Reflections|Projections 2020

Largest student run tech conference in the midwest, taking place September 21 through September 26
Virtual for the first time in 2020; featuring 20+ speakers and 5+ companies

Opportunity to hear about the bleeding edge on topics including

- Computer science for clinical methods (Tuesday @ 5)
- Ethical artificial intelligence (Wednesday @ 4, Friday @ 7)
- How to be successful in a startup (Monday @ 7, Wednesday @ 5)
- The essentials of product management (Thursday @ 5)
- Protecting privacy at a global scale (Friday @ 4)
- How to build accessible applications (Friday @ 6)
- And more! Check out our social media or website for more details

Network with top companies like Goldman Sachs, Grainger, IMC, Aechelon
Participate in a beginner-friendly, team-based AI competition with prizes
Swag pickup for attendees on campus
Opportunity to win free food!

Why wait? Register today at **acmrp.org**

Reflections | Projections

Mark Ciaccio

Senior Data Scientist @ AbbVie

Tuesday, September 22 from 6-7PM

reflectionsprojections.org

CS 466
Introduction to Bioinformatics
Lecture 8

Mohammed El-Kebir

September 18, 2020

Course Announcements

Instructor:

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

TAs:

- Sarah Christensen (sac2) – Mondays, 3-4pm
- Wesley Wei Qian (weiqian3) – Fridays, 9-10am

Homework 2 will be released 9/29 and due 10/1 by 11:59pm (CT)

Midterm will be 10/7, 7-10pm (CT)

Outline

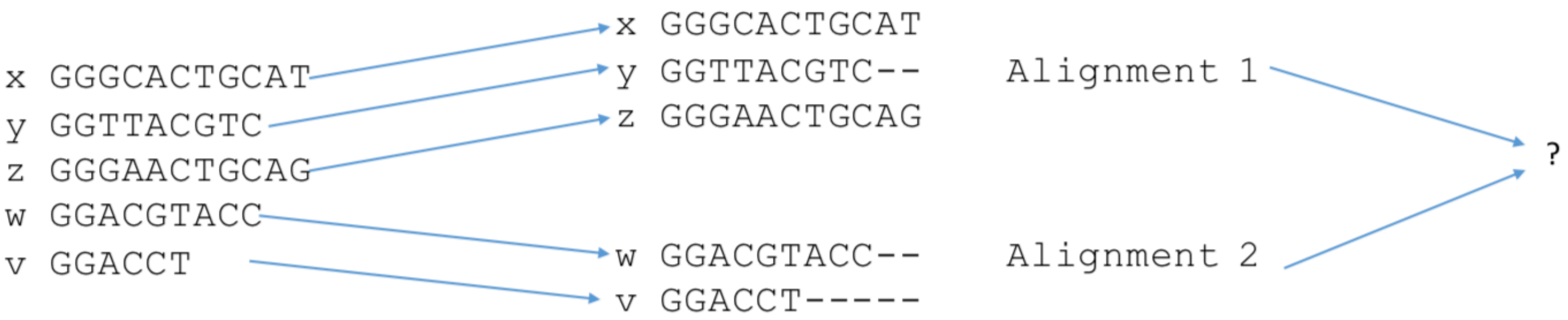
- Progressive alignment
  - Current methods
- Tree and star alignment

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?

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
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	

