Multiple sequence alignment

BCB410 presentation by Nirvana Nursimulu

Friday 25th November 2011

MSA: definition

- In MSA, *k* (greater than 2) sequences are aligned at the same time.
- Sequences can be of DNA, RNA, or protein.
- Want to write each sequence along the others to express any similarity between the sequences.

MSA: motivation

- Reveal biologically important sequence similarities.
  - These may be dispersed or hidden within sequences.
- Phylogenetic reconstruction.
  - Can obtain evolutionary history of respective sequences.

MSA: motivation

- Secondary structure prediction by homology modeling.
  - Structure of a protein is uniquely determined by its amino acid sequence.
  - During evolution, structure is more stable than sequence.

MSA versus Pairwise Sequence Alignment

- Can't we just do a number of pairwise sequence alignments?
- Needleman-Wunsch algorithm: uses dynamic programming (for 2 sequences, ie, pairwise sequence alignment)

MSA versus Pairwise Sequence Alignment

- Formulation of recursion for sequences A and B ($\delta<0$ is the gap penalty)

$$F(i,j)=\max\begin{cases}F(i-1,j-1)+S(A_i,B_j)\\F(i,j-1)+\delta\\F(i-1,j)+\delta\end{cases}$$

$$F(0,i)=i\cdot\delta$$

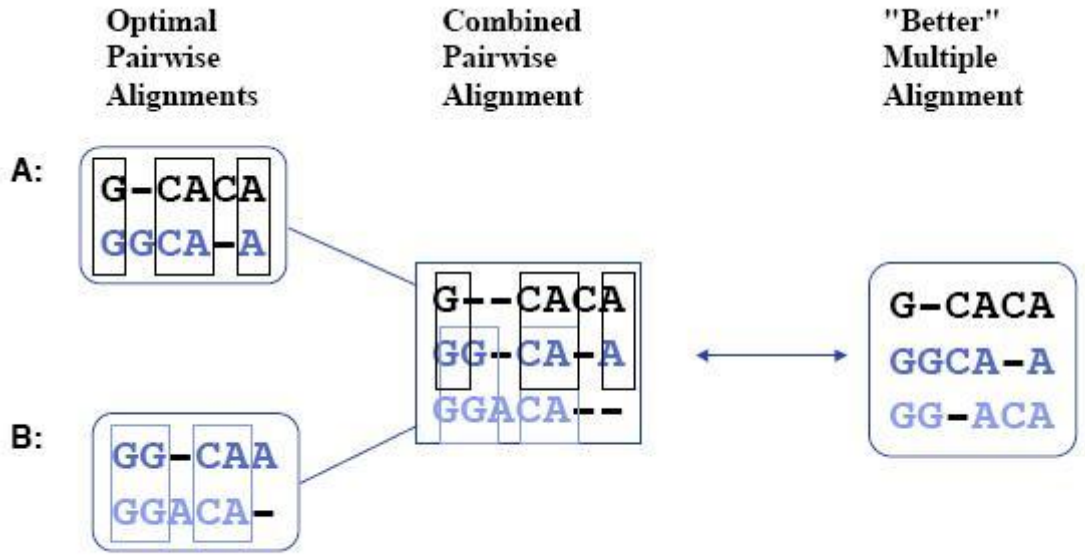
$$F(j,0)=j\cdot\delta$$

MSA versus Pairwise Sequence Alignment

- Time complexity is $O(L^2)$ for a pair
  - L is the length of the longer sequence.
- If we perform multiple pairwise sequence alignment to get an MSA: $O(k.L^2)$.
  - k is the number of sequences.
  - L is the length of the longest sequence.

…but:

- Does this *actually* work!?!? NO!



Source: BCH441H fall 2011 notes.

- "Better" has fewer gaps + more matches

Therefore:

Proper MSA algorithm needs to consider all the sequences, not just two at a time!

Naïve implementation of MSA

- Could use dynamic programming to get optimal solution (For more details see R. Durbin: 141-142)
- Takes $O(L^k)$
  - k is the number of sequences.
- This takes exponential time…

→Need to use heuristic methods instead.

Tools:

- ClustalW
- T-coffee
- MAFFT
- MUSCLE

MSA tools

- Different strategies.
- One objective usually:
  - Maximize sum of scores of all pairwise alignments.

