



Reflections | Projections



Christine Bakan

Vice President of Software
and Bioinformatics @ Roche
*Delivering Genomic Insights
with Advanced Analytics*

Fri Sept 20 4:00PM SC2405



Alfred Spector

CTO of Two Sigma
*Data Science - Immense Good
Yet Baffling Challenges*

Fri Sept 20 6:00PM SC1404



Donald Kossmann

Director of Microsoft
Research Redmond
*The Global AI
Supercomputer*

Fri Sept 20 5:00PM SC2405

CS 466

Introduction to Bioinformatics

Lecture 8

Mohammed El-Kebir

September 20, 2019



Course Announcements

Instructor:

- Mohammed El-Kebir (melkebir)
- Office hours: Wednesdays, 3:15-4:15pm

TA:

- Ashwin Ramesh (aramesh7)
- Office hours: Fridays, 11:00-11:59am in SC 3405

Homework 2 will be released 9/24 and will be due 10/2, 11:59pm

Outline

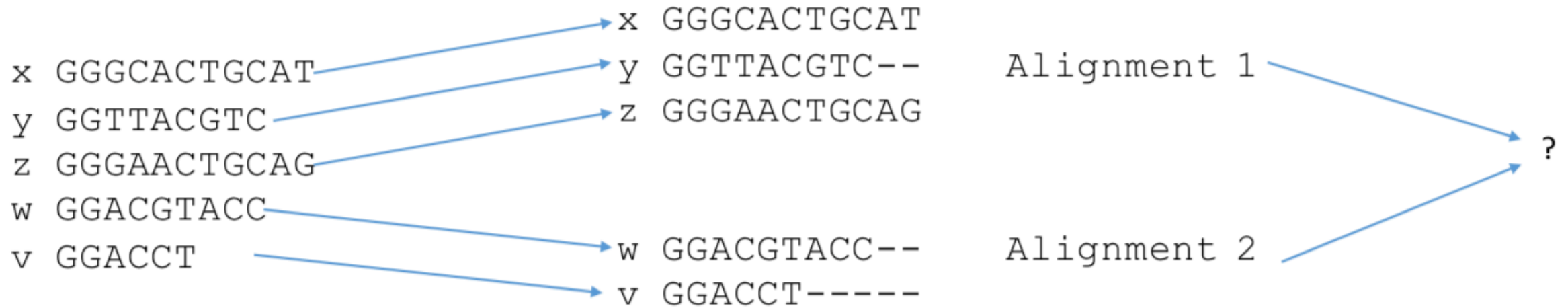
- Progressive alignment
 - Current methods
- Tree and star alignment

Reading:

- Material based on Chapter 14.6 in book “Algorithms on Strings, Trees and Sequences” by Dan Gusfield
- Lecture notes

Heuristic: Iterative/Progressive Alignment

Iteratively add strings (or alignments) to existing alignment(s).



Issues:

1. How to merge alignments?
2. What order to use in merging strings/alignments?

Heuristic Approach: Merge Pairwise Alignments

```
x  GGGCACTGCAT
y  GGTTACGTC--
z  GGGAACTGCAG
```

Alignment 1

```
w  GGACGTACC--
v  GGACCT-----
```

Alignment 2

Question:
Can we align two
alignments?

Need a way to summarize
an alignment and score
merged alignments

Profile Representation of Multiple Alignment

| | | | | | | | | | | | | | | | |
|---|----|---|---|----|---|---|----|----|---|----|----|----|----|---|---|
| | | - | A | G | G | C | T | A | T | C | A | C | C | T | G |
| | T | A | G | - | C | T | A | C | C | A | - | - | - | - | G |
| | C | A | G | - | C | T | A | C | C | A | - | - | - | - | G |
| | C | A | G | - | C | T | A | T | C | A | C | - | G | G | G |
| | C | A | G | - | C | T | A | T | C | G | C | - | G | G | G |
| A | | | 1 | | | | 1 | | | .8 | | | | | |
| C | .6 | | | | 1 | | | .4 | 1 | | .6 | .2 | | | |
| G | | | 1 | .2 | | | | | | .2 | | | .4 | 1 | |
| T | .2 | | | | | 1 | .6 | | | | | | .2 | | |
| - | .2 | | | .8 | | | | | | | .4 | .8 | .4 | | |

