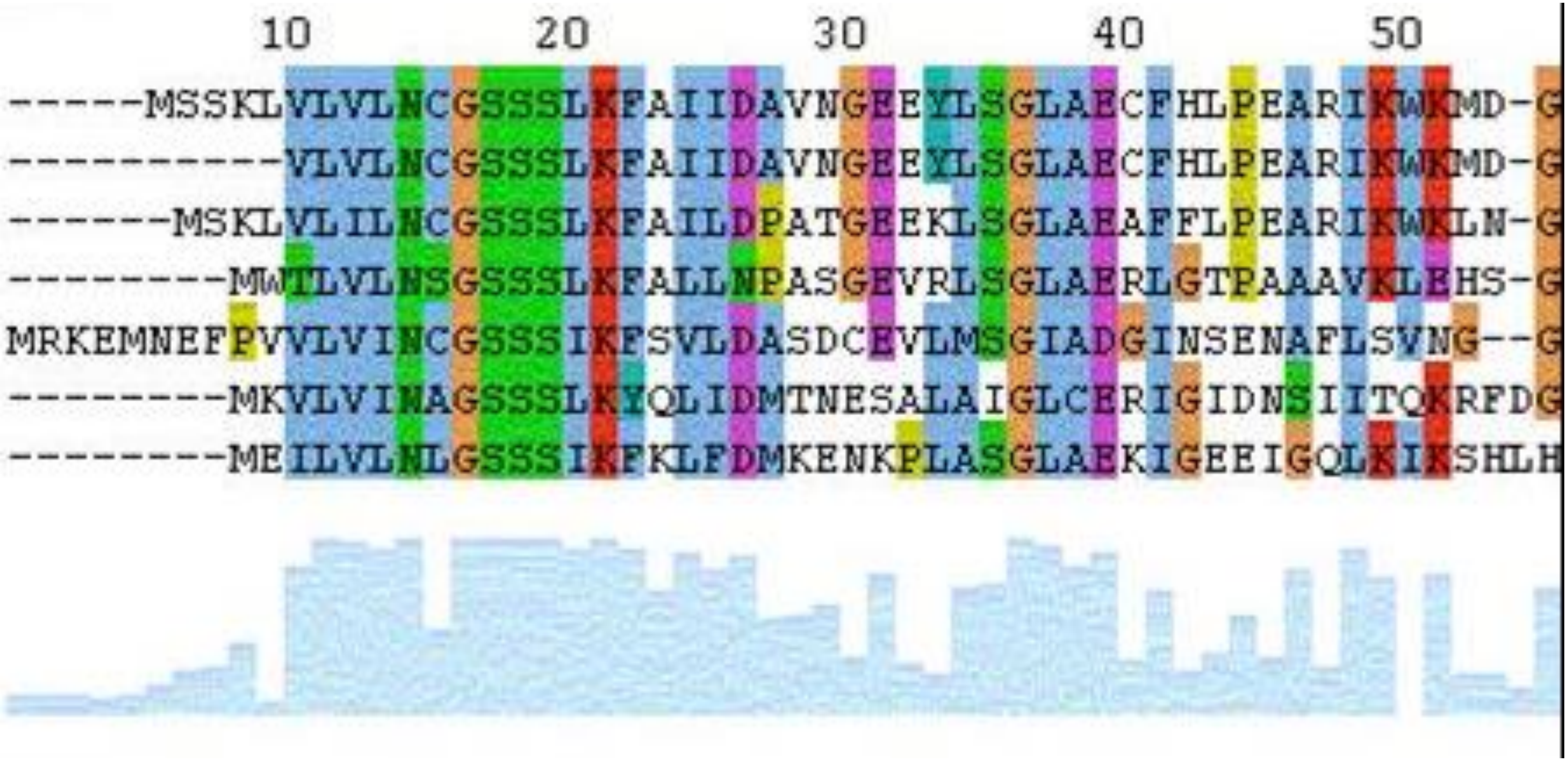
CISC 636 Computational Biology & Bioinformatics (Fall 2016)

