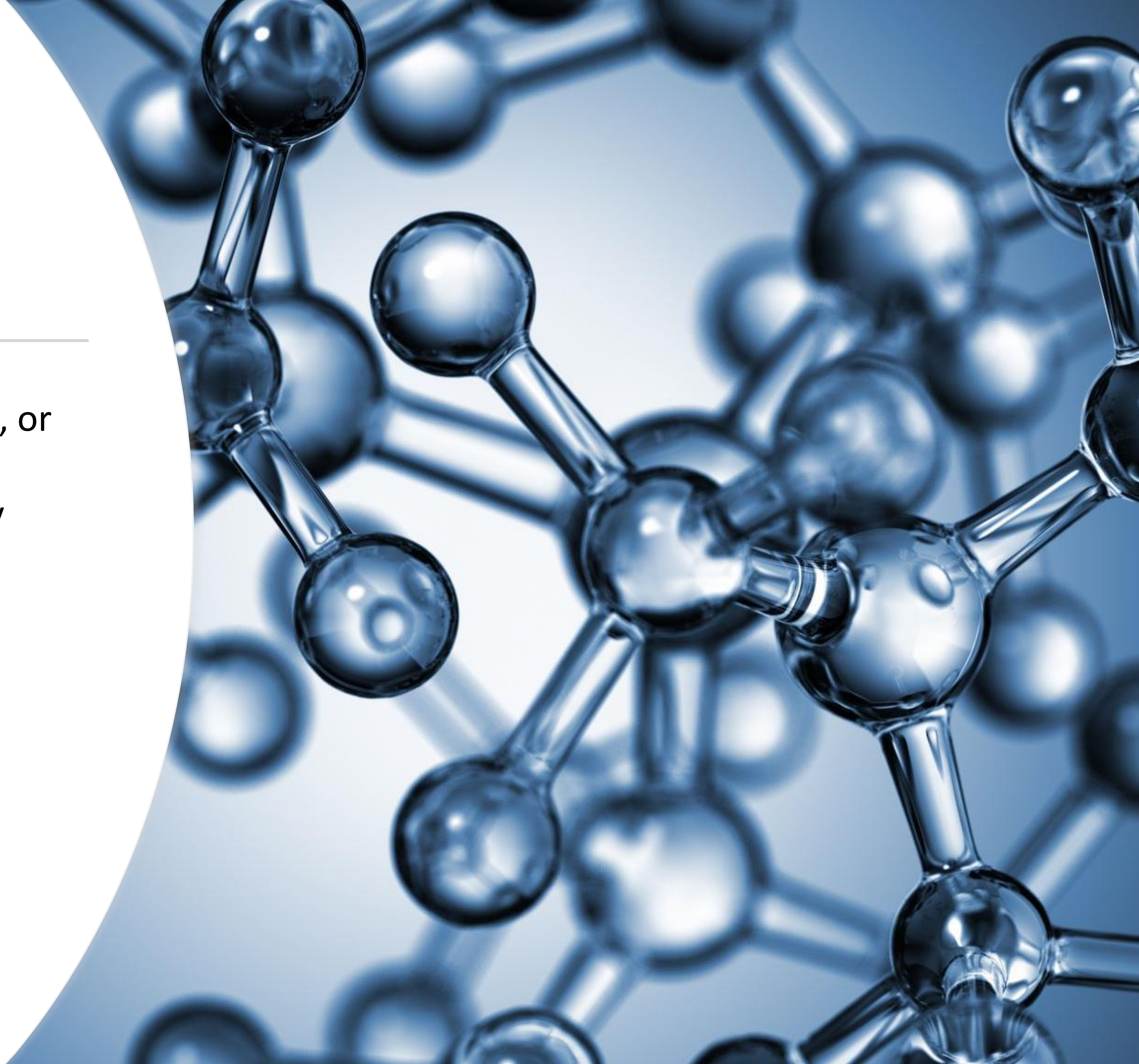
Tutorial 1: Alignments

MAFFT vs. Muscle

By Cameron Calv & Neel Jagad

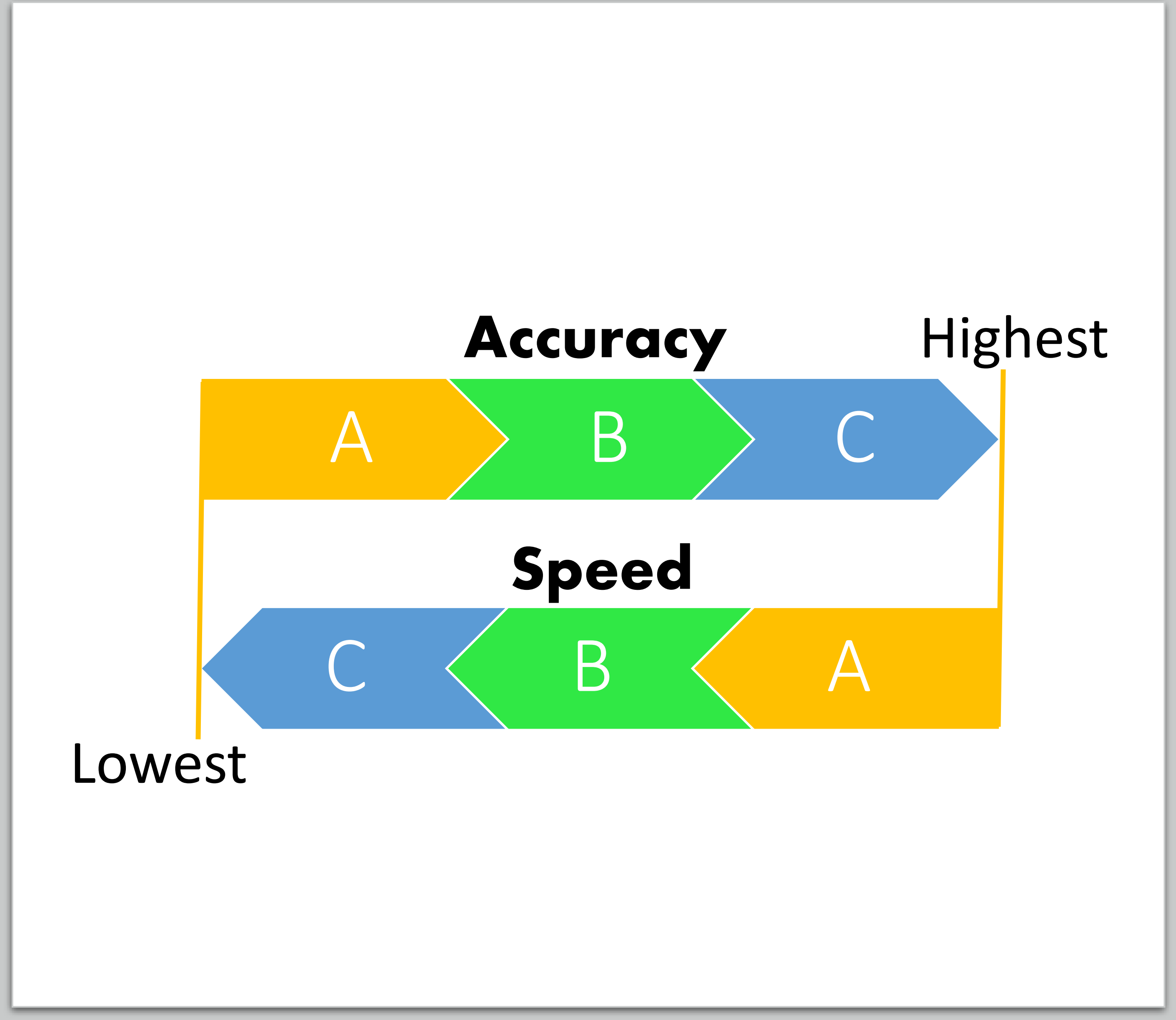
Sequence Alignments

- Arrangement of sequences of DNA, RNA, or protein
  - Used to identify regions of similarity
    - Functional Relationship
    - Structural Relationship
    - Evolutionary Relationship



MAFFT : Multiple Alignment using Fast Fourier Transform

- Split into 3 categories which exchange speed for accuracy
  A. Progressive Method
  B. Iterative Refinement Method with WSP Score
  C. Iterative Refinement Method with WSP and Consistency Scores



A. The Progressive Method

1. Create rough distance matrix between sequences using 6-mer similarity
2. Build UPGMA tree for all sequences
3. Align sequences giving highest priority in order of branch number