A **profile** $P = [p_{i,j}]$ is a $(|\Sigma| + 1) \times l$ matrix, where $p_{i,j}$ is the frequency of i -th letter in j -th position of alignment

Profile Representation of Multiple Alignment

We know how to align sequence against sequence

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | A | G | G | C | T | A | T | C | A | C | C | T | G |
| T | A | G | - | C | T | A | C | C | A | - | - | - | G |
| C | A | G | - | C | T | A | C | C | A | - | - | - | G |
| C | A | G | - | C | T | A | T | C | A | C | - | G | G |
| C | A | G | - | C | T | A | T | C | G | C | - | G | G |

| | | | | | | | | | | | | | |
|---|----|---|----|----|---|---|----|---|----|----|----|----|---|
| A | | 1 | | | | 1 | | | .8 | | | | |
| C | .6 | | | 1 | | | .4 | 1 | | .6 | .2 | | |
| G | | | 1 | .2 | | | | | .2 | | | .4 | 1 |
| T | .2 | | | | 1 | | .6 | | | | | .2 | |
| - | .2 | | .8 | | | | | | | .4 | .8 | .4 | |

Question: Can we align sequence against profile?

Question: Can we align profile against profile?

Aligning String to Profile

A **profile** $P = [p_{i,j}]$ is a $(|\Sigma| + 1) \times n$ matrix, where $p_{i,j}$ is the frequency of i -th letter in j -th position of alignment

Given: Sequences $\mathbf{v} = v_1, \dots, v_m$ and profile P with n columns

- $s[i, j]$ is optimal alignment of v_1, \dots, v_i and first j columns of P
- $\delta(x, y)$ is score for aligning characters x and y
- $\tau(x, j)$ is score for aligning character x and column j of P

Aligning String to Profile

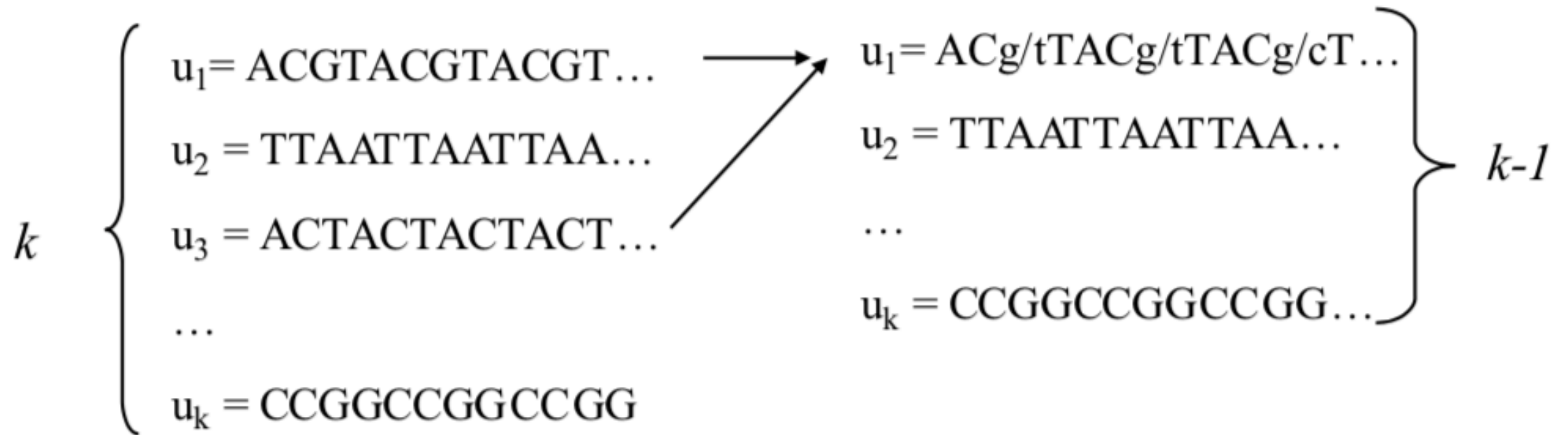
$$\tau(x, j) = \sum_{y \in \Sigma \cup \{-\}} p_{y,j} \cdot \delta(x, y)$$

$$s[i, j] = \max \begin{cases} 0, & \text{if } i = 0 \text{ and } j = 0, \\ s[i-1, j] + \delta(v_i, -), & \text{if } i > 0, \quad \text{Insert space in profile} \\ s[i, j-1] + \tau(-, j), & \text{if } j > 0, \quad \text{Insert space in string} \\ s[i-1, j-1] + \tau(v_i, j), & \text{if } i > 0 \text{ and } j > 0. \end{cases}$$