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

Aligning String to Profile

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

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

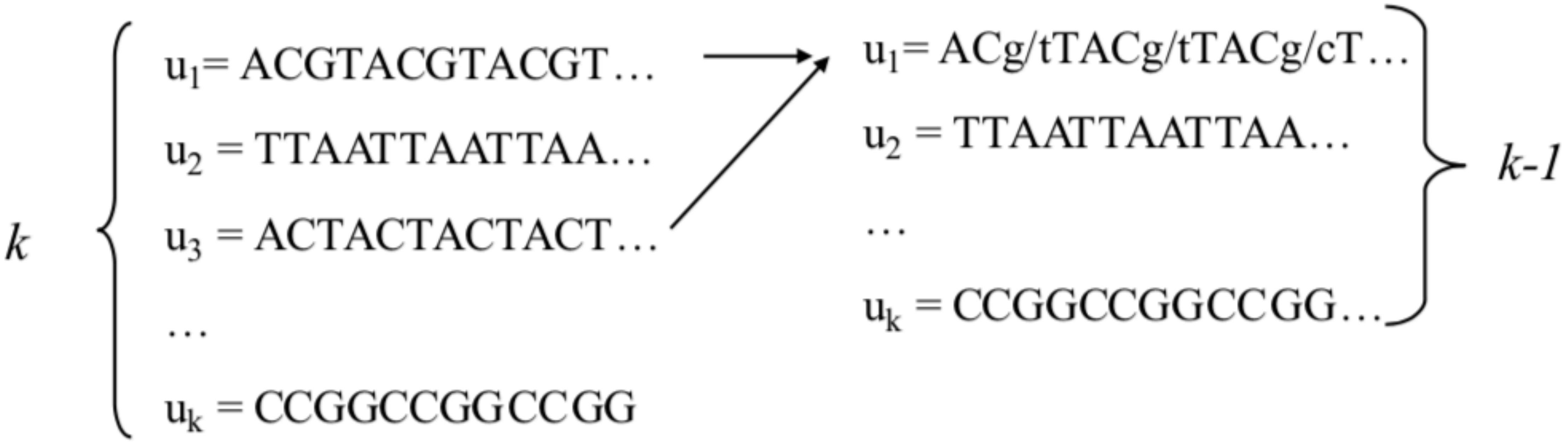
Insert space in profile (case $i > 0$)

Insert space in string (case $j > 0$)

- $s[i, j]$ is optimal alignment of $v_1, \ldots, v_i$ 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.



Outline

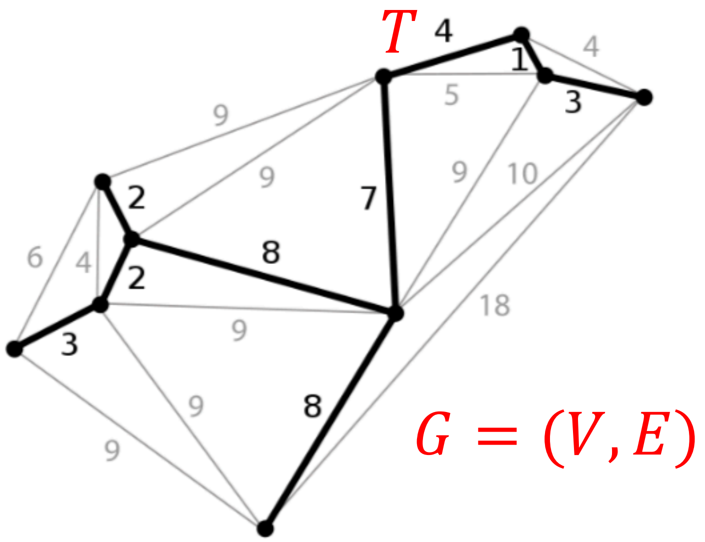
- Progressive alignment
  - Current methods
- Tree and star alignment

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 $w : E \to \mathbb{R}$
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)

<u>**Three steps:**</u>

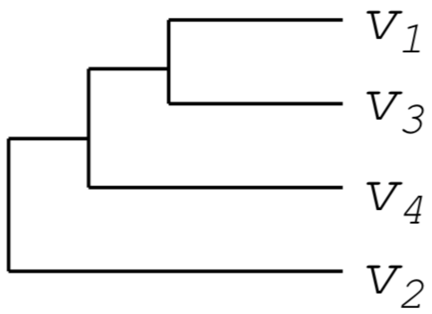
1. Construct pairwise alignments
2. Build guide tree T using neighbor joining*
3. Progressive profile alignment guided by T

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:

$v_{1,3}$ = alignment (v_1, v_3)