1

Create rough distance matrix between sequences using 6-mer similarity

2

Build UPGMA tree for all sequences

3

Align sequences giving highest priority in order of branch number

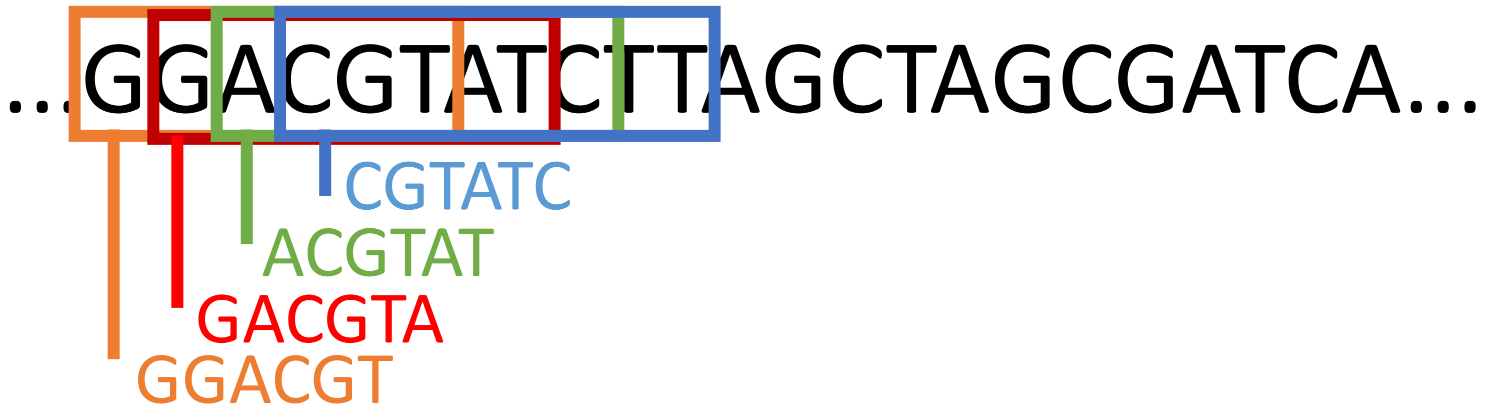
1

Create rough distance matrix between sequences using 6-mer similarity

	A	B	C	D	E	F	G
A							
B	19.00						
C	27.00	31.00					
D	8.00	18.00	26.00				
E	33.00	36.00	41.00	31.00			
F	18.00	**1.00**	32.00	17.00	35.00		
G	13.00	13.00	29.00	14.00	28.00	12.00	

$$D_{ij} = 1 - S_{ij} / \min(S_{ii}, S_{jj})$$

- Uses a distance calculated using 6-mer counting
  - How many times does a sequence of 6 base pairs occur in a row?
  - Number of shared 6-mers between sequences *i* and *j* = S_{ij}



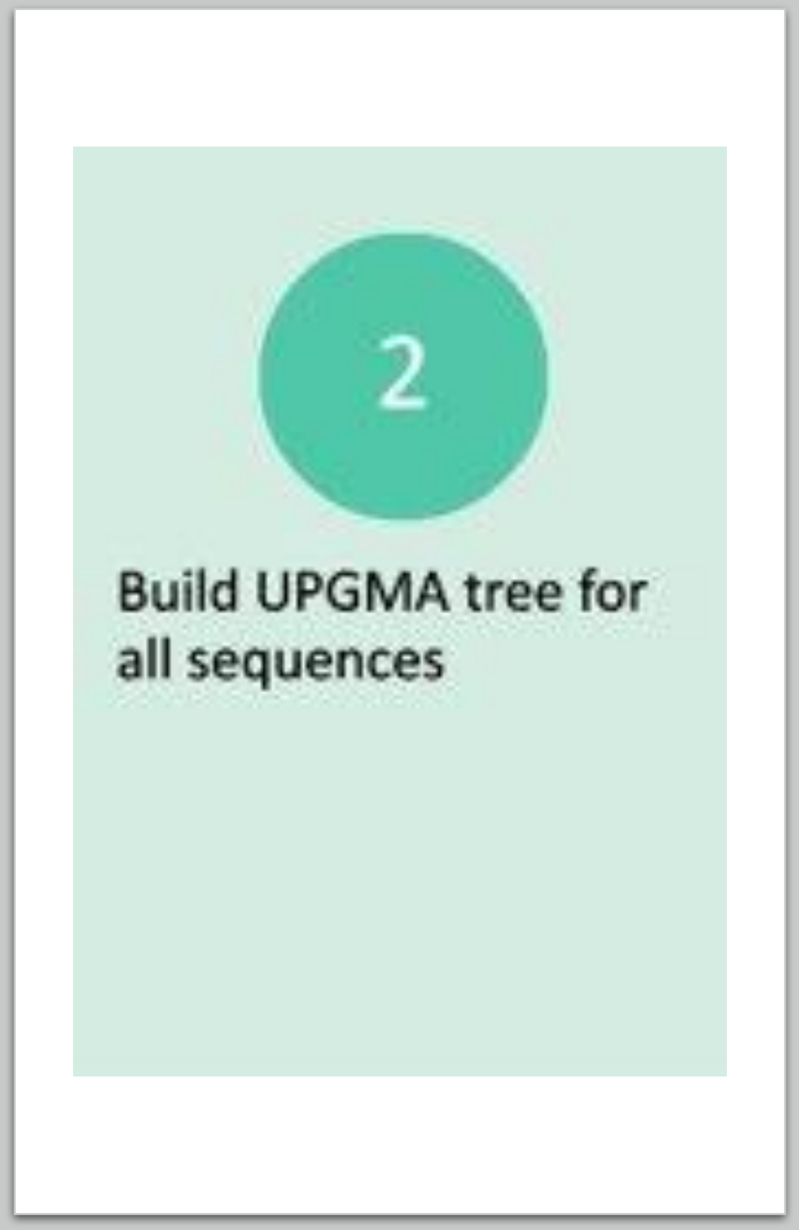
...GGACGTATCTTAGCTAGCGATCA...