Multiple Sequence Alignment

- Scoring
- Dynamic Programming algorithms
- Heuristic algorithms
  - CLUSTAL W



```
         10        20        30        40        50
-----MSSKLVLVLNCGSSSLKFAIIDAVNGEEYLSGLAECFHLPEARIKWKMD-G
----------VLVLNCGSSSLKFAIIDAVNGEEYLSGLAECFHLPEARIKWKMD-G
------MSKLVLILNCGSSSLKFAILDPATGEEKLSGLAEAFFLPEARIKWKLN-G
--------MWTLVLNSGSSSLKFALLNPASGEVRLSGLAERLGTPAAAVKLEHS-G
MRKEMNEFPVVLVINCGSSSIKFSVLDASDCEVLMSGIADGINSENAFLSVNG--G
--------MKVLVINAGSSSLKYQLIDMTNESALAIGLCERIGIDNSIITQKRFDG
--------MEILVLNLGSSSIKFKLFDMKENKPLASGLAEKIGEEIGQLKIKSHLH
```

Courtesy of jalview

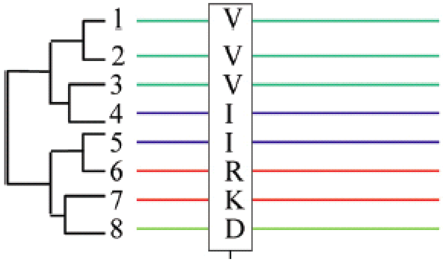
Motivations

- Collective statistic
- Protein families
- Identification and representation of conserved sequence features (motifs)
- Deduction of evolutionary history (Phylogeny)

Type of approaches

- Multidimensional dynamic programming
- Progressive alignment
  - Clustal W
- Iterative pairwise
- Probabilistic (HMMs)

Scoring a multiple alignment



- Ideally, should take into account
  - Some positions are more conserved than others – position specific scoring. (columns)
  - Sequences are not independent, they evolved as depicted by phylogenetic trees. (rows)
- In practice, each position (column) is scored independently

  $S(m) = G + \sum_i S(m_i)$ where m_i stands for column i of the multiple alignment m, G is a function for scoring the gaps.

  - Note: Hidden Markov models take into account position correlation, but just locally.

Column score

- Ideally, a column with three rows should scored as

$$S(a, b, c) = \log(p_{abc} / q_a q_b q_c) \qquad (1)$$

- Sum of pairs :SP scores

This means that the score in eq(1) is approximated as

$$S(a,b,c) = S(a,b) + S(a, c) + S(b, c) =$$
$$\log(p_{ab} / q_a q_b) + \log(p_{ac} / q_a q_c) + \log(p_{bc} / q_b q_c) \qquad (2)$$

To apply this SP scores to every position *i* in MSA m, we have

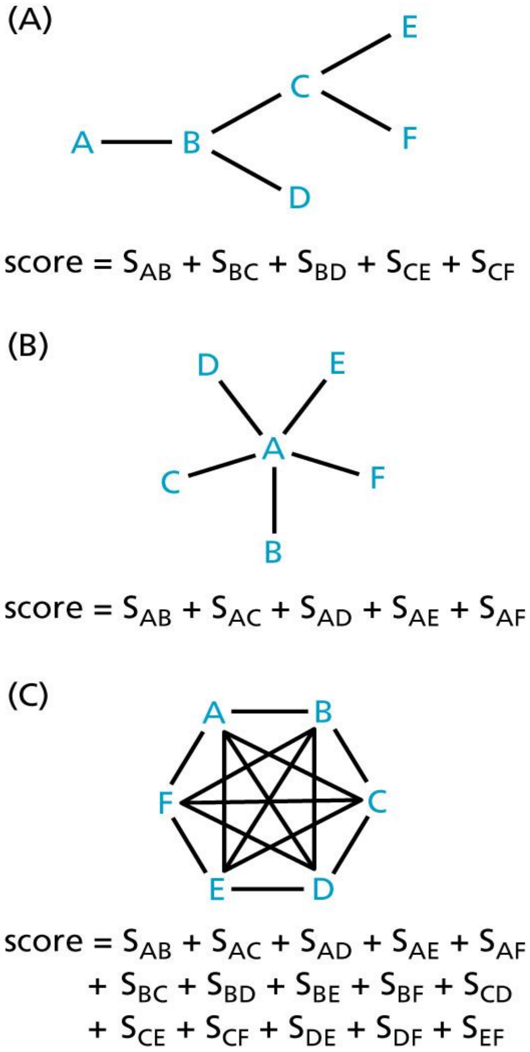
$$S(m_i) = \sum_{k<l} S(m_i^k, m_i^l),$$

where m_i^k stands for residue at position i of sequence k. Scores S(a, b) come from a substitution scoring matrix, e.g., PAM.