$v_{1,3,4}$ = alignment$((v_{1,3}), v_4)$

$v_{1,2,3,4}$ = alignment$((v_{1,3,4}), v_2)$

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-LDLTGGLPEATTPESEEAFTLPLLNDPEPK-PSLEPVKNISNMELKAEPFD
FOS_MOUSE       PEEMSVAS-LDLTGGLPEASTPESEEAFTLPLLNDPEPK-PSLEPVKSISNVELKAEPFD
FOS_CHICK       SEELAAATALDLG----APSPAAAEEAFALPLMTEAPPAVPPKEPSG--SGLELKAEPFD
FOSB_MOUSE      PGPGPLAEVRDLPG-----STSAKEDGFGWLLPPPPPPP-----------------LPFQ
FOSB_HUMAN      PGPGPLAEVRDLPG-----SAPAKEDGFSWLLPPPPPPP-----------------LPFQ
                .   . :   ** .     :..  *:.*   *   . *                   **:
```

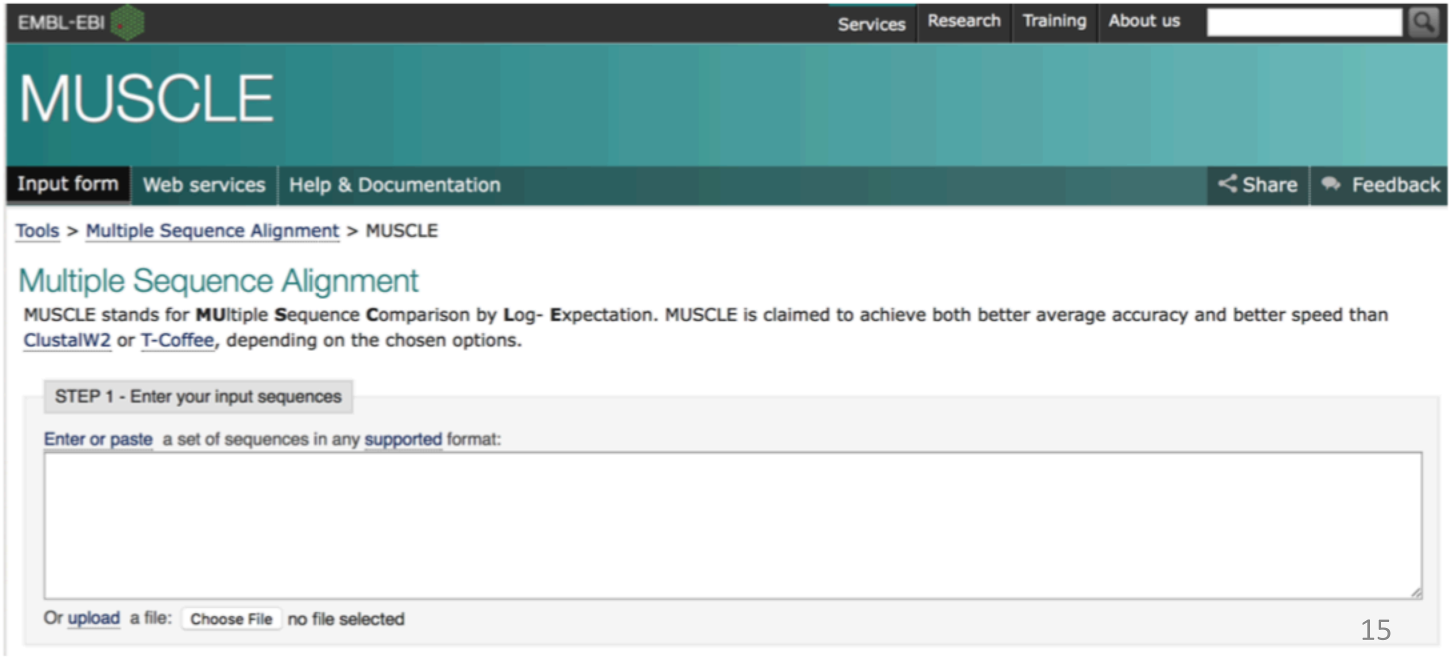
Dots and stars show how well-conserved a column is.

MUSCLE (Edgar, 2004)

<u>Mu</u>ltiple <u>S</u>equence <u>C</u>omparison by <u>L</u>og-<u>E</u>xpectation

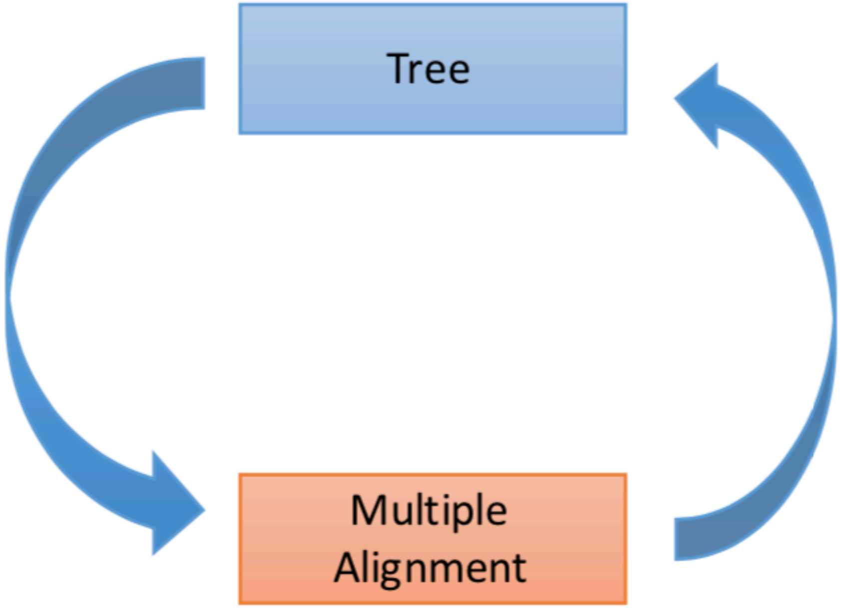
<u>Three phases:</u>

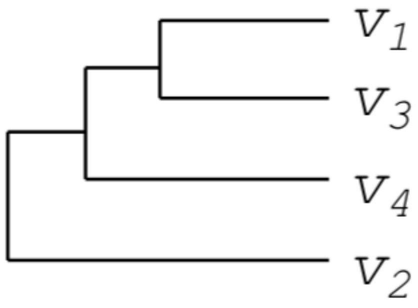