MSA strategies

- Progressive
  - Objective: align by phylogeny
  - align most similar first, then merge together
- Consistency-based
  - Objective: retain conserved regions
  - conserved regions guide alignment

MSA strategies

- Probabilistic
  - Objective: maximize similarity to model
  - Create a model + align each sequence to that

- Iterated
  - Objective: find important regions + extend alignment from secure seeds
  - Improve alignment from draft alignments

ClustalW

ClustalW: command-line interface

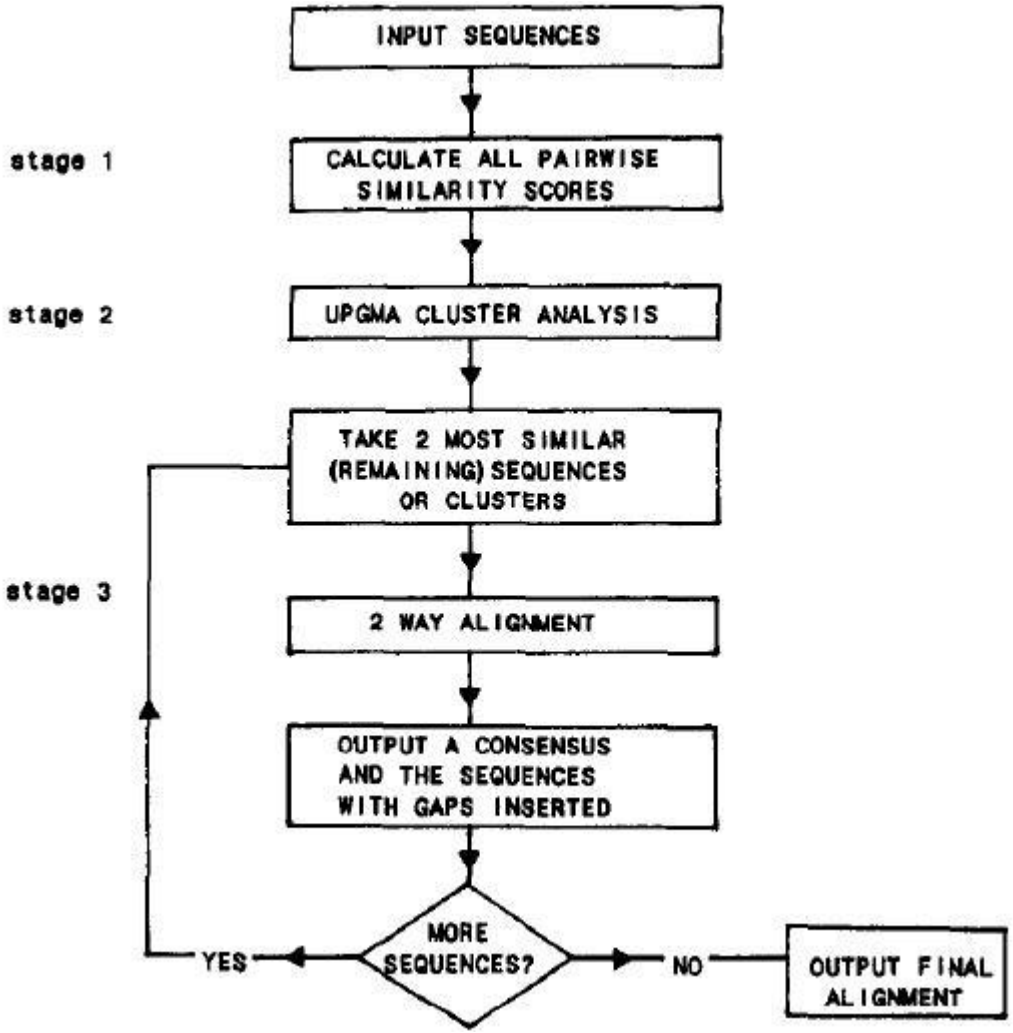
ClustalX: GUI

- Clustal has been in use for the longest time amongst all tools.
  - “Old is gold”?!?

