Multiple Sequence Alignment

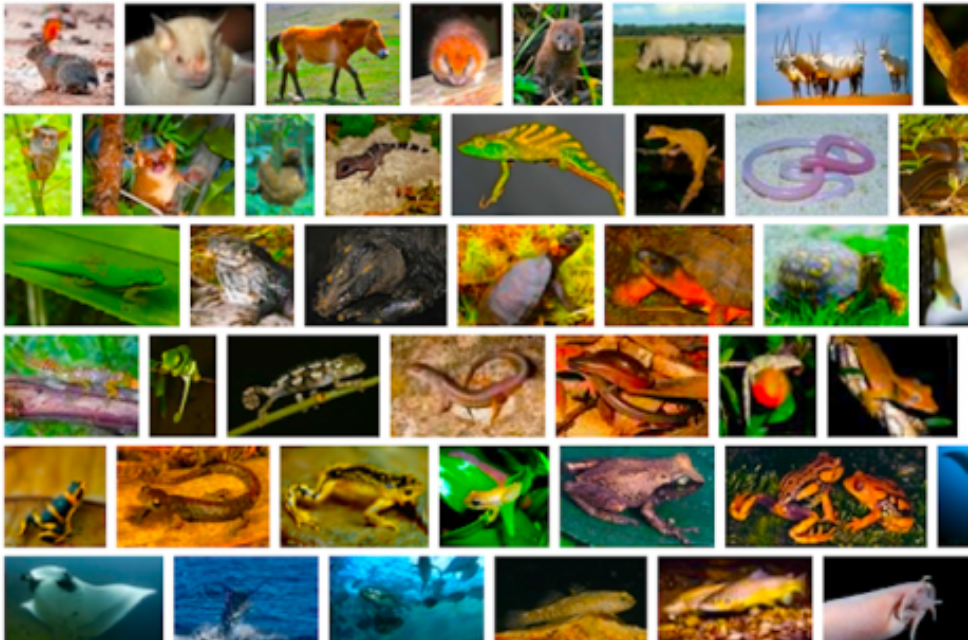
Irene Ong
BMI/CS 576
www.biostat.wisc.edu/bmi576/
Irene.ong@wisc.edu
Fall 2017

Key concepts

- The Multiple Sequence Alignment Problem
- Scoring Multiple Sequence Alignments
  - Scoring an alignment of a "profile" and a sequence
- Heuristic Algorithms for Multiple Sequence Alignment
  - General strategies
    - Progressive alignment
      - Star alignment
      - Tree-based alignment
    - Iterative alignment

From Pairwise to Multiple Alignment

- Up until now we have mainly aligned two sequences.
- A faint (and statistically insignificant) similarity between two sequences becomes significant if it is present in many other sequences.
- Multiple alignments can reveal subtle similarities that pairwise alignments do not reveal.



Alignment of Three A-domains

```
YAFDLGYTCMFPVLLGGGELHIVQKETYTAPDEIAHYIKEHGITYIKLTPSLFHTIVNTASFAFDANFESLRLIVLGGEKIIPIDVIAFRKMYGHTE-FINHYGPTEATIGA
-AFDVSAGDFARALLTGGQLIVCPNEVKMDPASLYAIIKKYDITIFEATPALVIPLMEYI-YEQKLDISQLQILIVGSDSCSMEDFKTLVSRFGSTIRIVNSYGVTEACIDS
IAFDASSWEIYAPLLNGGTVVCIDYYTTIDIKALEAVFKQHHIRGAMLPPALLKQCLVSA----PTMISSLEILFAAGDRLSSQDAILARRAVGSGV-Y-NAYGPTENTVLS
```

