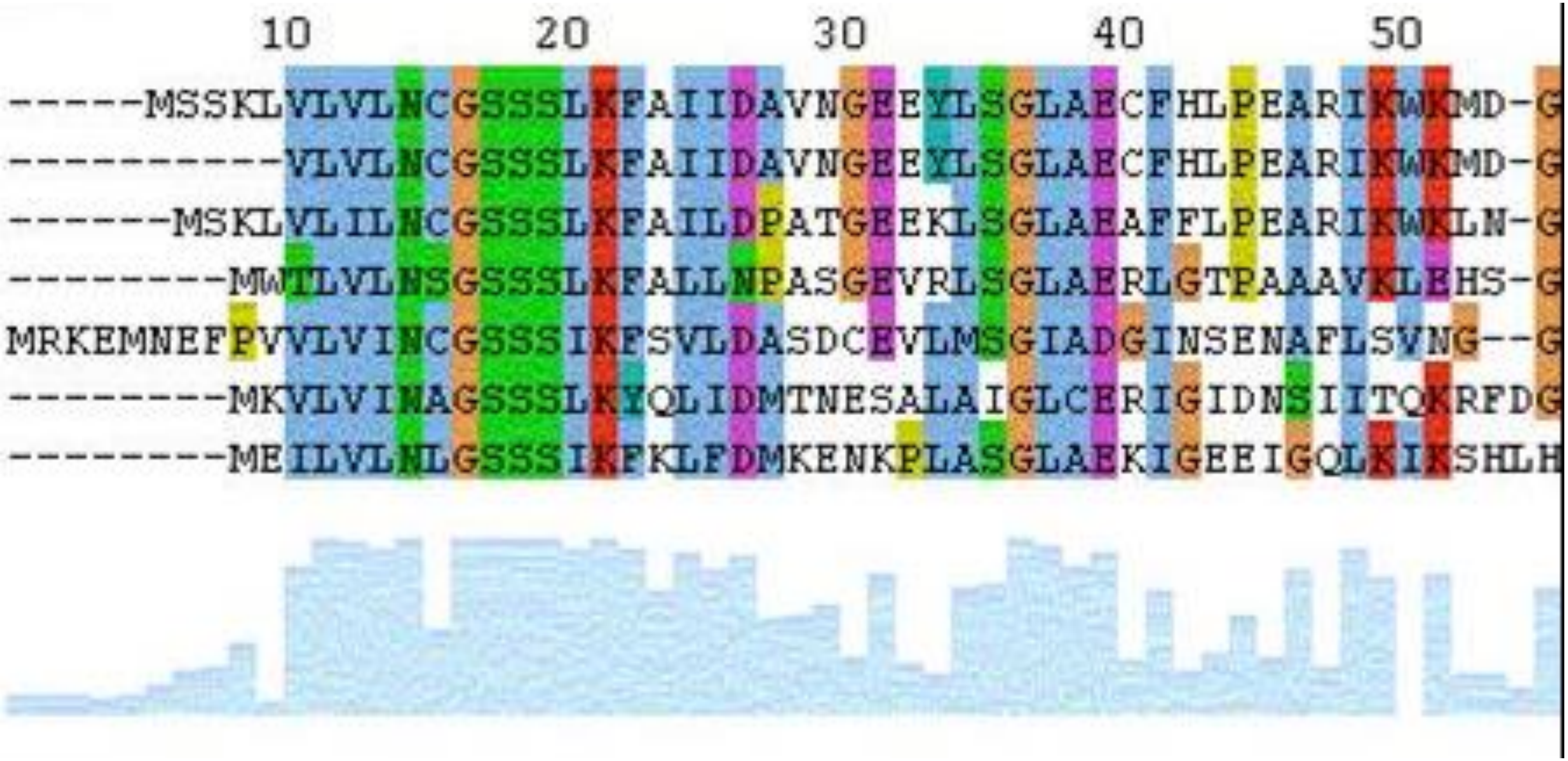
GLOBEX Bioinformatics (Summer 2015) Multiple Sequence Alignment