- $s[i, j]$ is optimal alignment of v_1, \dots, v_i and first j columns of P
- $\delta(x, y)$ is score for aligning characters x and y
- $\tau(x, j)$ is score for aligning character x and column j of P

Progressive Multiple Alignment: Greedy Algorithm

Choose most similar pair among k input strings, combine into a profile. This reduces the original problem to alignment of $k-1$ sequences to a profile. Repeat.



Example

Score of +1 for matches, -1 otherwise.

s2 GTCTGA
s4 GTCAGC (score = 2)

s1 GATTCA--
s4 G-T-CAGC (score = 0)

s1 GAT-TCA
s2 G-TCTGA (score = 1)

s2 G-TCTGA
s3 GATAT-T (score = -1)

s1 GAT-TCA
s3 GATAT-T (score = 1)

s3 GAT-ATT
s4 G-TCAGC (score = -1)

Example

Score of +1 for matches, -1 otherwise.

s2 GTCTGA
s4 GTCAGC (score = 2)

s1 GATTCA--
s4 G-T-CAGC (score = 0)

s1 GAT-TCA
s2 G-TCTGA (score = 1)

s2 G-TCTGA
s3 GATAT-T (score = -1)

s1 GAT-TCA
s3 GATAT-T (score = 1)

s3 GAT-ATT
s4 G-TCAGC (score = -1)

Question: Any theoretical guarantees on optimality?

No guarantees!

Outline

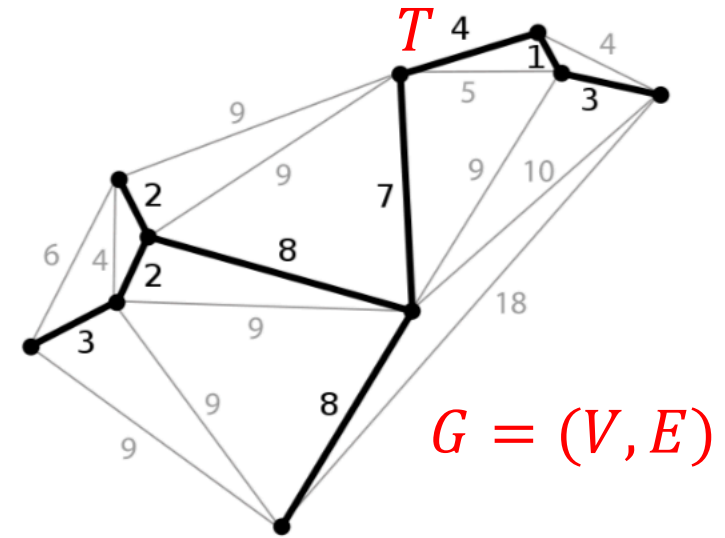
- Progressive alignment
 - Current methods
- Tree and star alignment

Reading:

- Material based on Chapter 14.6 in book “Algorithms on Strings, Trees and Sequences” by Dan Gusfield
- Lecture notes

Progressive Alignment – Feng and Doolittle (1987)

1. Compute pairwise sequence alignments of n sequences
2. Generate complete graph $G = (V, E)$ with edge weights $w : E \rightarrow \mathbb{R}$
3. Compute a (rooted) minimum spanning tree T of G
4. Perform sequence-sequence, sequence-alignment and alignment-alignment alignment to construct MSA according to guide tree T (from most similar to least similar)



Minimum spanning tree is a tree T spanning all vertices of G with minimum total weight

‘Once a gap, always a gap’

Progressive Alignment – ClustalW (1994)

- Widely used alignment method by Thompson, Higgins and Gibson (1994)
- W stands for weighted:
 - Input sequences are weighted to compensate for biased representation
 - Different substitution matrices depending on expected similarity in guide tree (BLOSUM80 for closely related sequences, and BLOSUM50 for distant sequences)
 - Position-specific gap-open and gap-extend penalties depending on context (hydrophobic vs. hydrophilic)

Three steps:

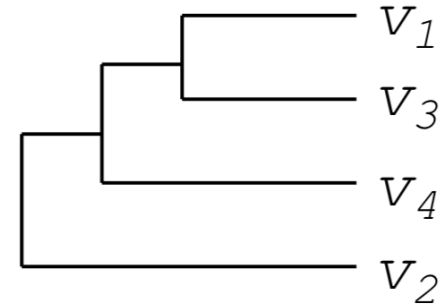
1. Construct pairwise alignments
2. Build guide tree T using neighbor joining*
3. Progressive profile alignment guided by T

ClustalW – Step 2: Guide Tree

Create Guide Tree using the similarity matrix

(“cluster” distances. Details to come...)

| | V_1 | V_2 | V_3 | V_4 |
|-------|-------|-------|-------|-------|
| V_1 | – | | | |
| V_2 | .17 | – | | |
| V_3 | .87 | .28 | – | |
| V_4 | .59 | .33 | .62 | – |



ClustalW uses the neighbor-joining method

Guide tree roughly reflects evolutionary relationships


Calculate:

$$\begin{aligned} V_{1,3} &= \text{alignment}(v_1, v_3) \\ V_{1,3,4} &= \text{alignment}((V_{1,3}), v_4) \\ V_{1,2,3,4} &= \text{alignment}((V_{1,3,4}), v_2) \end{aligned}$$

ClustalW – Step 3: Progressive Alignment

- Start by aligning the two most similar sequences
- Following the guide tree, add in the next sequences, aligning to the existing alignment
- Insert gaps as necessary

```
FOS_RAT      PEEMSVTS-LDLTGGLPEATTPESEEAF TLPLLNDPEPK-PSLEPVKNI SNMELKAEPFD
FOS_MOUSE    PEEMSVAS-LDLTGGLPEASTPESEEAF TLPLLNDPEPK-PSLEPVKSI SNVELKAEPFD
FOS_CHICK     SEELAAATALDLG-----APSPAAAEAFALPLMTEAPPAVPPKEPSG--SGLELKAEPFD
FOSB_MOUSE    PGPGPLAEVRDLPG-----STSAKEDGFGWLLPPPPPPP-----LPPFQ
FOSB_HUMAN    PGPGPLAEVRDLPG-----SAPAKEDGFSWLLPPPPPPP-----LPPFQ
.             . : ** . :... * : . * * . * ** :
```



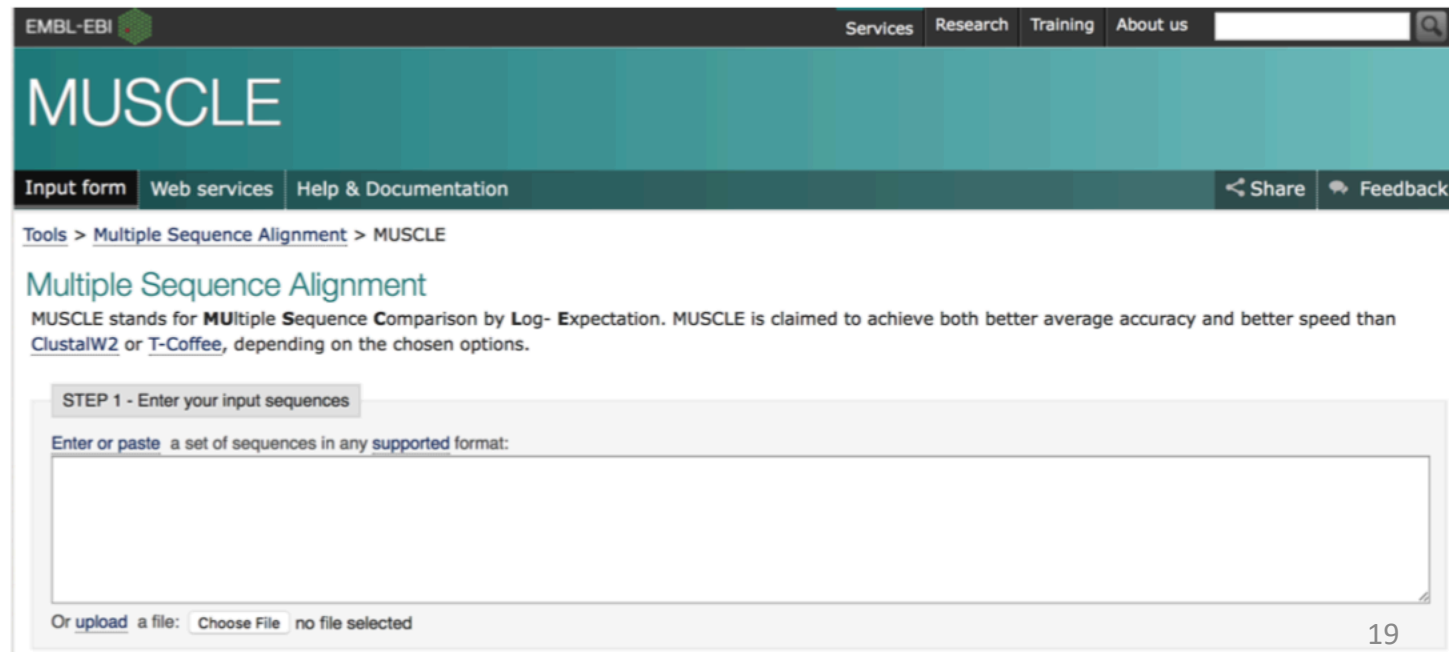
Dots and stars show how well-conserved a column is.