1. Draft progressive alignment: fast heuristic
2. Improved progressive: use tree derived in phase 1
3. Refinement of MSA
   - Remove sequence from MSA and realign to profile of remaining sequences
   - Repeat until convergence

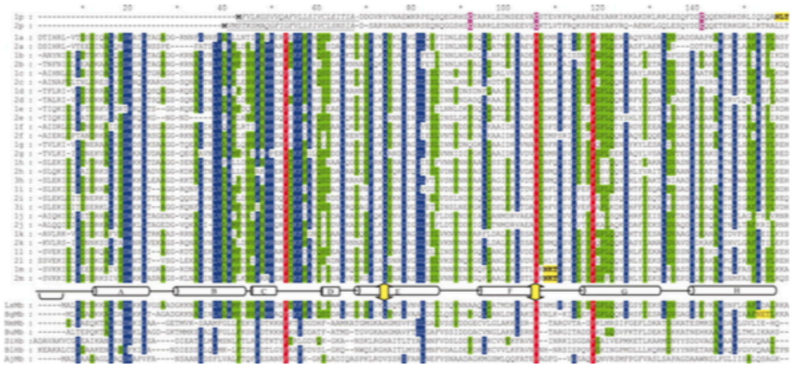


Progressive MSA

Progressive alignment







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

Multiple Sequence Alignment Problem w/ SP-Score

MSA-SP problem: Given strings $\mathbf{v}_1, \ldots, \mathbf{v}_k$ and scoring function $\delta : (\Sigma \cup \{-\}) \times (\Sigma \cup \{-\}) \rightarrow \mathbb{R}$, find multiple sequence alignment $\mathcal{M}^*$ with **maximum** value of SP-score$(\mathcal{M}^*) = \sum_{i=1}^{k} \sum_{j=i+1}^{k} S(\mathbf{v}_i, \mathbf{v}_j)$ where $S(\mathbf{v}_i, \mathbf{v}_j)$ is the score of the induced pairwise alignment of $(\mathbf{v}_i, \mathbf{v}_j)$ in $\mathcal{M}^*$ using δ