CGTATC

ACGTAT

GACGTA

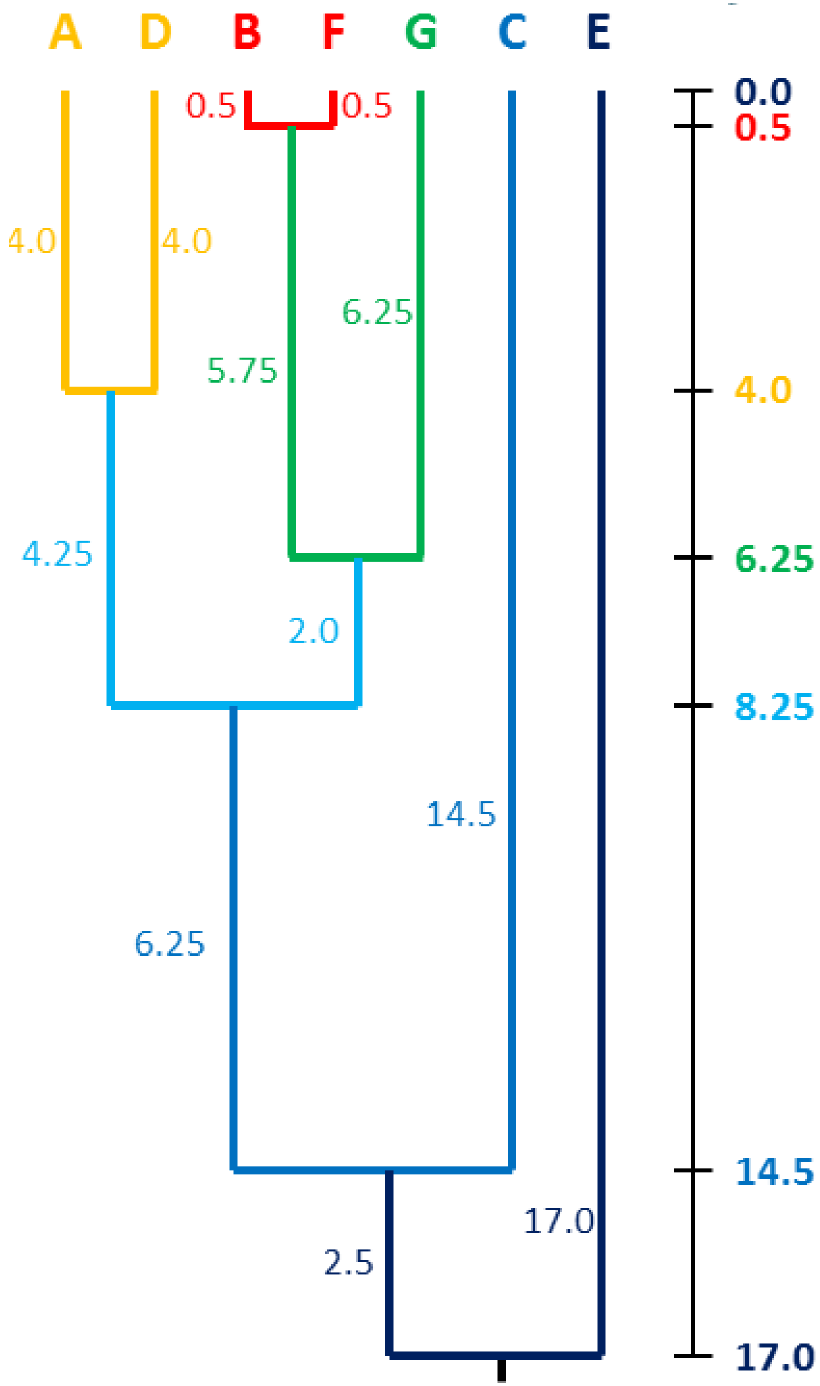
GGACGT



2

Build UPGMA tree for all sequences

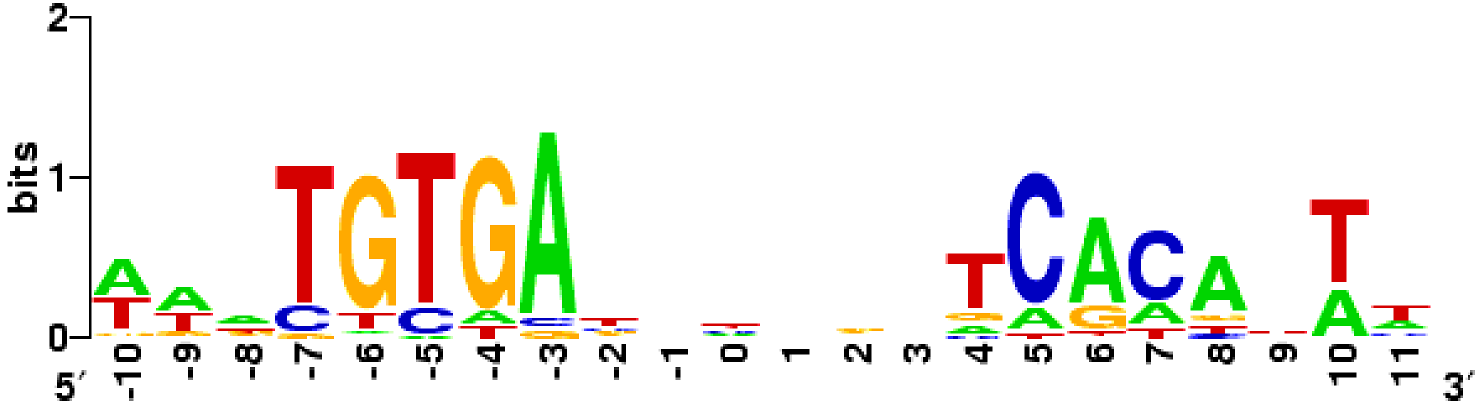
- Use the distance matrix to create an UPGMA (Unweighted Pair-Group Method with Arithmetic mean)
  - Clusters the sequences together based on their relative similarity
  - Computed using the distance matrix calculate in step 1