ClustalW: progressive MSA

- 3 stages:
  - Calculation of all pairwise sequence similarities
  - Construction of a guide tree from the similarity matrix built by initial step
  - Multiple alignment in a pairwise manner, following order of clustering in guide tree
- Finally, align according to guide tree

ClustalW: progressive MSA



(Higgins D.G., Sharp P.M.: figure 1)

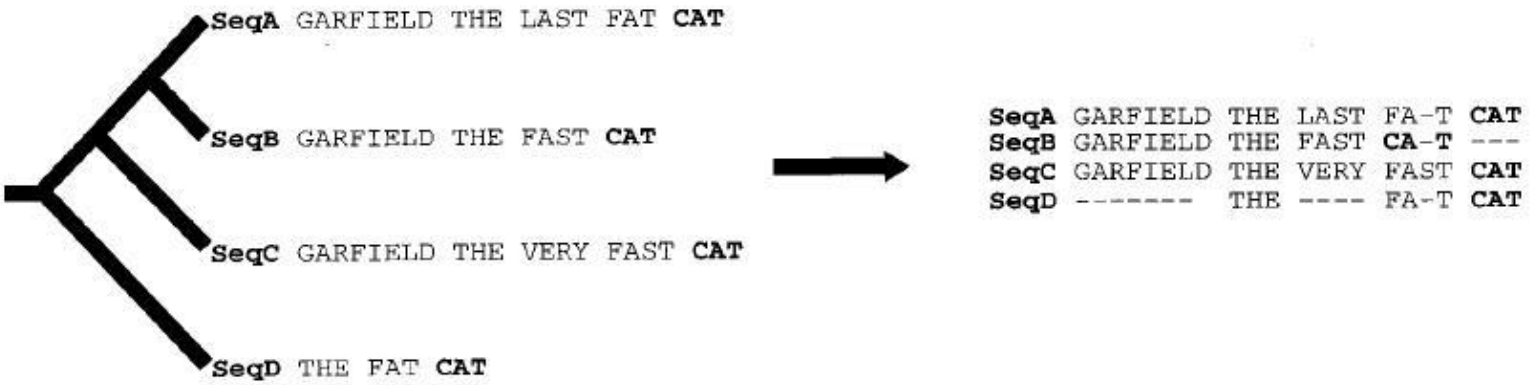
ClustalW: progressive MSA

- UPGMA cluster analysis
  - **U**nweighted **P**air **G**roup **M**ethod with **A**rithmetic Mean.
  - Assumes a constant rate of evolution.
  - Iteratively joins the two nearest clusters, until one cluster is left.
  - Distance between clusters A and B = mean distance between elements of each cluster

ClustalW: key limitation

- Errors early-on persist
- Performance deteriorates for multidomain protein and distant similarities
  - Works best when gap-poor, globally alignable
  - …but these are uninteresting!

ClustalW: example error



Notredame C., Higgins D.G., Heringa J.: figure 2(a)

"CAT" is misaligned here.

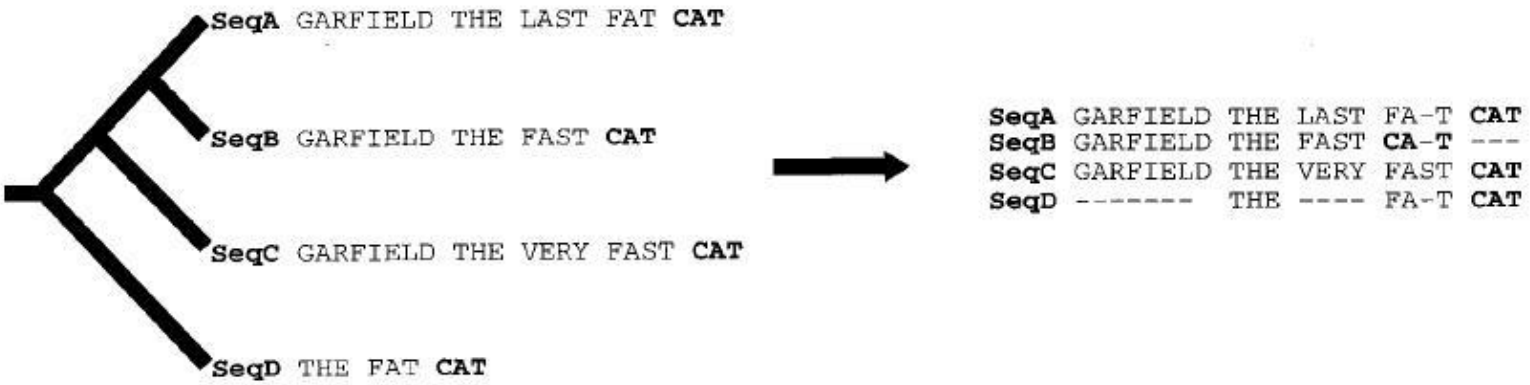
T-coffee: consistency-based

- **T**ree-based **C**onsistency **O**bjective **F**unction **F**or alignment **E**valuation
- Two attractive features:
  - Can use heterogeneous data sources to generate MSA
    - Data from these sources provided via a library of pairwise alignments
  - Optimization method finds the MSA that best fits the pairwise alignments (in library)

