CS 466
Introduction to Bioinformatics
Lecture 11

Mohammed El-Kebir

October 15, 2018

Course Announcements

Instructor:

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

TA:

- Anusri Pampari (pampari2)
- Office hours: Thursdays, 11:00-11:59am in SC 4105

Grades of midterm exam will be released on Wednesday, Oct. 17

Outline

- Multiple sequence alignment
- Scoring matrices
- Tree/star alignment
- Progressive alignment methods

Reading:

- Material based on Chapter 14.6 in book “Algorithms on Strings, Trees and Sequences” by Dan Gusfield
- Chapter 6.7 in Jones and Pevzner
- Lecture notes

Multiple Sequence Alignment (MSA)

A **multiple sequence alignment** $\mathcal{M}$ between k strings $\mathbf{v}_1, \ldots, \mathbf{v}_k$ is a $k \times q$ matrix, where $q = \{\max\{\, |\mathbf{v}_i| : i \in [k] \,\}, \ldots, \sum_{i=1}^{k} |\mathbf{v}_i|\}$ such that the i-th row contains the characters of $\mathbf{v}_i$ in order with spaces '-' interspersed and no column contains k spaces

$\mathbf{v}_1$	A	T	–	G	C	G	–
$\mathbf{v}_2$	A	–	C	G	T	–	C
$\mathbf{v}_3$	A	T	C	A	C	–	A

Multiple Alignment Induces Pairwise Alignments

v_1	A	T	-	G	C	G	-
v_2	A	-	C	G	T	-	C
v_3	A	T	C	A	C	-	A

Multiple sequence alignment $\mathcal{M}$

v_1	A	T	-	G	C	G	-
v_2	A	-	C	G	T	-	C

v_1	A	T	-	G	C	G	-
v_3	A	T	C	A	C	-	A

v_2	A	-	C	G	T	C
v_3	A	T	C	A	C	A

Resulting columns with -/- are removed

Sum-of-Pairs (SP) Score

$\mathbf{v}_1$	A	T	–	G	C	G	–
$\mathbf{v}_2$	A	–	C	G	T	–	C
$\mathbf{v}_3$	A	T	C	A	C	–	A

Multiple sequence alignment $\mathcal{M}$

$S(\mathbf{v}_i, \mathbf{v}_j)$ is score of induced pairwise alignment of sequences $(\mathbf{v}_i, \mathbf{v}_j)$

$\mathbf{v}_1$	A	T	–	G	C	G	–
$\mathbf{v}_2$	A	–	C	G	T	–	C

$\mathbf{v}_1$	A	T	–	G	C	G	–
$\mathbf{v}_3$	A	T	C	A	C	–	A

$\mathbf{v}_2$	A	–	C	G	T	–	C
$\mathbf{v}_3$	A	T	C	A	C	–	A

$$\text{SP-score}(\mathcal{M}) = \sum_{i=1}^{k} \sum_{j=i+1}^{k} S(\mathbf{v}_i, \mathbf{v}_j)$$

Sum-of-Pairs (SP) Score

$\mathbf{v}_1$	A	T	–	G	C	G	–
$\mathbf{v}_2$	A	–	C	G	T	–	C
$\mathbf{v}_3$	A	T	C	A	C	–	A

Multiple sequence alignment $\mathcal{M}$

Question: What is a lower bound on SP-score($\mathcal{M}$)?

$\mathbf{v}_1$	A	T	–	G	C	G	–
$\mathbf{v}_2$	A	–	C	G	T	–	C

$\mathbf{v}_1$	A	T	–	G	C	G	–
$\mathbf{v}_3$	A	T	C	A	C	–	A

$\mathbf{v}_2$	A	–	C	G	T	C	
$\mathbf{v}_3$	A	T	C	A	C	A	

$$\text{SP-score}(\mathcal{M}) = \sum_{i=1}^{k} \sum_{j=i+1}^{k} S(\mathbf{v}_i, \mathbf{v}_j)$$

Multiple Sequence Alignment Problem w/ SP-Score

A **multiple sequence alignment** $\mathcal{M}$ between k strings $\mathbf{v}_1, \ldots, \mathbf{v}_k$ is a $k \times q$ matrix, where $q = \{\max\{ |\mathbf{v}_i| : i \in [k]\}, \ldots, \sum_{i=1}^{k} |\mathbf{v}_i|\}$ such that the i-th row contains the characters of $\mathbf{v}_i$ in order with spaces '-' interspersed and no column contains k spaces