Note: scoring gaps

$s(a, -) = s(-, a) = -d$, $s(-,-) = 0$ (Once a gap, always a gap)

Common ways to construct alignment score from pairwise scores.



score = $S_{AB} + S_{BC} + S_{BD} + S_{CE} + S_{CF}$

score = $S_{AB} + S_{AC} + S_{AD} + S_{AE} + S_{AF}$

$$\begin{aligned}\text{score} = {} & S_{AB} + S_{AC} + S_{AD} + S_{AE} + S_{AF} \\ & + S_{BC} + S_{BD} + S_{BE} + S_{BF} + S_{CD} \\ & + S_{CE} + S_{CF} + S_{DE} + S_{DF} + S_{EF}\end{aligned}$$

This is the SP score used in the previous slide

Example of SP scoring

F
F
F
I
V

$$
\begin{aligned}
S = \; & S(F,F) + S(F,F) + S(F, I) + S(F,V) \\
& + S(F,F) + S(F,I) + S(F,V) \\
& + S(F,I) + S(F,V) \\
& + S(I,V) \\
& = 8 + 8 + 0 -1 + 8 + 0 -1 +0 -1 + 4 = 25
\end{aligned}
$$

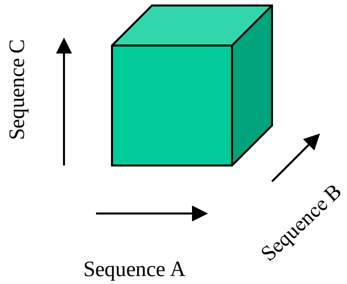
F
F
F
I
N

$$
\begin{aligned}
S = \; & S(F,F) + S(F,F) + S(F, I) + S(F,N) \\
& + S(F,F) + S(F,I) + S(F,N) \\
& + S(F,I) + S(F,N) + S(I,N) \\
& = 8 + 8 + 0 -4 + 8 + 0 -4 +0 -4 + 4 = 16
\end{aligned}
$$

Note: Blosum 50 is used

Approach 1: Multidimensional dynamic programming

- Given the scoring scheme, multiple sequences can be aligned using the same dynamic programming procedure used for aligning two sequences
- For example, when aligning three sequences, the DP table becomes a cube. Time required to filled out the cube is L^3 where L is the length of the sequences



- Thus, Aligning N sequences requires L^N time
  - **NP complete problem** (L. Wang and T. Jiang, 1994)
- An exact optimal alignment of multiple sequences has been considered as the Holy Grail in bioinformatics.

Approach 2: Progressive Alignment

- Basic procedure
  - Determine pairwise distance between sequences
  - Use a distance-based method to construct a guide tree
  - Add sequences to the growing alignment following the order in the guide tree

- Pros and cons
  - Progressive alignments are fast
  - Heuristic (greedy algorithm without backtracking) may get trapped at the local optimum
  - Error propagation