T-coffee: consistency-based

- Technique is similar to Clustal's
  - Greedy progressive strategy
- But different and better
  - Consider information from **all** the sequences during each alignment step
    - …not just those being aligned at that stage

Recall, with ClustalW…



Notredame C., Higgins D.G., Heringa J.: figure 2(a)

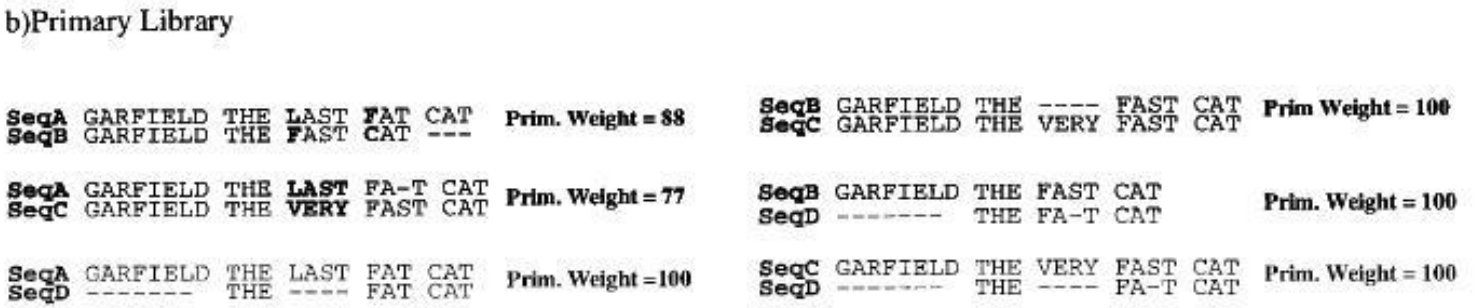
"CAT" is misaligned here.

T-coffee: algorithm

- Creation of a primary library
  - Construct global pairwise alignments for all the sequences (can use ClustalW)
  - Compute top ten non-intersecting local alignments between each pair of sequences (using Lalign)
  - Weighting of pairwise alignments
    - Weight of each pair of residue = average identity amongst matched residues

T-coffee: primary library example

- Combine local and global alignment libraries
  - If find duplicated pair between the 2 libraries: merge into a single entry
    - Weight = sum of the 2 weights
  - Otherwise, new entry created.

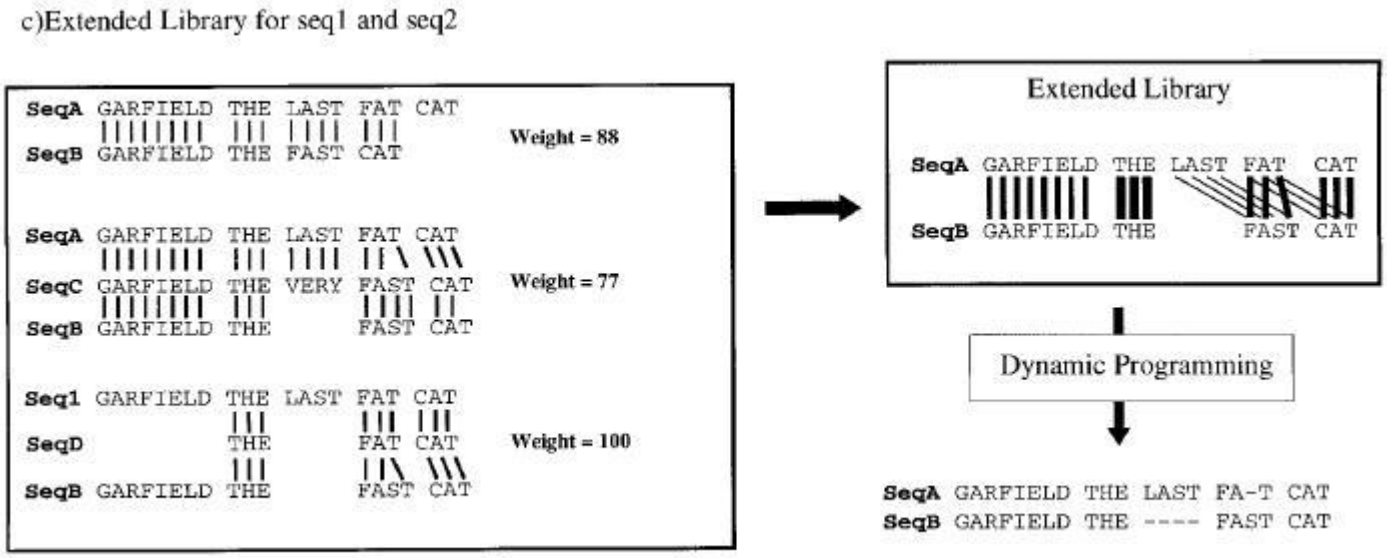


Notredame C., Higgins D.G., Heringa J.: figure 2(b)