Alignment of Three A-domains

YAFDLGYTCMFPVLLGGGELHIVQKETYTAPDEIAHYIKEHGITYIKLTPSLFHTIVNTASFAFDANFESLRLIVLGGEKIIPIDVIAFRKMYGHTE-FINHYGPTEATIGA

-AFDVSAGDFARALLTGGQLIVCPNEVKMDPASLYAIIKKYDITIFEATPALVIPLMEYI-YEQKLDISQLQILIVGSDSCSMEDFKTLVSRFGSTIRIVNSYGVTEACIDS

IAFDASSWEIYAPLLNGGTVVCIDYYTTIDIKALEAVFKQHHIRGAMLPPALLKQCLVSA----PTMISSLEILFAAGDRLSSQDAILARRAVGSGV-Y-NAYGPTENTVLS

Alignment of Three A-domains

```
YAFDLGYTCMFPVLLGGGELHIVQKETYTAPDEIAHYIKEHGITYIKLTPSLFHTIVNTASFAFDANFESLRLIVLGGEKIIPIDVIAFRKMYGHTE-FINHYGPTEATIGA

-AFDVSAGDFARALLTGGQLIVCPNEVKMDPASLYAIIKKYDITIFEATPALVIPLMEYI-YEQKLDISQLQILIVGSDSCSMEDFKTLVSRFGSTIRIVNSYGVTEACIDS

IAFDASSWEIYAPLLNGGTVVCIDYYTTIDIKALEAVFKQHHIRGAMLPPALLKQCLVSA----PTMISSLEILFAAGDRLSSQDAILARRAVGSGV-Y-NAYGPTENTVLS
```

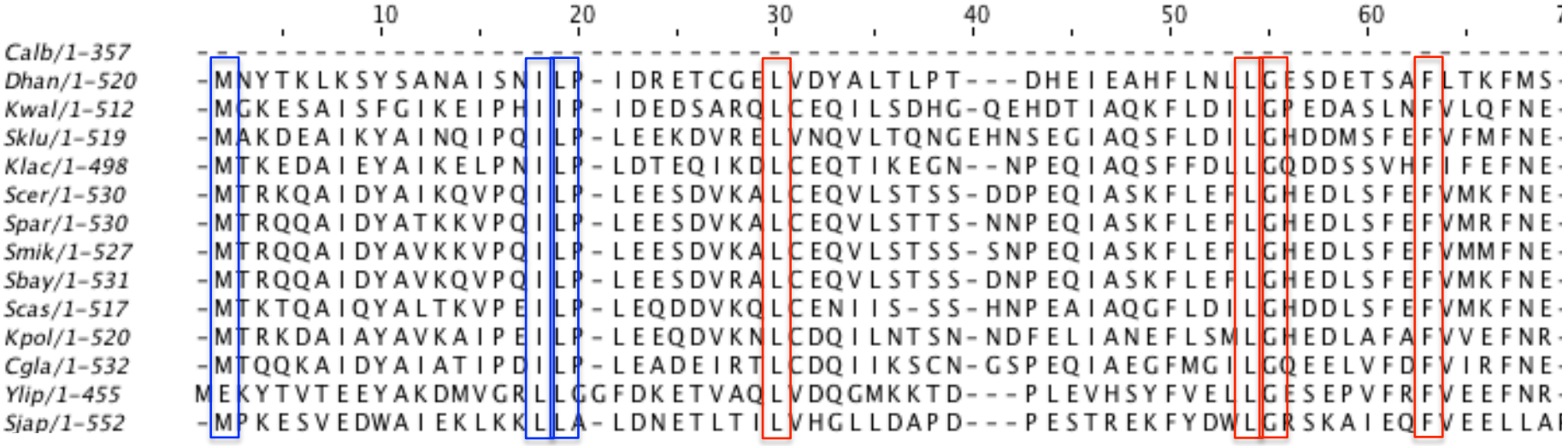
What is multiple sequence alignment?

Given: three or more related biological sequences

Goal: identify the subsets of positions across sequences that are truly related

In other words: find a simultaneous alignment of all input sequences such that the implied pairwise alignments identify the truly related positions between each pair of sequences

An example multiple sequence alignment



Why multiple sequence alignment?