MSA-SP problem: Given strings strings $\mathbf{v}_1, \ldots, \mathbf{v}_k$ 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 score of the induced pairwise alignment of $(\mathbf{v}_i, \mathbf{v}_j)$ in $\mathcal{M}^*$

Multiple Sequence Alignment Problem w/ SP-Score

MSA-SP problem: Given strings strings $\mathbf{v}_1, \ldots, \mathbf{v}_k$ 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 score of the induced pairwise alignment of $(\mathbf{v}_i, \mathbf{v}_j)$ in $\mathcal{M}^*$

Question: Can we align k sequences each of length n in time $O(\text{poly}(n))$?

No, MSA-SP is NP-hard.

[WANG, L., & JIANG, T. (1994). On the Complexity of Multiple Sequence Alignment. *Journal of Computational Biology*, *1*(4), 337–348. http://doi.org/10.1089/cmb.1994.1.337]

Outline

- Multiple sequence alignment
- Scoring matrices
- Tree/star alignment
- Progressive alignment methods

Reading:

- Material based on Chapter 14.6 in book “Algorithms on Strings, Trees and Sequences” by Dan Gusfield
- Chapter 6.7 in Jones and Pevzner
- Lecture notes

Substitution Matrices

- Given a pair $(\mathbf{v}, \mathbf{w})$ of aligned sequences, we want to assign a score that measure the **relative likelihood** that the sequences are **related** as opposed to being **unrelated**
- We need two models:
  - Random model R: each letter $a \in \Sigma$ occurs independently with probability q_a
  - Match model M: aligned pair $(a, b) \in \Sigma \times \Sigma$ occur with joint probability $p_{a,b}$

$$\Pr(\mathbf{v}, \mathbf{w}|\, R) = \prod_i q_{v_i} \cdot \prod_i q_{w_i}$$

$$\Pr(\mathbf{v}, \mathbf{w}|\, M) = \prod_i p_{v_i, w_i}$$

$$\log \frac{\Pr(\mathbf{v}, \mathbf{w}|\, M)}{\Pr(\mathbf{v}, \mathbf{w}|\, R)} = \sum_i s(v_i, w_i) \text{ where } s(a, b) = \log \frac{p_{a,b}}{q_a q_b}$$

BLOSUM (Blocks Substitution Matrices)

- Henikoff and Henikoff, 1992
- Computed using ungapped alignments of protein segments from BLOCKS database
- Two sequences are put in same cluster if % identical residues exceeds L
  - BLOSUM40: derived from alignments that are at least $L = 40\%$ identical
  - BLOSUM62: derived from alignments that are at least $L = 62\%$ identical [most widely used]
  - BLOSUM80: derived from alignments that are at least $L = 80\%$ identical
- From clustering matrix $E = [e_{a,b}]$ was obtained, where $e_{a,b}$ is the number of times a and b were present in all pairs of distinct clusters C and C' in the same column multiplied by $1/(|C| \cdot |C'|)$.
- $q_a = \sum_b e_{a,b} / \sum_{c,d} E_{c,d}$ and $q_{a,b} = e_{a,b} / \sum_{c,d} E_{c,d}$