T-coffee: algorithm

- Extended library: triplet approach
  - For each aligned residue pair(a,b) in library:
    - Check alignment of (a,b) with residues from remaining sequences
    - More intermediate seq. supporting alignment ➔ higher weight
  - When all included pairwise alignments are totally inconsistent: $O(N^3L^2)$
    - N = num. sequences; L = average seq. length
  - In practice: $O(N^3L)$

T-coffee: extended library example

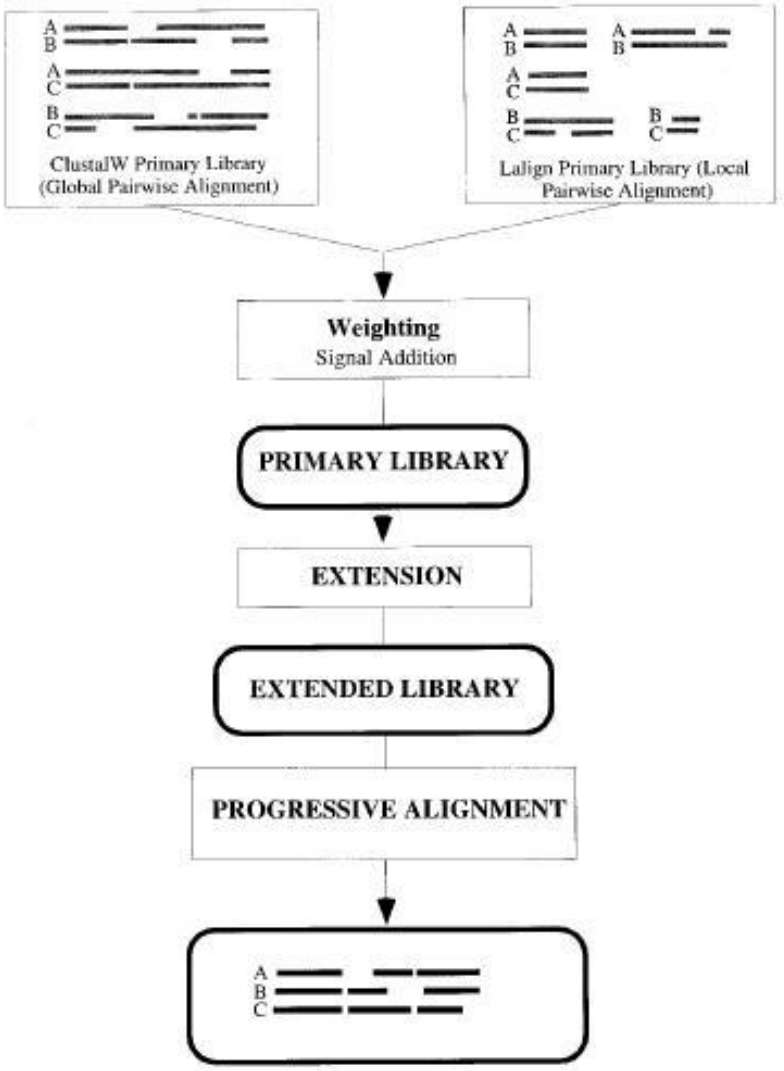


Notredame C., Higgins D.G., Heringa J.: figure 2(c)

T-coffee: algorithm

- Progressive alignment
  - Produce guide tree
  - Use the same strategy as was used with Clustal…
    - …but use the weights in the extended library to align the residues

T-coffee: summary



Notredame C.,
Higgins D.G.,
Heringa J.: figure 1

T-coffee versus Clustal

- Takes info from local alignments in consideration
- More accurate
  - A bit slower

MAFFT: algorithm

- **M**ultiple **A**lignment using **F**ast **F**ourier **T**ransform.
- Amino acid residues are converted to vectors of volume and polarity
- Intuition:
  - Substitutions between physico-chemically similar amino acid tend to preserve the structure of proteins.