Weighted SP-Edit Distance problem: Given strings $\mathbf{v}_1, \ldots, \mathbf{v}_k$ and cost function $\delta : (\Sigma \cup \{-\}) \times (\Sigma \cup \{-\}) \rightarrow \mathbb{R}$, find multiple sequence alignment $\mathcal{M}^*$ with **minimum** value of SP-score$(\mathcal{M}^*) = \sum_{i=1}^{k} \sum_{j=i+1}^{k} S(\mathbf{v}_i, \mathbf{v}_j)$ where $S(\mathbf{v}_i, \mathbf{v}_j)$ is the cost of the induced pairwise alignment of $(\mathbf{v}_i, \mathbf{v}_j)$ in $\mathcal{M}^*$ using δ

Tree Alignment

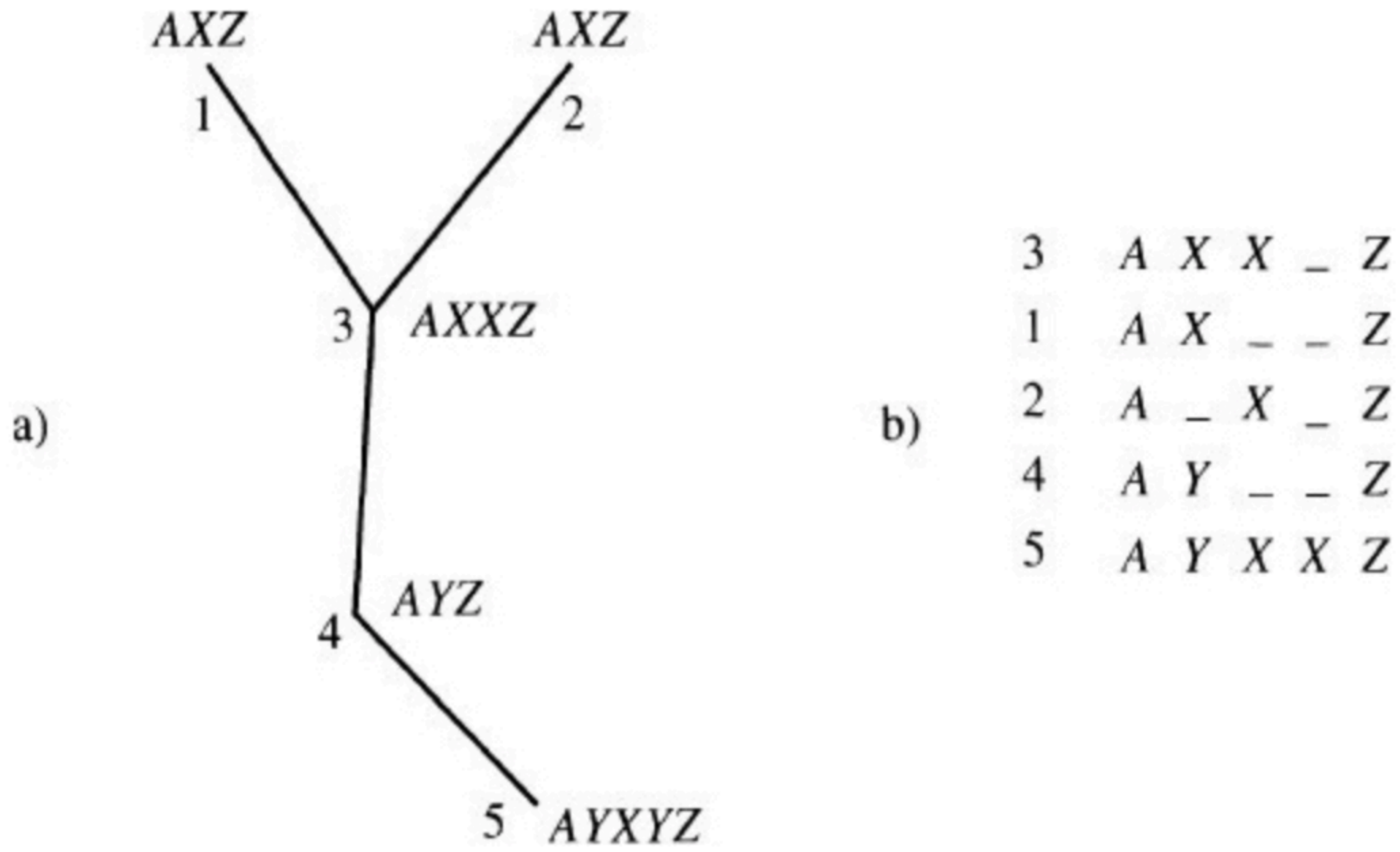


Figure 14.6: a. A tree with its nodes labeled by a (multi)set of strings, b. A multiple alignment of those strings that is consistent with the tree. The pairwise scoring scheme scores a zero for each match and a one for each mismatch or space opposite a character. The reader can verify that each of the 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