3

Align sequences giving highest priority in order of branch number

- Use the branching order to align your sequences
  - Sample alignment shown below using WebLogo



A. The Progressive Method (Part II!?)

4

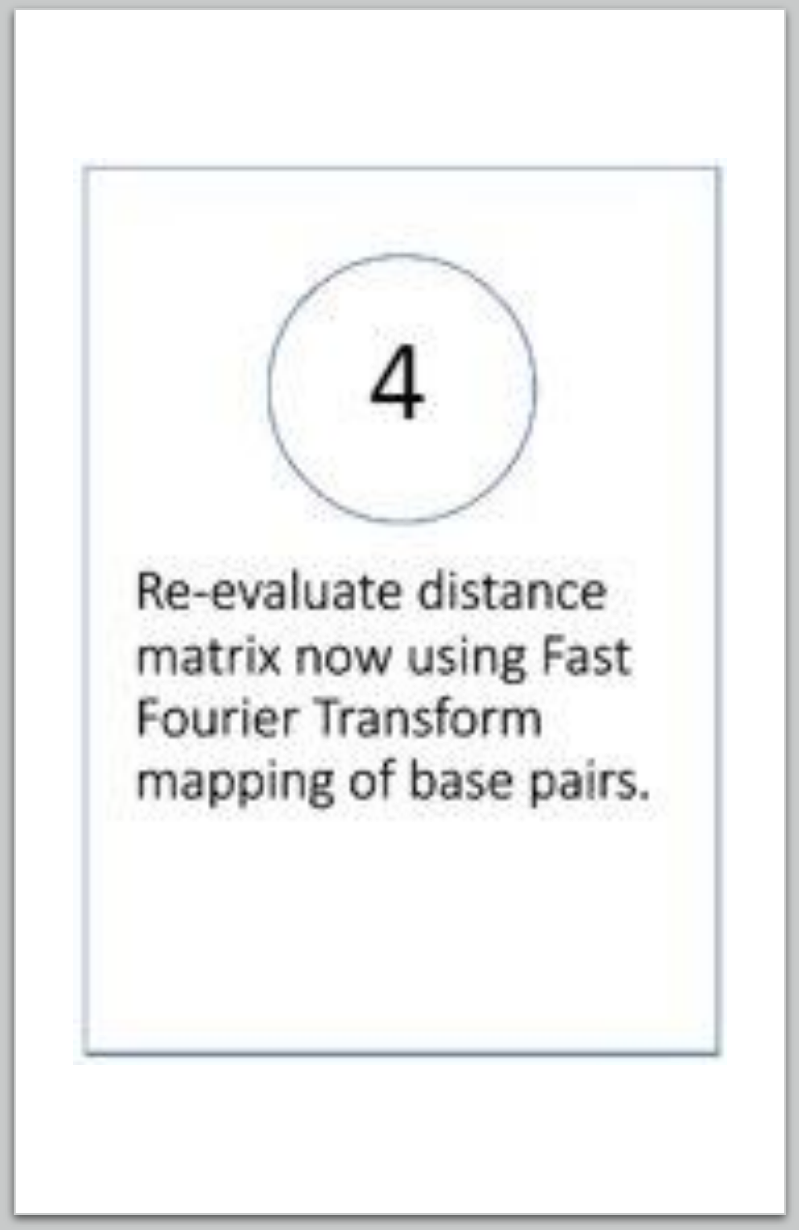
Re-evaluate distance matrix now using Fast Fourier Transform mapping of base pairs.

5

Re-build **UPGMA tree for all sequences**

6

Re-align sequences



Re-evaluate distance matrix now using Fast Fourier Transform mapping of base pairs.

Putting the FFT in MAFFT

- Distance matrix recalculated using this cost function:

$$c(k) = c_v(k) + c_p(k),$$**1**

- Where:

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

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

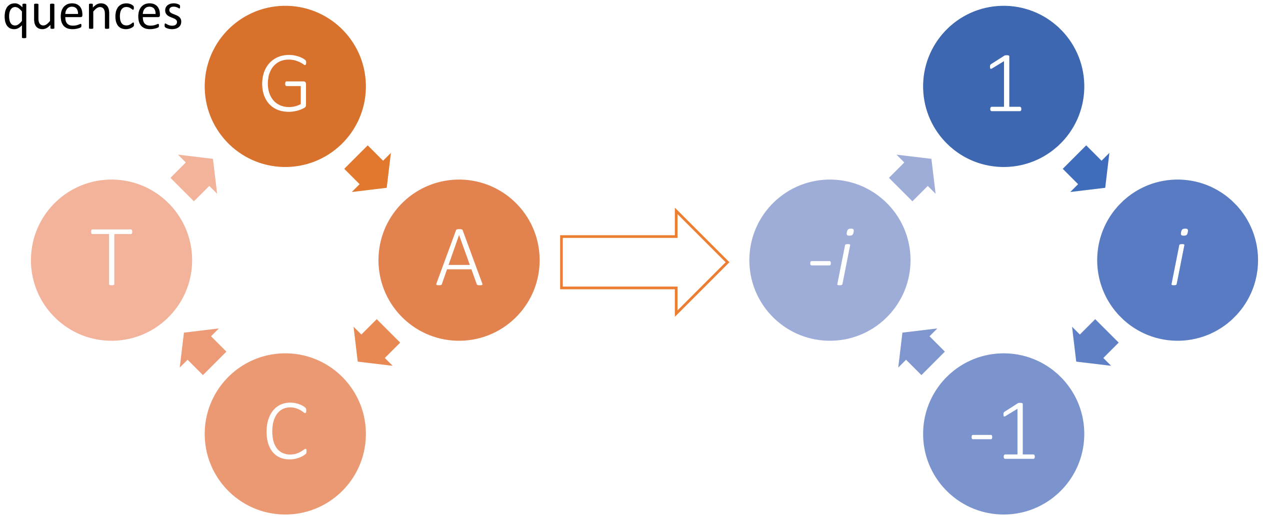
- Normally this calculation would take O(n^2) time, however using the FFT, it can be reduced to **O(*N* log *N*)**

4

Re-evaluate distance matrix now using Fast Fourier Transform mapping of base pairs.

Putting the FFT in MAFFT (continued)

- Map base pairs like so and convert 6-mers to complex sequences



...GGACGTATCTTAGCTAGCGATCA...

CGTATC -- ($-1, 1, -i, i, -i, -1$)

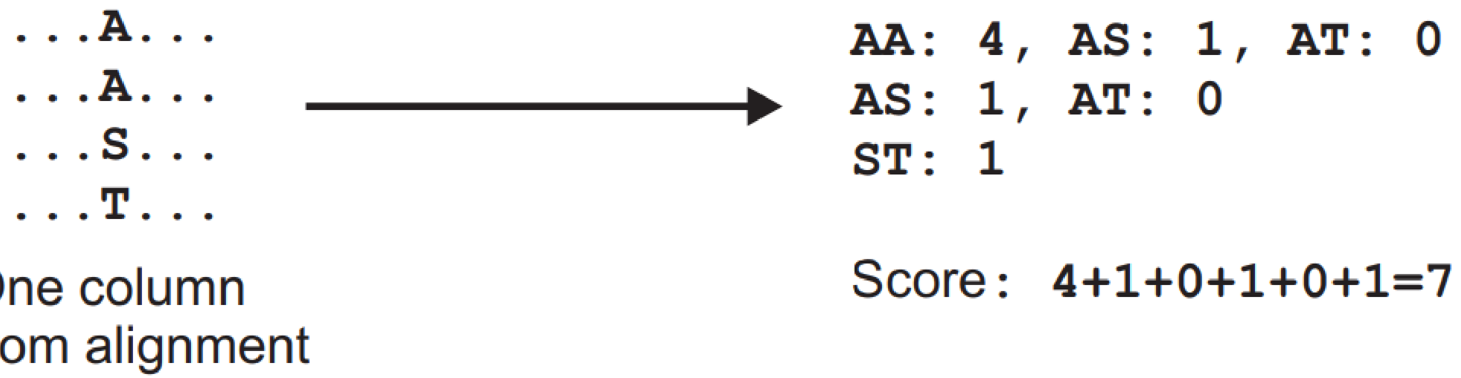
ACGTAT -- ($i, -1, 1, -i, i, -i$)

GACGTA -- ($1, i, -1, 1, -i, i$)

GGACGT -- ($1, 1, i, -1, 1, -i$)

B. The Iterative Refinement Method *with the WSP Score*

- Adds a scoring system to reduce alignment bias
  - SP (Sum-of-Pairs) Score



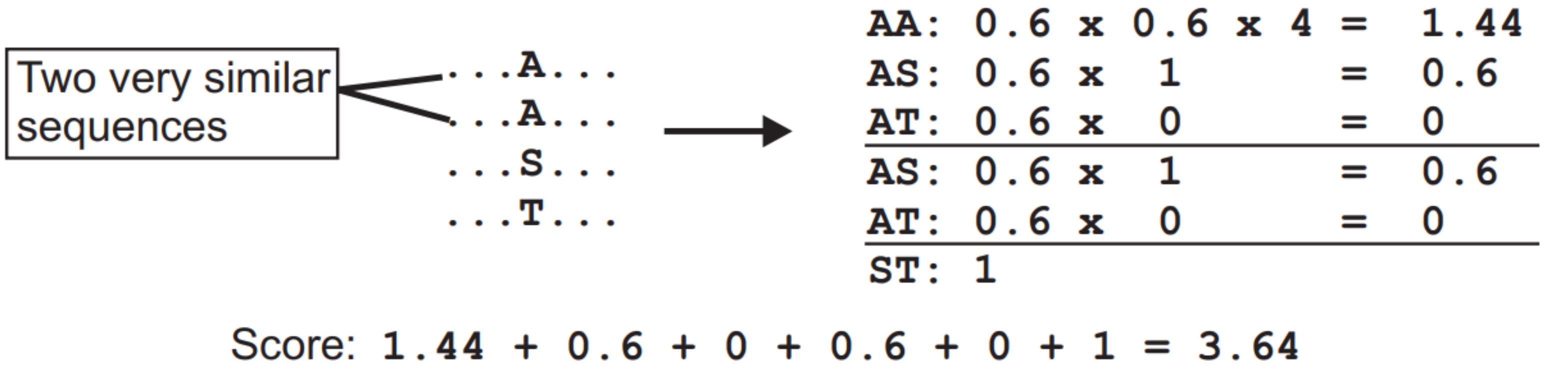
...A...
...A...
...S...
...T...