MUSCLE (Edgar, 2004)

Multiple Sequence Comparison by Log-Expectation

Three phases:

1. Draft progressive alignment: fast heuristic
2. Improved progressive: use tree derived in phase 1
3. Refinement of MSA
 - Remove sequence from MSA and realign to profile of remaining sequences
 - Repeat until convergence



The screenshot shows the EMBL-EBI MUSCLE web interface. At the top, there is a navigation bar with links for Services, Research, Training, and About us. Below this is a teal header with the word 'MUSCLE' in white. A secondary navigation bar contains links for Input form, Web services, and Help & Documentation, along with Share and Feedback buttons. The main content area is titled 'Multiple Sequence Alignment' and includes a brief description of the tool. A 'STEP 1 - Enter your input sequences' section features a large text input field and a file upload option.

EMBL-EBI

Services Research Training About us

MUSCLE

Input form Web services Help & Documentation

Share Feedback

Tools > Multiple Sequence Alignment > MUSCLE

Multiple Sequence Alignment

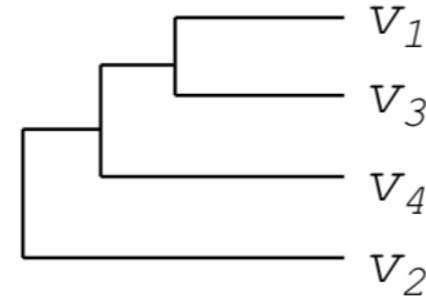
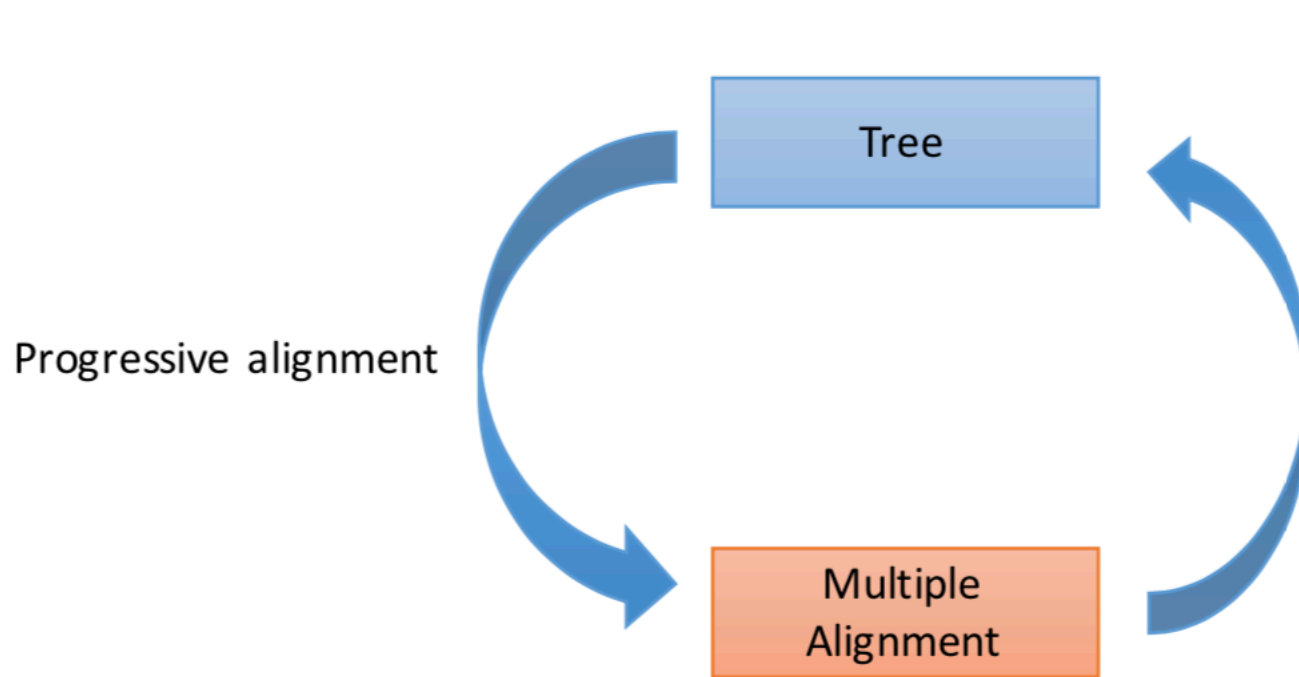
MUSCLE stands for **M**ultiple **S**equence **C**omparison by **L**og- **E**xpectation. MUSCLE is claimed to achieve both better average accuracy and better speed than [ClustalW2](#) or [T-Coffee](#), depending on the chosen options.