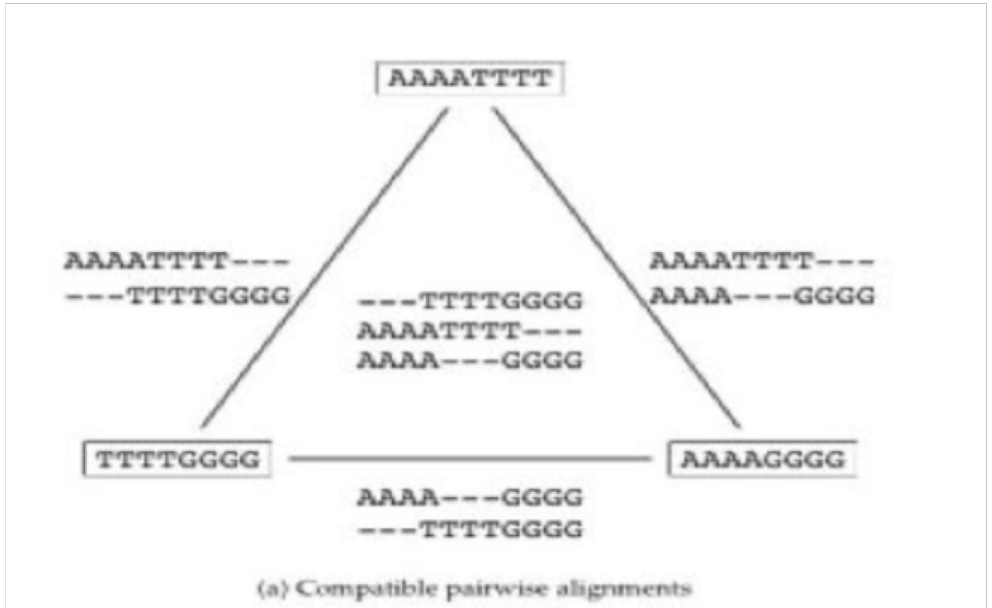
BLOSUM62

	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	4																			
Arg	−1	5																		
Asn	−2	0	6																	
Asp	−2	−2	1	6																
Cys	0	−3	−3	−3	9															
Gln	−1	1	0	0	−3	5														
Glu	−1	0	0	2	−4	2	5													
Gly	0	−2	0	−1	−3	−2	−2	6												
His	−2	0	1	−1	−3	0	0	−2	8											
Ile	−1	−3	−3	−3	−1	−3	−3	−4	−3	4										
Leu	−1	−2	−3	−4	−1	−2	−3	−4	−3	2	4									
Lys	−1	2	0	−1	−3	1	1	−2	−1	−3	−2	5								
Met	−1	−1	−2	−3	−1	0	−2	−3	−2	1	2	−1	5							
Phe	−2	−3	−3	−3	−2	−3	−3	−3	−1	0	0	−3	0	6						
Pro	−1	−2	−2	−1	−3	−1	−1	−2	−2	−3	−3	−1	−2	−4	7					
Ser	1	−1	1	0	−1	0	0	0	−1	−2	−2	0	−1	−2	−1	4				
Thr	0	−1	0	−1	−1	−1	−1	−2	−2	−1	−1	−1	−1	−2	−1	1	5			
Trp	−3	−3	−4	−4	−2	−2	−3	−2	−2	−3	−2	−3	−1	1	−4	−3	−2	11		
Tyr	−2	−2	−2	−3	−2	−1	−2	−3	2	−1	−1	−2	−1	3	−3	−2	−2	2	7	
Val	0	−3	−3	−3	−1	−2	−2	−3	−3	3	1	−2	1	−1	−2	−2	0	−3	−1	4

