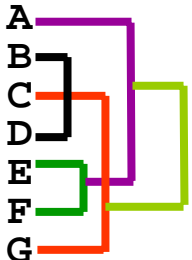
A
B
C
D
E
F
G



acd
a
c.
d..

(d,c) → e

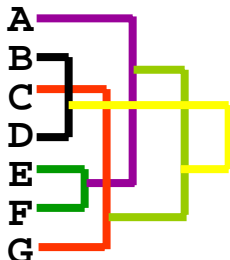
A
B
C
D
E
F
G



ae
a
e.

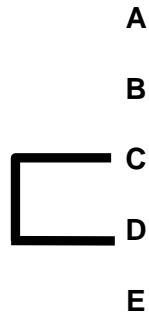
(a,e) → f

A
B
C
D
E
F
G

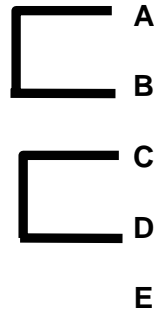


Konstruktion des Guide Trees

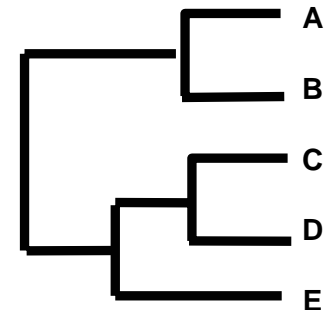
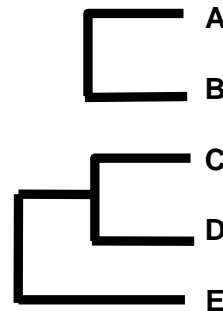
	A	B	C	D	E
A		17	59	59	77
B			37	61	53
C				13	41
D					21



	A	B	E	CD
A		17	77	59
B			53	49
E				31



	E	CD	AB
E		31	65
CD			54



Schritt 3: Progressive MSA Generierung

- Berechnung paarweiser Alignments in der Reihenfolge des Guide Trees
- Alignment eines MSA M_1 mit einem MSA M_2
 - Dynamische Programmierung mit linearem Gapscore
 - Wert eines Mismatches/Matches ist der **Durchschnittsscore aller Paare** mit einem Zeichen aus M_1 und dem anderen aus M_2
 - **Gaps** werden mit dem schlechtesten Score der verwendeten Substitutionsmatrix bestraft
 - Bei k Sequenzen sind das maximal $k/2 * k/2 = O(k^2)$ Scores

- Beispiel

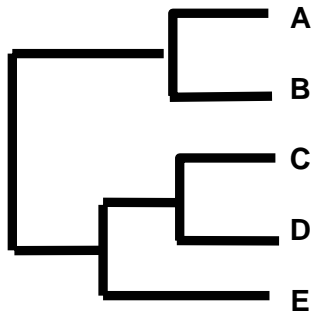
A ...P...
 B ...G...
 C ...P...

 D ...A...
 E ...A...
 F ...Y...

- Score

$$\begin{aligned}
 & - (2 * s(P, A) + s(P, Y) + \\
 & \quad 2 * s(G, A) + s(G, Y) + \\
 & \quad 2 * s(P, A) + s(P, Y)) / 9
 \end{aligned}$$

Beispiel



C PADKTNVKAAWGKVG**A**HAGEYGA

D AADKTNVKAAWSKVGGHAGEYGA

A PEEKSAVTALWGKVN**V**DEYGG

B GEEKAAVLALWDKVNEEEYGG

C PADKTNVKAAWG_KVGAHAGEYGA

D AADKTNVKAAWS_KVGGHAGEYGA

E AA__TNVKTAWSSKVGGHAPA__A

A PEEKSAV_TALWG_KVN__VDEYGG

B GEEKAAV_LALWD_KVN__EEYGG

C PADKTNV**KAA**_WG_KVGAHAGEYGA

D AADKTNV**KAA**_WS_KVGGHAGEYGA

E AA__TNV**KTA**_WSSKVGGHAPA__A

Once a gap, always a gap

Erweiterungen

- Viele Tricks und Heuristiken
 - Individuelle Scores für das **Öffnen eines Gaps** in Abhängigkeit der Umgebung, Abstand zu anderen Gaps, Länge der Sequenz, ...
 - Parameter gelernt aus vielen Beispielen
 - Verwendung **unterschiedlicher Substitutionsmatrizen**, je nachdem wie hoch man schon im Baum ist
 - Denn damit steigt der evolutionäre Abstand, und PAM-X bzw. BLOSUM-X Matrizen werden nach dem geschätzten Abstand gewählt
 - **Gewichten der Sequenzen** im MSA/MSA Alignment je nach Ähnlichkeit zu anderen Sequenzen
 - Aussenseiter kommen später ins MSA, zählen dann aber mehr
 - Verhindert, dass sehr ähnliche Sequenzen das ganze MSA dominieren
 - Verwendung von **Neighbour Joining** statt hierarchischem Clustering
- Ergebnis: Deutlich verbesserte Ergebnisse
 - Durch Benchmarking gezeigt
 - Orientiert sich am 3D Strukturalignment als Goldstandard



Iterative Verfahren

- CLUSTAL W ist „greedy“
 - Ergebnis abhängig von der Reihenfolge des Hinzunehmens von Sequenzen
 - Der Guide Tree kann (evolutionär) falsch sein oder wenig Aussagekraft haben, wenn Unterschiede nur sehr klein sind
 - Was am Anfang schief läuft, ist besonders schlimm
 - „Once a gap, always a gap“
 - Einmal getroffene Alignments werden nie mehr korrigiert, nur „gestreckt“
- Was kann man tun?
 - Verschiedene Trees probieren und Ergebnisse vergleichen
 - Sampling – verschiedene Sequenzmengen versuchen
 - **Iterative Verfahren**
 - Sukzessive Verbesserung eines (progressiv gefundenen) Alignments
 - Jede Sequenz einmal entfernen und neu alignieren
 - Solange bis Konvergenz

Zusammenfassung

- Multiples Sequenzalignment
 - Irgendwie eine natürliche Erweiterung des paarweisen Alignments
 - Aber **Optimalität lässt sich auf viele Wege definieren**
- Sum-Of-Pair MSA mit exponentieller Komplexität
 - Keine Verwendung in der Praxis
- Praktisch werden nur heuristische Verfahren verwendet
 - ClustalW, Dalign, hmmt, T-Coffee, ...