- 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

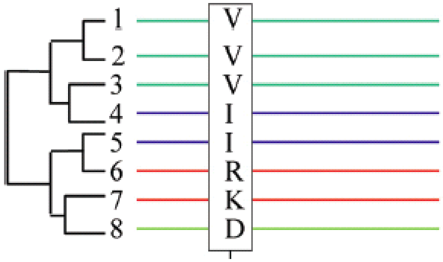
Motivations

- Collective (or aggregate) 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

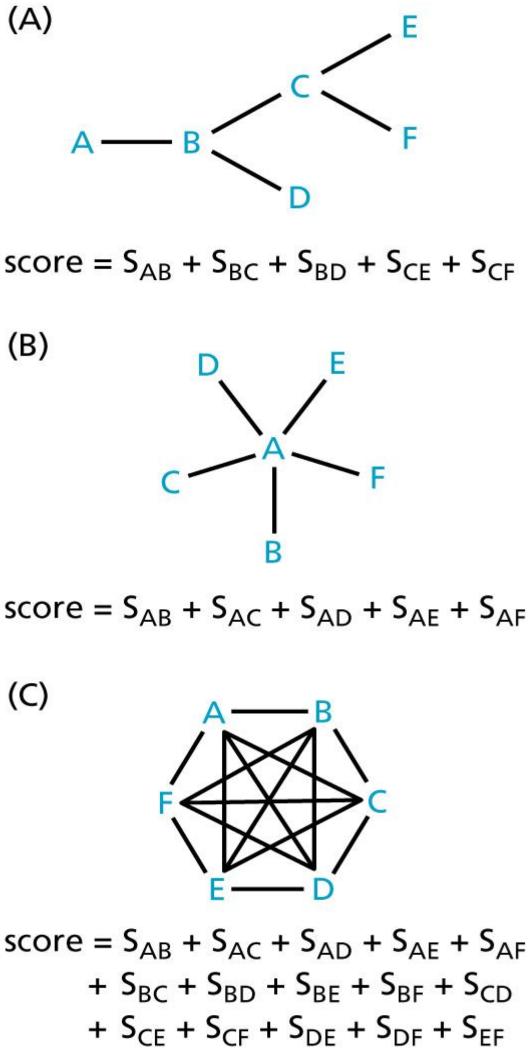
$$S(m_i) = \sum_{k<l} S(m_i^k, m_i^l),$$

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

$$s(a, -) = s(-, a) = -d, \quad s(-,-) = 0 \quad \text{(Once a gap, always a gap)}$$

Common ways to construct alignment score from pairwise scores.



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

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

(C) score = $S_{AB} + S_{AC} + S_{AD} + S_{AE} + S_{AF}$
$+ S_{BC} + S_{BD} + S_{BE} + S_{BF} + S_{CD}$
$+ S_{CE} + S_{CF} + S_{DE} + S_{DF} + S_{EF}$

This is the SP score used in the previous slide

Example of SP scoring

F
F
F
I
V

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

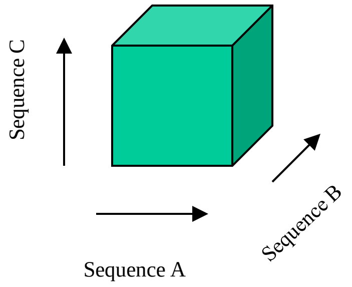
F
F
F
I
N

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

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 matrix 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

```
X:      GAAGTT
Y:      GAC-TT

Z:      GAACTG
W:      GTACTG
```

Alignment (XY) is frozen, even in light of new examples (ZW) that suggest Y: GA-CTT