Compatibility

Compatible: Pairwise alignments can be combined into multiple alignment

Incompatible: Pairwise alignments *cannot* be combined into multiple alignment

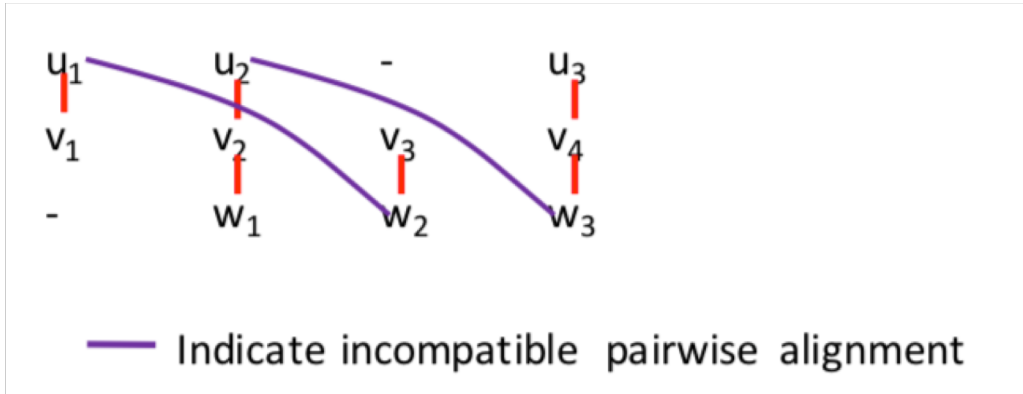


(a) Compatible pairwise alignments

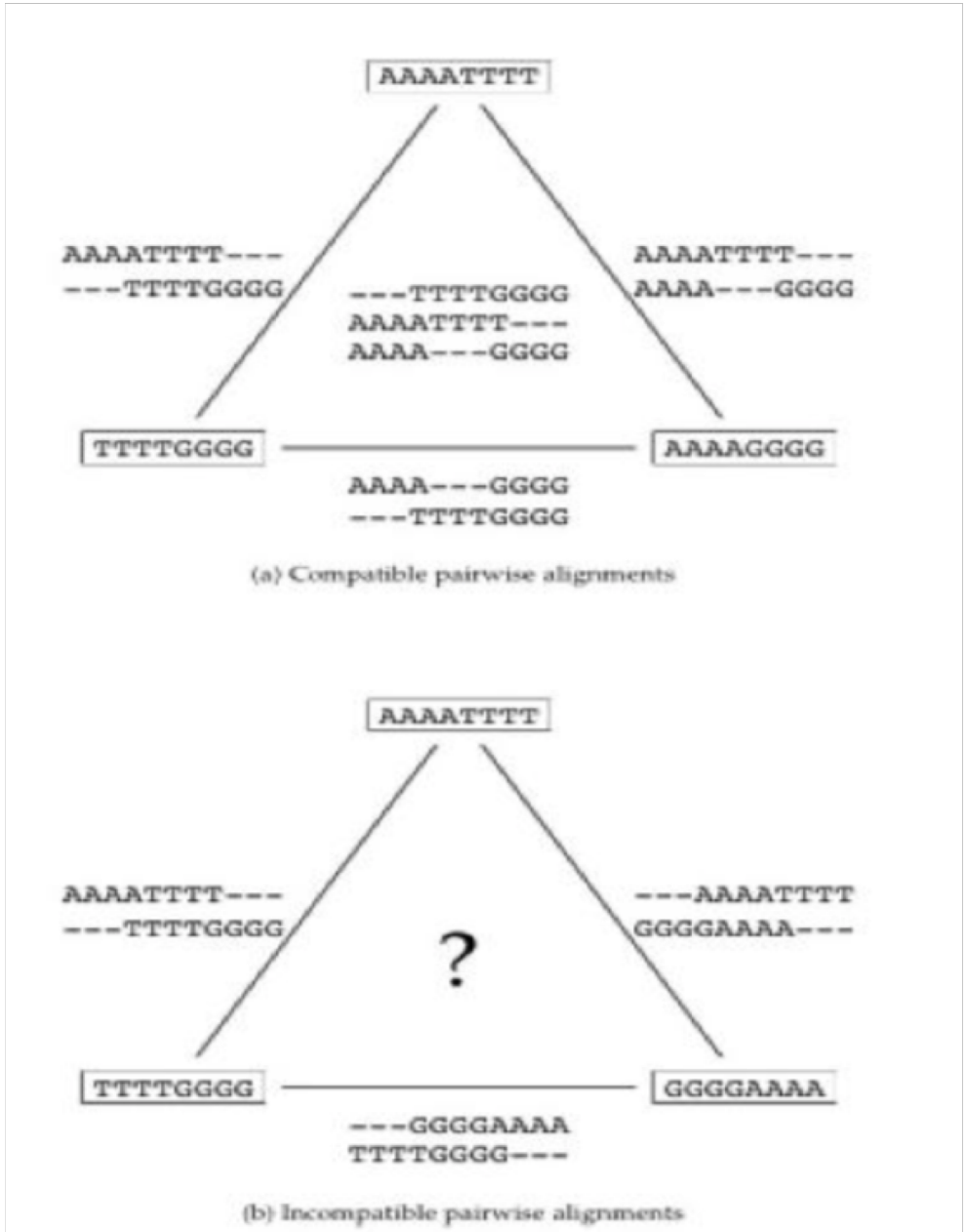
(b) Incompatible pairwise alignments

Compatibility

Compatible: Pairwise alignments can be combined into multiple alignment



Incompatible: Pairwise alignments *cannot* be combined into multiple alignment



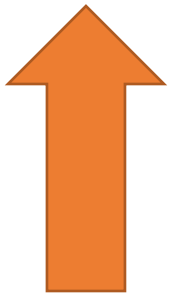
From Compatible Pairwise to Multiple Alignment

Optimal multiple alignment

Easy

Pairwise alignments between *all* pairs of sequences, but they are *not* necessarily optimal