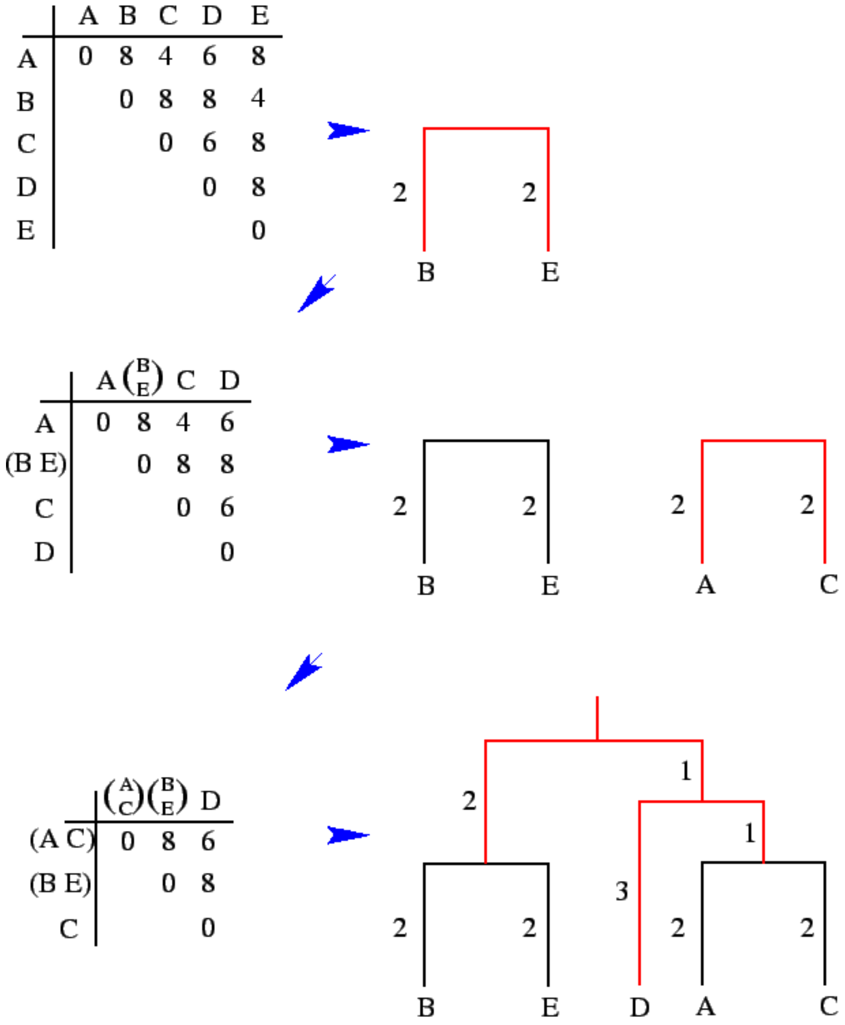
Approach 2: Progressive Alignment

- Distance-based guide tree
  - Distances may be obtained from
    - Pairwise alignment
    - Hybridization
  - Tree can be built by using
    - UPGMA (Unweighted Pair Group Method of Averages)
    - Neighbor joining

Approach 2: Progressive Alignment

UPGMA

- Fast and easy
- Robust to sequence errors
- Assumption of molecular clock, i.e. constant rate for evolution



	A	B	C	D	E
A	0	8	4	6	8
B		0	8	8	4
C			0	6	8
D				0	8
E					0

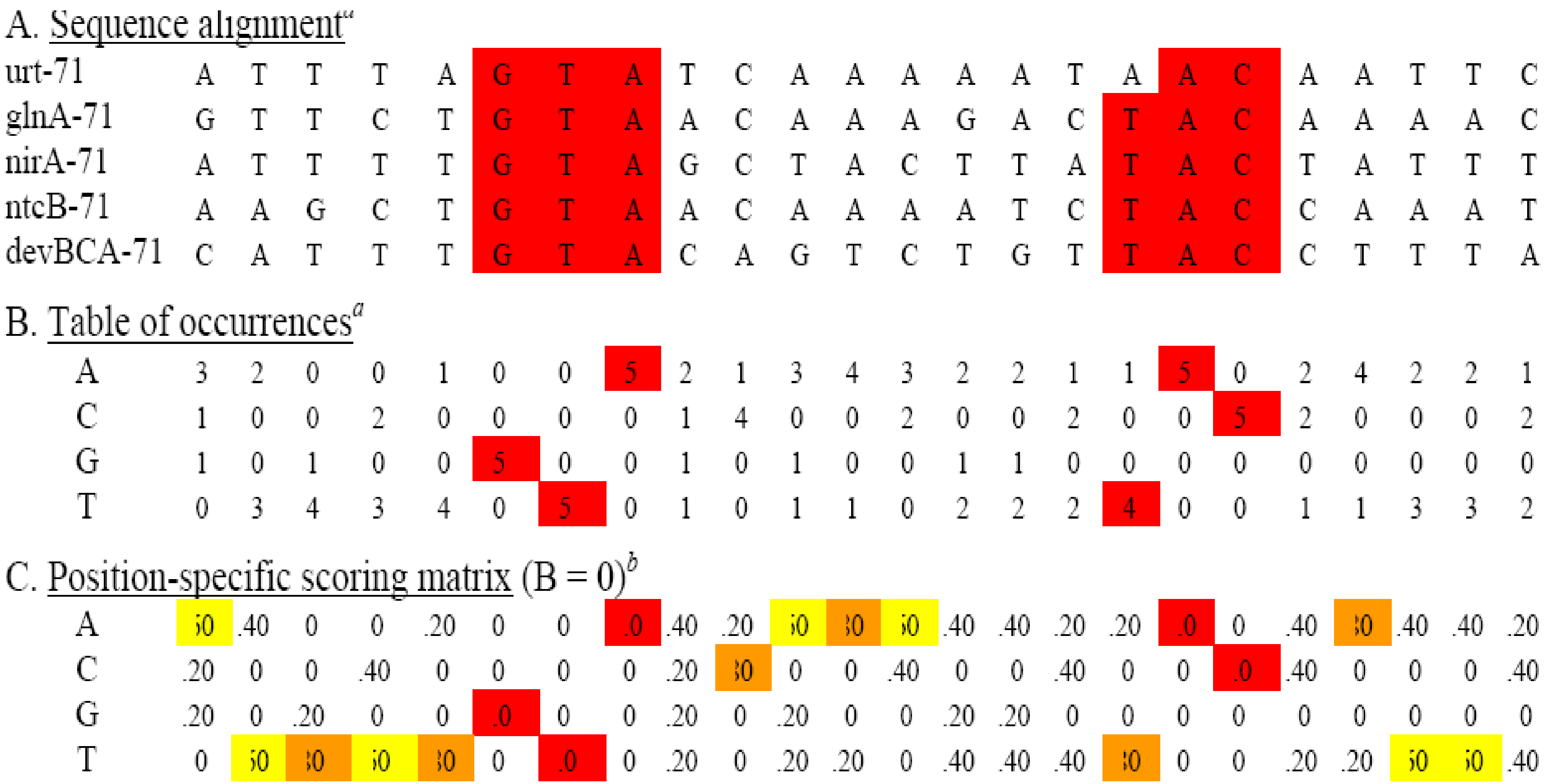
	A	(B E)	C	D
A	0	8	4	6
(B E)		0	8	8
C			0	6
D				0