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

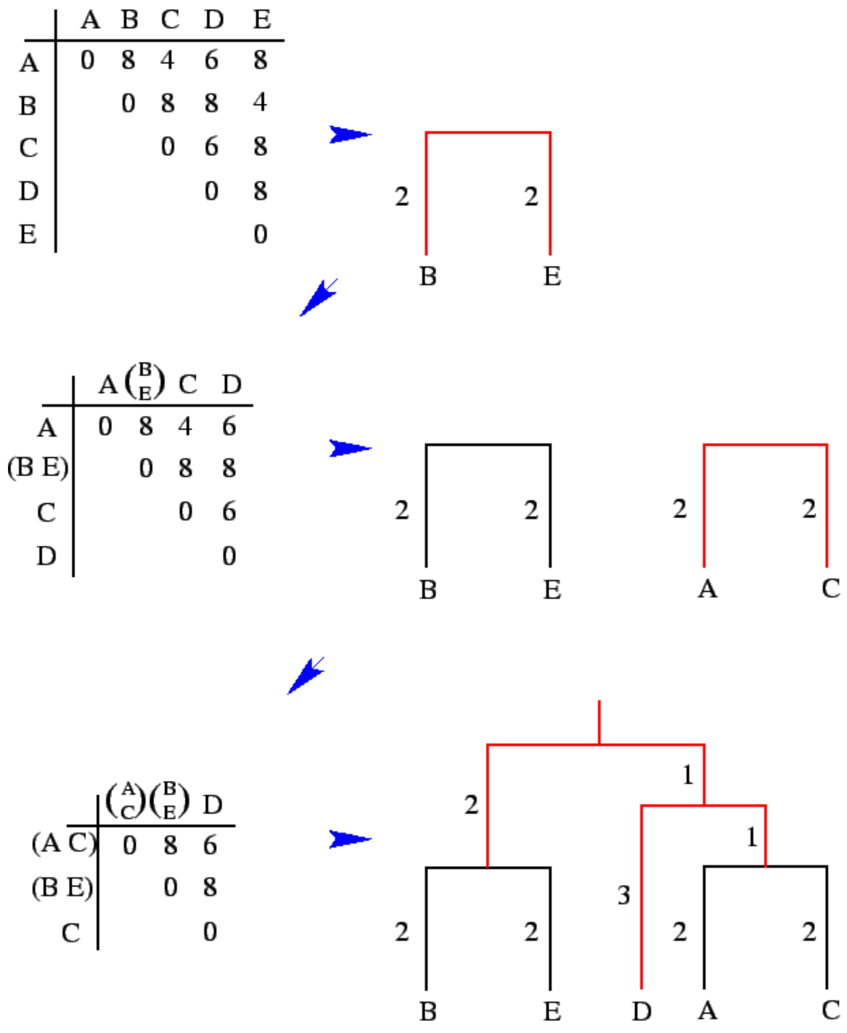
Approach 2: Progressive Alignment

UPGMA

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

Distance d_{ij} between cluster C_i and C_j is defined as:

$$d_{ij} = \frac{1}{|C_i||C_j|} \sum_{p \text{ in } C_i, q \text{ in } C_j} d_{pq},$$



	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	$\binom{B}{E}$	C	D
A	0	8	4	6
(B E)		0	8	8
C			0	6
D				0

	$\binom{A}{C}$	$\binom{B}{E}$	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.