Optimal multiple alignment

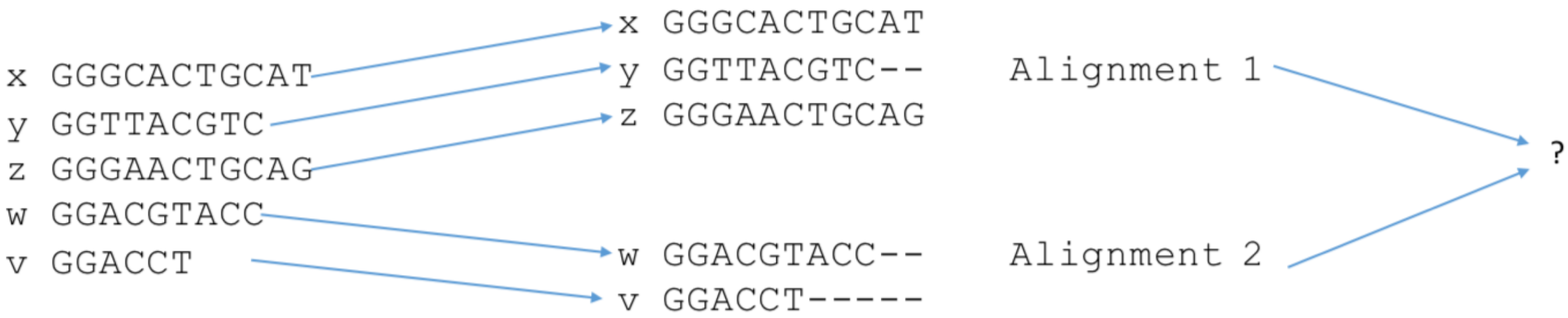


Challenging

Good (not necessary optimal) *compatible* pairwise alignments between all sequences

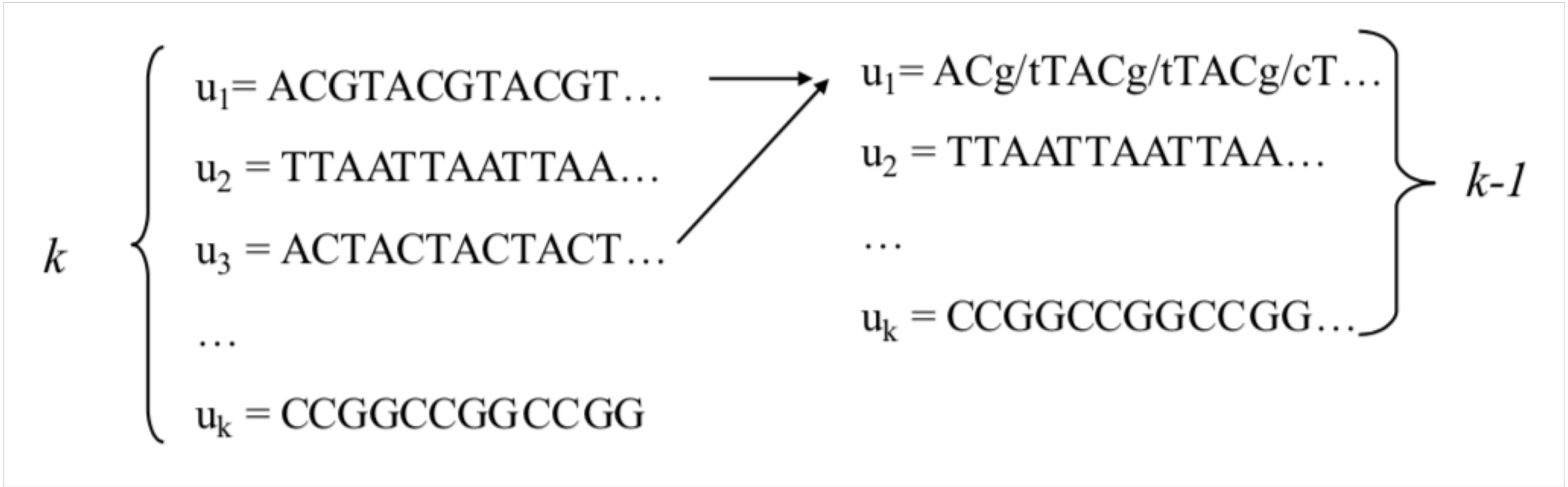
Heuristic: Iterative/Progressive Alignment

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



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

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

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

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

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

Question: Any theoretical guarantees on optimality?

No guarantees!