STEP 1 - Enter your input sequences

[Enter or paste](#) a set of sequences in any [supported](#) format:

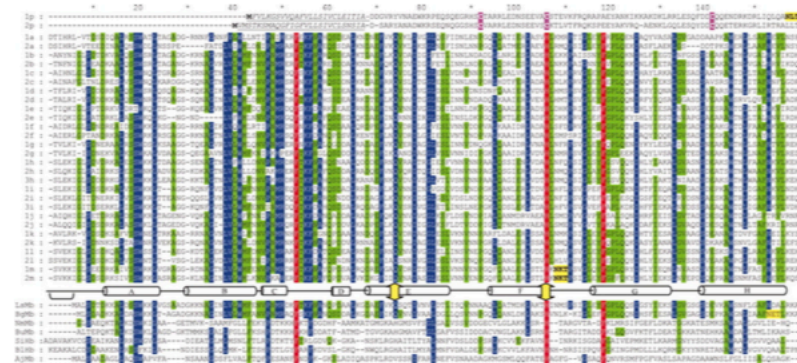
Or [upload](#) a file: [Choose File](#) no file selected

Progressive MSA



Circularity!

Ideally, want to derive alignment and tree simultaneously → Hard



Outline

- Progressive alignment
 - Current methods
- Tree and star alignment

Reading:

- Material based on Chapter 14.6 in book “Algorithms on Strings, Trees and Sequences” by Dan Gusfield
- Lecture notes