One column from alignment

AA: 4, AS: 1, AT: 0
AS: 1, AT: 0
ST: 1

Score: 4+1+0+1+0+1=7

  - WSP (Weighted Sum-of-Pairs) Score



Two very similar sequences

...A...
...A...
...S...
...T...

AA: 0.6 x 0.6 x 4 = 1.44
AS: 0.6 x 1 = 0.6
AT: 0.6 x 0 = 0
AS: 0.6 x 1 = 0.6
AT: 0.6 x 0 = 0
ST: 1

Score: 1.44 + 0.6 + 0 + 0.6 + 0 + 1 = 3.64

C. The Iterative Refinement Method with the WSP *and Consistency Scores*

- Reduce bias *even more* with a non-biased scoring system
  - COFFEE (Consistency based Objective Function For alignmEnt Evaluation)

$$\text{COFFEE score} = \frac{\left[\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} W_{i,j} \times \text{SCORE}(A_{i,j})\right]}{\left[\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} W_{i,j} \times \text{LEN}(A_{i,j})\right]}$$

where:

$\text{SCORE}(A_{i,j})$ = number of aligned pairs of residues that are shared between $A_{i,j}$ and the library

- The greater the score, the better the alignment

MUSCLE

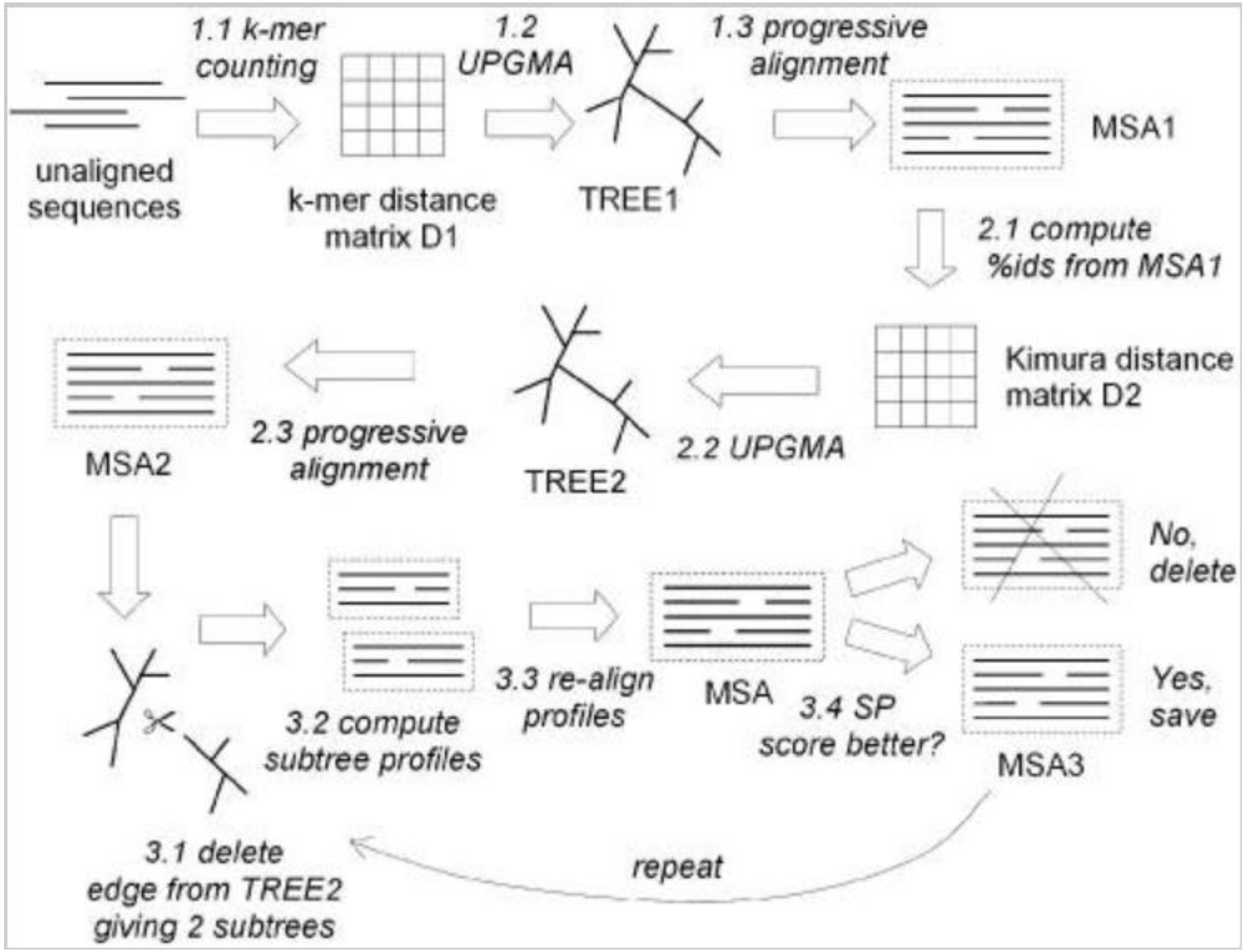
(multiple sequence comparison by log expectation)