Position Specific Score Matrix (PSSM) and Profile

R06098 \TCA**CACGTG**GGA\

R06099 \GGC**CACGTG**CAG\

R06100 \TGA**CACGTG**GGT\

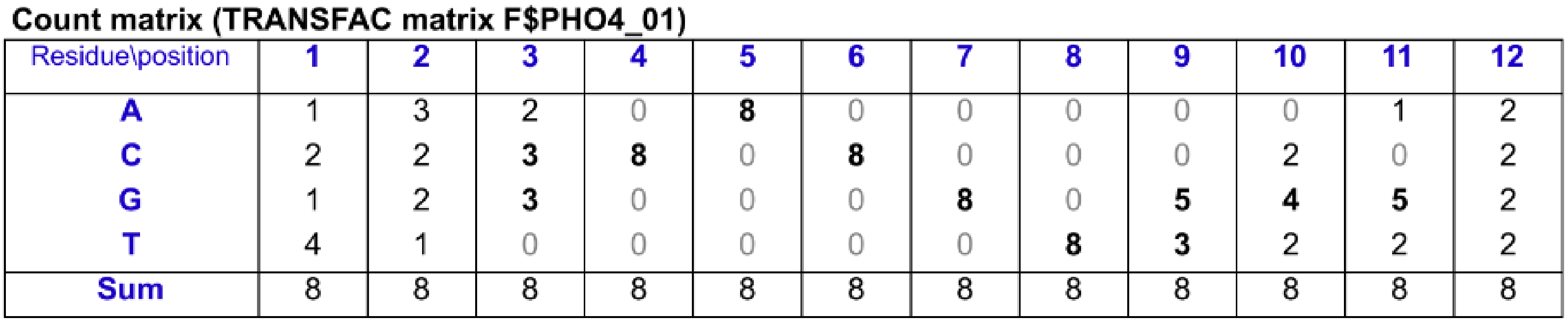
R06102 \CAG**CACGTG**GGG\

R06103 \TTC**CACGTG**CGA\

R06104 \ACG**CACGTT**GGT\

R06097 \CAG**CACGTT**TTC\

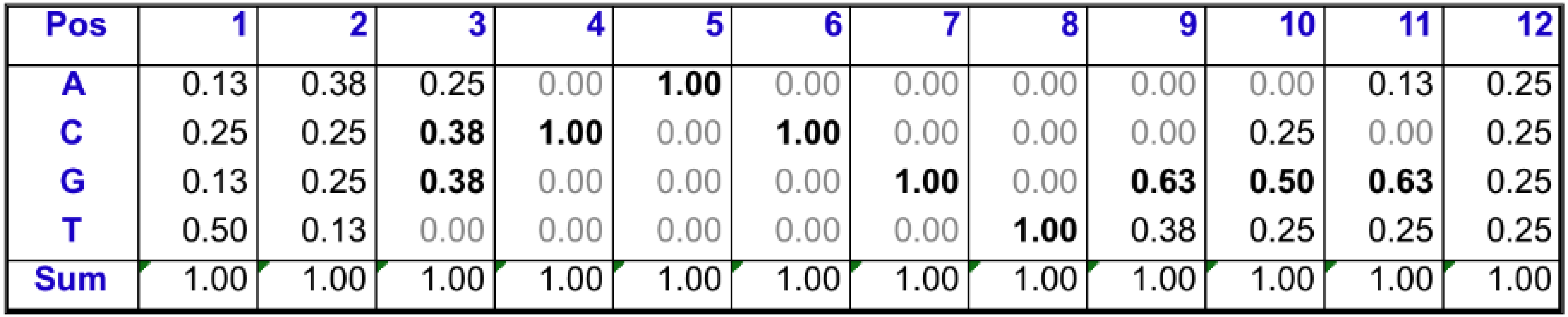
R06101 \TAC**CACGTT**TTC\



Count matrix (TRANSFAC matrix F$PHO4_01)

Residue\position	1	2	3	4	5	6	7	8	9	10	11	12
A	1	3	2	0	**8**	0	0	0	0	0	1	2
C	2	2	**3**	**8**	0	**8**	0	0	0	2	0	2
G	1	2	**3**	0	0	0	**8**	0	**5**	**4**	**5**	2
T	4	1	0	0	0	0	0	**8**	**3**	2	2	2
Sum	8	8	8	8	8	8	8	8	8	8	8	8