MAFFT: algorithm

- Note:
  - Can also use with nucleotide bases:
  - Convert to vectors of imaginary and complex numbers
  - But, here, will focus with amino acids.

MAFFT: algorithm

- Find correlation (of volume and polarity components) between two sequences.

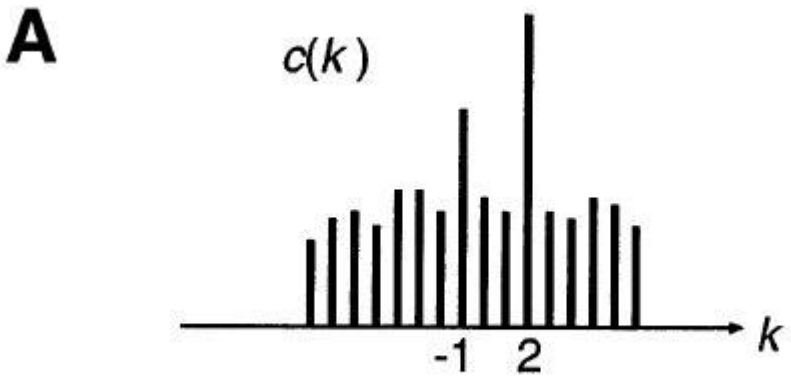
$$c(k) = c_v(k) + c_p(k)$$

$$c_v(k) = \sum_{1 \le n \le N, 1 \le n+k \le M} \hat{v}_1(n)\hat{v}_2(n+k)$$

$$c_p(k) = \sum_{1 \le n \le N, 1 \le n+k \le M} \hat{p}_1(n)\hat{p}_2(n+k)$$

- FFT trick reduces the complexity of finding this to O(Nlog N) from $O(N^2)$.

MAFFT: example FFT result

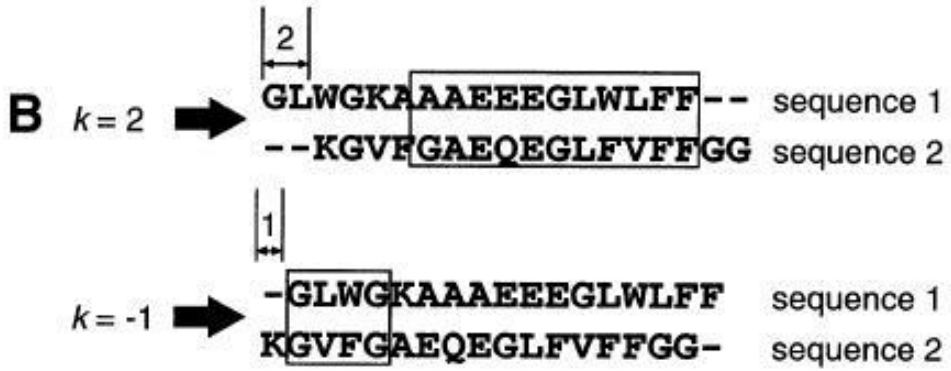


Katoh K., Misawa K., Kuma K., Miyata T.: fig 1(A)

peaks ➔ high correlation ➔ homologous regions

MAFFT: algorithm

- Having performed FFT analysis, we don't know the positions of homologous regions.
- Therefore, perform sliding window analysis:



Katoh K., Misawa K., Kuma K., Miyata T.: fig 1(B)

MAFFT: algorithm

- Construct homology matrix, S:
  - If the ith homologous segment on sequence 1 corresponds to the jth homologous segment on sequence 2, S[i, j] has score value of homologous segment.
  - Otherwise, S[i, j] = 0
- Therefore, matrix is divided into sub-matrices.
- Area for DP is reduced!