	(A C)	(B E)	D
(A C)	0	8	6
(B E)		0	8
C			0

Figure: Construction of an ultrametric tree

Approach 2: Progressive Alignment

- Add sequences to the growing alignment by following the order in the guide tree
  - Represent a multiple alignment as profile (Position Specific Scoring Matrix)
    - Given an alignment, a profile at each column is a vector of 20 specifying the frequencies of 20 amino acids appearing in that column.
    - Construction of profiles based on multiple sequence alignment.



A. Sequence alignment[a]

urt-71	A	T	T	T	A	G	T	A	T	C	A	A	A	A	A	T	A	A	C	A	A	T	T	C
glnA-71	G	T	T	C	T	G	T	A	A	C	A	A	A	G	A	C	T	A	C	A	A	A	A	C
nirA-71	A	T	T	T	T	G	T	A	G	C	T	A	C	T	T	A	T	A	C	T	A	T	T	T
ntcB-71	A	A	G	C	T	G	T	A	A	C	A	A	A	A	T	C	T	A	C	C	A	A	A	T
devBCA-71	C	A	T	T	T	G	T	A	C	A	G	T	C	T	G	T	T	A	C	C	T	T	T	A

B. Table of occurrences[a]

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

C. Position-specific scoring matrix (B = 0)[b]

A	.60	.40	0	0	.20	0	0	1.0	.40	.20	.60	.80	.60	.40	.40	.20	.20	1.0	0	.40	.80	.40	.40	.20
C	.20	0	0	.40	0	0	0	0	.20	.80	0	0	.40	0	0	.40	0	0	1.0	.40	0	0	0	.40
G	.20	0	.20	0	0	1.0	0	0	.20	0	.20	0	0	.20	.20	0	0	0	0	0	0	0	0	0
T	0	.60	.80	.60	.80	0	1.0	0	.20	0	.20	.20	0	.40	.40	.40	.80	0	0	.20	.20	.60	.60	.40

Approach 2: Progressive Alignment

- Align a sequence to a profile

Treat as aligning two sequences. To align column i of profile P to sequence j-th residue (with amino acid b), the score is computed as follows.

$$s(i,j) = \sum_{a \in [20\ \text{amino acids}]} P_i\,(a)\ S(a, b)$$

where S(a,b) is any amino acid substitution score matrix that is in use (e.g., PAM250, or BLOSUM62).

Then, a DP algorithm can be applied to find an optimal alignment.