PSSM

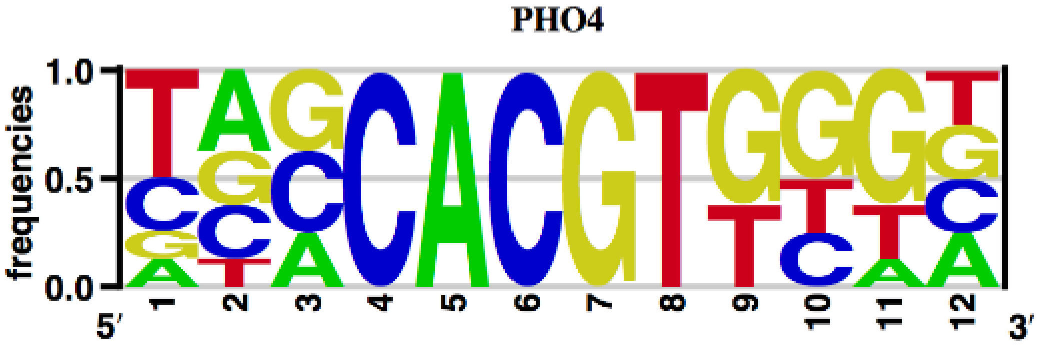


Pos	1	2	3	4	5	6	7	8	9	10	11	12
A	0.13	0.38	0.25	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.13	0.25
C	0.25	0.25	**0.38**	**1.00**	0.00	**1.00**	0.00	0.00	0.00	0.25	0.00	0.25
G	0.13	0.25	**0.38**	0.00	0.00	0.00	**1.00**	0.00	**0.63**	**0.50**	**0.63**	0.25
T	0.50	0.13	0.00	0.00	0.00	0.00	0.00	**1.00**	0.38	0.25	0.25	0.25
Sum	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

$$f_{i,j} = \frac{n_{i,j}}{\sum_{i=1}^{A} n_{i,j}}$$

A	*alphabet size (=4)*
$n_{i,j,}$	*occurrences of residue i at position j*
p_i	*prior residue probability for residue i*
$f_{i,j}$	*relative frequency of residue i at position j*

Tom Schneider’s sequence logo. http://weblogo.berkeley.edu/logo.cgi



Ref: Hertz (1999) Bioinformatics 15:563-577

Approach 2: Progressive Alignment

- Align a sequence to a profile

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

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

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 column i of P to column 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

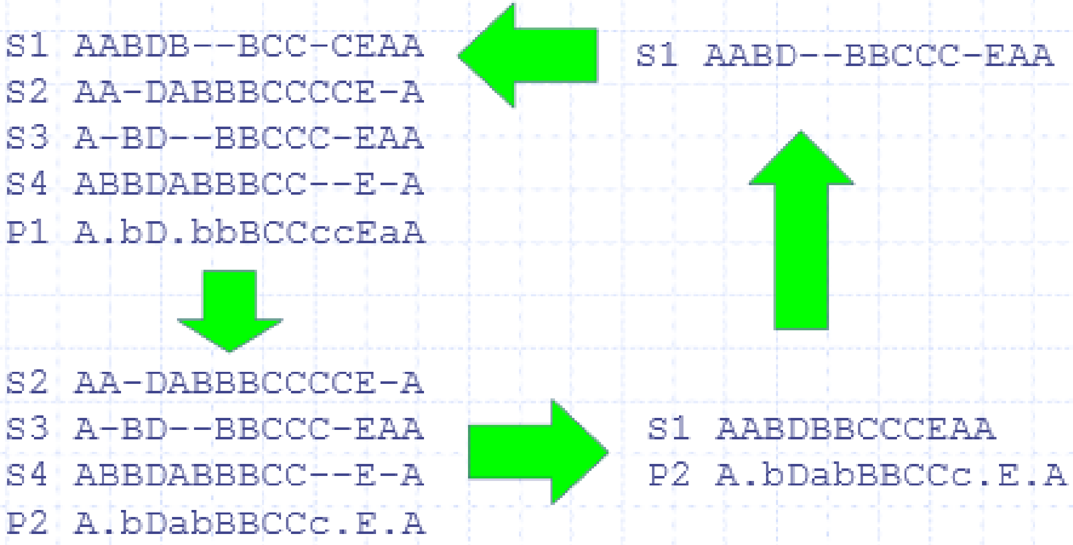
Stage : builds a guide tree based on fast scoring (k-mer counting)

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
```

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

```
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
```

Credit: Yechiam Yemini (Columbia U)