MAFFT: homology matrix example

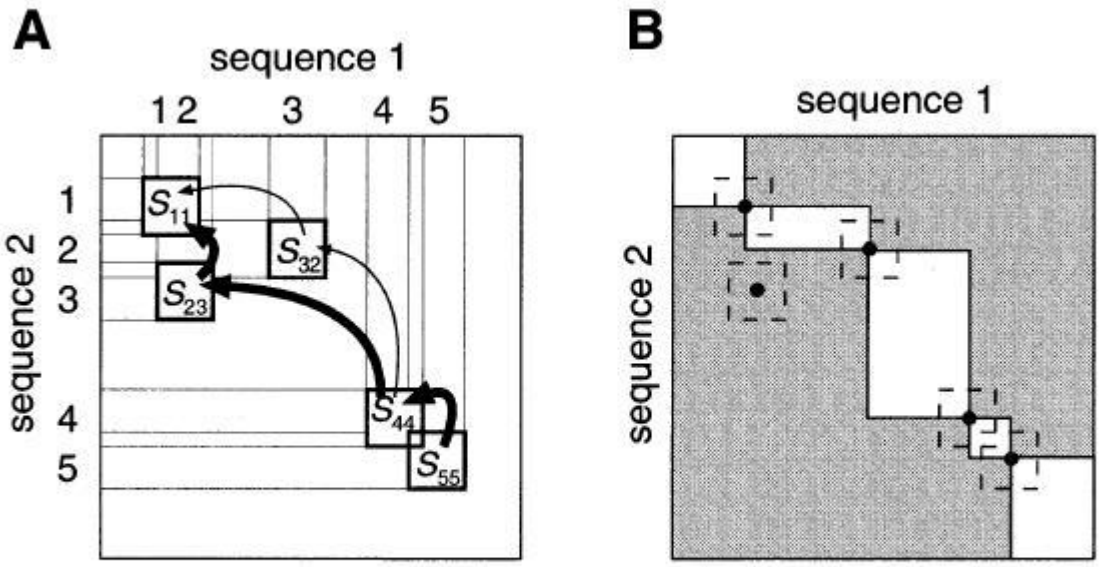


Figure 2. (**A**) An example of the segment-level DP; (**B**) Reducing the area for DP on a homology matrix.

Katoh K., Misawa K., Kuma K., Miyata T.: fig 2(A),(B)

MAFFT: algorithm

- But we have only been talking of 2 sequences…
- Eventually, the MAFFT is only a progressive method (recall: Clustal).
- But it uses a two-cycle progressive method: FFT-NS-2
  - Calculate rough one, then, from this, a refined one is found.

MAFFT: algorithm

- But Clustal had a problem:
  - A gap incorrectly introduced at a step is never removed later.
- Two ways of dealing with this:
  - Iterative refinement method
    - Correct mistakes in initial alignment
  - Consistency-based method
    - Try to avoid mistakes in advance
- Both work equally well.

MAFFT: time complexity

- $O(N^2L) + O(NL^2)$
  - L = sequence length
  - N = number of sequences
- But when input sequences are highly similar: $O(N^2L) + O(NL) = O(N^2L)$ because of FFT-based alignment method