For example: PSI-BLAST

Approach 2: Progressive Alignment

- Align profile P to profile Q
  - The score for aligning position i of P to position j of Q

    $$S(i,j) = \sum_a \{P_i\ (a) \sum_b [Q_j\ (b)\ S(a,b)]\}$$

    Note: there are different scoring schemes. One other example is to use relative entropy:

    $$S(i,j) = \sum_a P_i(a) \log\ [P_i(a)\ /\ Q_j(a)]$$

  - Use DP to find optimal alignment, i.e., maximizing the total score.

Approach 2: Progressive Alignment

Algorithm: clustalw (Higgins and Sharp 1989)

i. construct a distance matrix of all N(N-1)/2 pairs by pairwise DP alignment
ii. construct a guide tree by a neighbor-joining method
iii. Progressively align at nodes in order of decreasing similarity, using sequence-sequence, sequence-profile, and profile-profile alignment.

Heuristic

- Column once aligned, will not change later when new sequences are added

can handle < 1,000 sequences

Algorithm: T-COFFEE

can handle < 10,000 sequenece

Iterative Approach

- MUSCLE (**Mu**ltiple **S**equence **C**omparison by **L**og-**E**xpectation)

http://www.ebi.ac.uk/Tools/msa/muscle/

Faster and more accurate

Stage 1: builds a guide tree based on fast scoring (k-mer counting)

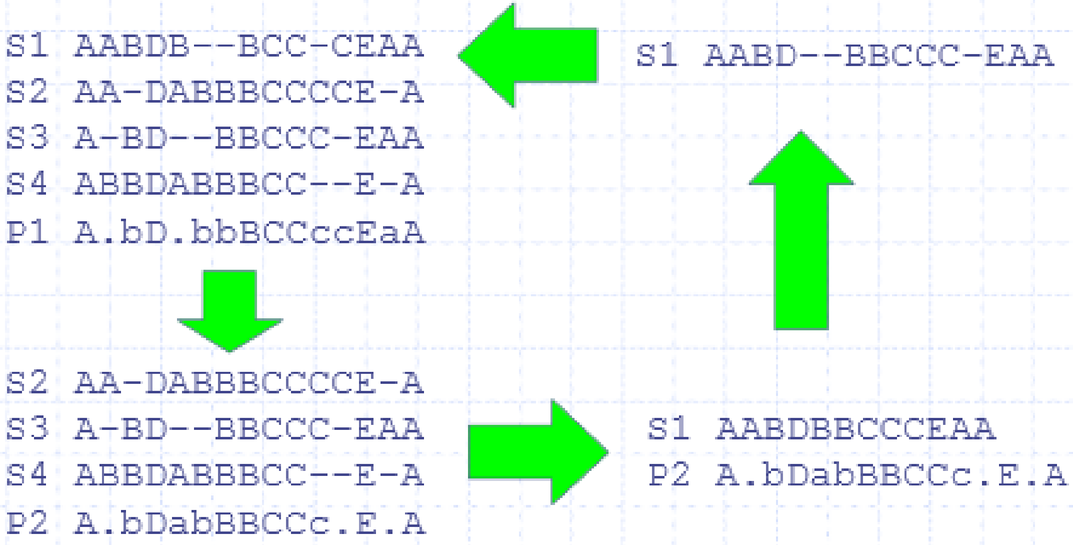
Stage 2: improves the tree through iterative improvements of distance measures

Stage 3: improves MSA through iterative profile-alignment of tree fragments to maximize SP score.

Iterative Techniques [Barton Sternberg 87]

- Key Idea: use profile to optimize MSA
- Input: MSA
- Iterate the following process until convergence:
  - Select a sequence X_k compute profile of the other sequences
  - Align Xk against this profile to create new MSA

- Example:



```
S1 AABDB--BCC-CEAA
S2 AA-DABBBCCCCE-A
S3 A-BD--BBCCC-EAA
S4 ABBDABBBCC--E-A
P1 A.bD.bbBCCccEaA
```

```
S2 AA-DABBBCCCCE-A
S3 A-BD--BBCCC-EAA
S4 ABBDABBBCC--E-A
P2 A.bDabBBCCc.E.A
```

```
S1 AABDBBCCCEAA
P2 A.bDabBBCCc.E.A
```

```
S1 AABD--BBCCC-EAA
```

Credit: Yechiam Yemini (Columbia U)