Tree Alignment

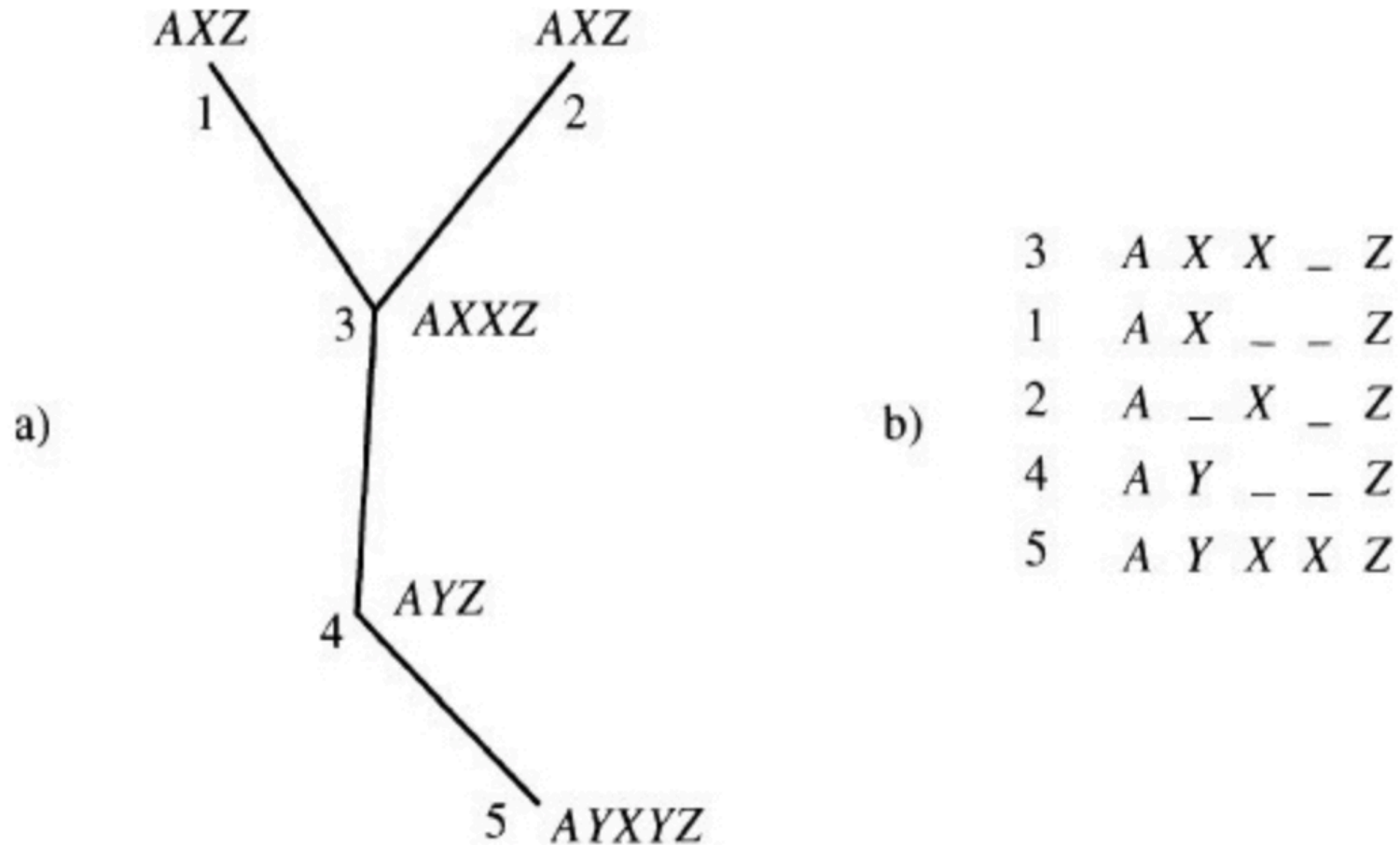


Figure 14.6: a. A tree with its nodes labeled by a (multi)set of strings, b. A multiple alignment of those strings that is consistent with the tree. The pairwise scoring scheme scores a zero for each match and a one for each mismatch or space opposite a character. The reader can verify that each of the four induced alignments specified by an edge of the tree has a score equal to its respective optimal distance. However, the induced alignment of two strings which do not label adjacent nodes may have a score greater than their optimal pairwise distance.

Summary

1. Optimal pairwise alignment by dynamic programming in $O(n^2)$ time
2. Optimal multiple alignment with SP-score by dynamic programming in $O(k^2 2^k n^k)$ time
3. Multiple alignment with SP-score is NP-hard (Jiang and Wang, 1994)
4. Carrillo-Lipman enables us to decide whether alignment passes through a vertex (i_1, i_2, i_3) for $k = 3$ sequences (generalizes to $k > 3$)
5. Progressive alignment methods are widely used, but come with no theoretical bounds on alignment quality
6. Star alignment gives 2-approximation algorithm

History

- 1975 Sankoff
Formulated MSA problem and gave dynamic programming solution
- 1988 Carrillo-Lipman
Branch and Bound approach for MSA
- 1990 Feng-Doolittle
Progressive alignment
- 1993 Gusfield
Star alignment: 2-approximation algorithm
- 1994 Jiang and Wang
MSA with SP-score is NP-hard
- 1994 Thompson-Higgins-Gibson: ClustalW
Most popular multiple alignment program
- 2000 Notredam-Higgins-Heringa: T-coffee
Use library of pairwise alignments
- 2004 Edgar: MUSCLE
Refinement