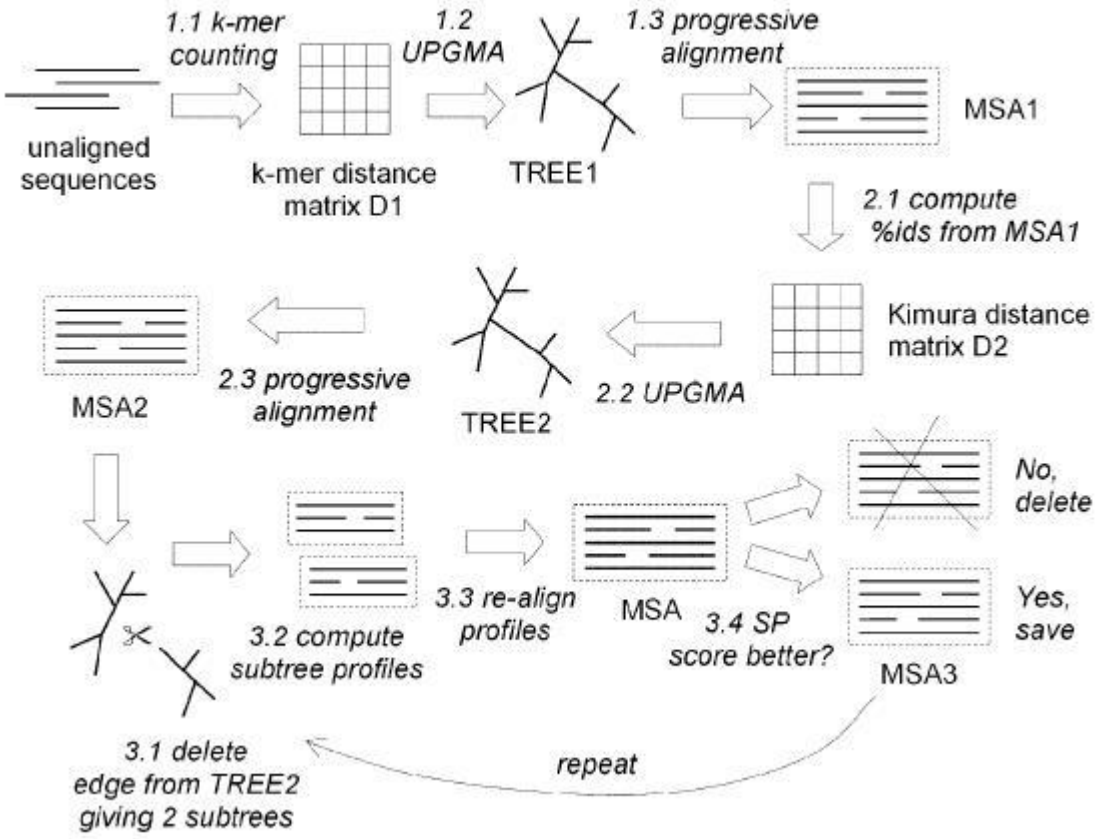
MUSCLE: algorithm

- **MU**ltiple **S**equence **C**omparison by **L**og-**E**xpectation
- Even without refinement:
  - Average accuracy similar to T-coffee and MAFFT
  - Fastest!

MUSCLE: algorithm

- Uses:
  - Progressive draft alignment
  - Iterated improvement

MUSCLE: program flow



Edgar R.C.: fig 2

MUSCLE: algorithm

- 3 main stages:
  - Stage 1: Draft progressive
    - Progressive alignment
  - Stage 2: Improved progressive
    - Progressive alignment
  - Stage 3: Refinement
    - Iterative refinement
- First two stages = MUSCLE-p
- Profile calculated uses log-expectation score

MUSCLE: algorithm

- Stage 1: Draft progressive
  - Goal: produce a MSA, emphasis on speed rather than accuracy
  - Approximate kmer distance used:
    - Derived from fraction of kmers in common in compressed alphabet
  - Result: get TREE1
  - Visit in prefix order, and give a new profile to internal node A from pairwise alignment of A's children profiles ➔ MSA1

MUSCLE: algorithm

- Stage 2: Improved Progressive
  - Goal: re-estimates the first tree using Kimura distance
    - Apply Kimura correction for multiple substitutions at a single site.
  - Result: get TREE2, and MSA2:
    - Optimize by computing alignments only for subtrees whose branching orders changed relative to TREE1.

MUSCLE: algorithm

- Stage 3: Refinement
  - Until convergence or until user-defined limit is reached:
    - Choose an edge e (visit in order of decreasing distance from root)
    - Delete e to get two subtrees: T1, T2.
    - Compute profiles of T1 and T2.
    - Realign profiles to get a new MSA.
    - If score is better, keep new alignment.

MUSCLE: time complexity

- MUSCLE-p (ie, first two stages)
  - Time complexity: $O(N^2L + NL^2)$
  - Space complexity: $O(N^2 + NL + L^2)$
- Refinement
  - Time complexity: $O(N^3L)$
- MUSCLE is comparable in speed with ClustalW.

List of references

- BCH441 Fall 2011 lecture notes on MSA (Lecture 13).
- Durbin R., Eddy S., Krogh A., Mitchison G.: *Biological Sequence Analysis: Probabilistic models of proteins and nucleic acids.* Cambridge University Press, 2002 .
- Edgar R.C.: *MUSCLE: multiple sequence alignment with high accuracy and high throughput.* Nucleic Acids Research. (2004) **32**(5), 1792-1797.
- Higgins D.G., Sharp P.M.: *Clustal—a package for performing multiple sequence alignment on a microcomputer*. Gene. (1988) **73**(1), 237-244.
- Katoh K., Misawa K., Kuma K., Miyata T.: *MAFFT: A Novel Method for Rapid Multiple Sequence Alignment based on Fast Fourier Transform.* Nucleic Acids Research. (2002) **30**(14), 3059-3066.
- Katoh K., Toh H.: Recent developments in the MAFFT multiple sequence alignment program. Briefings in Bioinformatics. (2008) **9**(4), 286-298.
- Notredame C., Higgins D.G., Heringa J.: *T-Coffee: A Novel Method for Fast and Accurate Multiple Sequence Alignment.* J. Mol. Biol. (2000) **302**, 205-217.

Any (more) questions?

- Contact info: nirvana.nursimulu@utoronto.ca