- Elements of the Algorithm
  - Distance estimation with kmer counting
  - Progressive alignment with log-expectation Score
  - Refinement using tree-dependent restricted Partitioning

Methods

- Optimizing the sum of pairs score for multiple alignment
- Progressive Method

MUSCLE Algorithm


1.1 k-mer counting
1.2 UPGMA
1.3 progressive alignment
unaligned sequences
k-mer distance matrix D1
TREE1
MSA1
2.1 compute %ids from MSA1
Kimura distance matrix D2
2.2 UPGMA
TREE2
2.3 progressive alignment
MSA2
3.1 delete edge from TREE2 giving 2 subtrees
3.2 compute subtree profiles
3.3 re-align profiles
MSA
3.4 SP score better?
No, delete
Yes, save
MSA3
repeat

Draft Progressive

Kmer distance is computed for each pair of input sequences which gives a distance matrix

- Is a contiguous subsequence of length k

UPGMA is used to cluster the distance matrix to produce a binary tree

Progressive alignment is constructed

- At each leaf a profile is constructed from an input sequence
- At each internal node, a pairwise alignment is contructed of the two child profiles for that internal node

Improved Progressive

*Because error is usually in the kmer distance measure, MUSCLE re-estimates the tree using the Kimura distance

It takes the tree from draft progressive and computes a distance for each pair of input sequences which gives a distance matrix

UPGMA is used to cluster the distance matrix to produce a new binary tree

Progressive alignment is similar to Draft progressive

Refinement

While not converged or user limited

Taking the tree from Improved Progressive, an edge is chosen and is deleted.

- Splits the tree and the profile is recalculated

If the sum of pairs score is improved, it is kept, else it is dropped.

PROs and CONs

- MUSCLE
  - Pros
    - Can handle medium to large alignments with up to 1000 sequences
    - Fast
    - Easy to use
      - Defaults work for most applications
  - Cons
    - Not suitable for sequences with low homology N-terminal and C-terminal extensions
    - Must have a deep understanding to change parameters
- MAFFT
  - Pros
    - Large datasets with up to 30,000 sequences
    - Suitable for sequences with long, low homology N-terminal or C-terminal extensions
    - Suitable for sequences with long internal gaps (use *L-ins-i* algorithm)
  - Cons
    - Complex
      - Need to know parameters to use and when to use them

References

C. NOTREDAME, L. HOLM, and D. G. HIGGINS, “COFFEE : an objective function for multiple sequence alignments,” *Bioinformatics (Oxford, England)*, vol. 14, no. 5, pp. 407–422, 1998, doi: 10.1093/bioinformatics/14.5.407.

F. S.-M. Pais, P. de C. Ruy, G. Oliveira, and R. S. Coimbra, “Assessing the efficiency of multiple sequence alignment programs,” *Algorithms for molecular biology*, vol. 9, no. 1, pp. 4–4, 2014, doi: 10.1186/1748-7188-9-4.

G. E. Crooks, G. Hon, J.-M. Chandonia, and S. E. Brenner, “WebLogo: A Sequence Logo Generator,” *Genome research*, vol. 14, no. 6, pp. 1188–1190, 2004, doi: 10.1101/gr.849004.

K. Katoh, K. Misawa, K. Kuma, and T. Miyata, “MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform,” *Nucleic acids research*, vol. 30, no. 14, pp. 3059–3066, 2002, doi: 10.1093/nar/gkf436.

R. C. Edgar, “MUSCLE: multiple sequence alignment with high accuracy and high throughput,” *Nucleic acids research*, vol. 32, no. 5, pp. 1792–1797, 2004, doi: 10.1093/Nar/Gkh340.

R. J. Edwards, “EdwardsLab,” *Dr Richard Edwards - UPGMA Walkthrough*, 2016. [Online]. Available: http://www.slimsuite.unsw.edu.au/teaching/upgma/. [Accessed: 23-Apr-2021].

“Multiple alignment program for amino acid or nucleotide sequences,” *MAFFT ver.7 - a multiple sequence alignment program.* [Online]. Available: https://mafft.cbrc.jp/alignment/software/algorithms/algorithms.html. [Accessed: 21-Apr-2021].

“Multiple alignments: scoring.” [Online]. Available: http://www.cbs.dtu.dk/dtucourse/cookbooks/gorm/transparencies/mulalign/mulalign4.pdf. [Accessed: 23-Apr-2021].

Kimura, Motoo. "The neutral theory and molecular evolution." *My Thoughts on Biological Evolution*. Springer, Singapore, 2020. 119-138.