- Build phylogenetic trees (next module)
  - Determine evolutionary relationships between sequences
- A multiple sequence alignment can represent a family of proteins with similar function
  - Compare new sequence to a "family" of known proteins
  - For example the BLOCKS database used for BLOSUM contains several ungapped alignments for known protein families
- Discover common signatures or protein domains among a group of proteins
- Identify genetic variation among individuals of a population

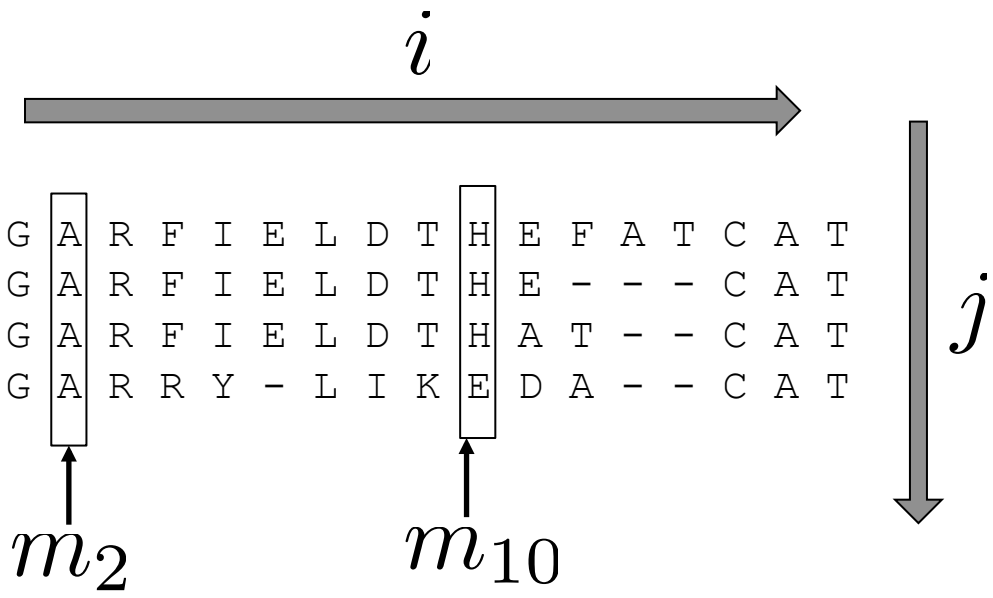
The tasks in Multiple Sequence Alignment

- Scoring an alignment
- Algorithms for creating an alignment

Notation

- Let m denote a Multiple Sequence Alignment
- m_i is the i^{th} column of the alignment m
- m_i^j is the i^{th} column and j^{th} row
- c_i^a count of residue a in column i

Example using notation



$$m_3^1 = R$$

$$c_3^A = 0$$
$$c_2^A = 4$$
$$c_{10}^H = 3$$

Scoring a Multiple Sequence Alignment (MSA)

- Key issue: how do we score a multiple sequence alignment?
- Usually, we assume that *columns* of an alignment are independent

$$Score(m) = G(m) + \sum_i S(m_i)$$

gap function

score of i^{th} column

- For now, we will simplify the score by assuming a linear gap penalty

$$Score(m) = \sum_i S(m_i)$$

Gap penalty

- We will use a simple linear gap penalty function
  - Penalty for a space: s
- Let S(a,b) denote the cost of substituting a by b.
- Linear gap penalty can be incorporated into the substitution matrix
  - S(a,-) = -s = S(-,a)
  - S(-,-) = 0

Two common ways of scoring a multiple alignment

- Entropy based scores
- Sum of pairs

Entropy of a distribution

- A measure of uncertainty of an outcome
- For a discrete distribution $P(X)$, where X takes k values x_1, .. x_k it is defined as

$$H(X) = -\sum_{i=1}^{k} P(x_i)\log P(x_i)$$

- Entropy is greatest when we are most uncertain, that is, for a uniform distribution
- Entropy is least when we are most certain, e.g. deterministic event

Score of a column: Entropy based

- Score of the i^{th} column of alignment m is

$$S(m_i) = -\sum_a c_i^a \log(p_{ia})$$

p_{ia} : Probability of character a in column i
c_i^a : Number of occurrences of a in column i

- This has an entropy-based interpretation
  - Let X_i be a random variable representing a character in column i
  - Consider each entry of column i to be observations of X_i across multiple independent experiments
  - We estimate $P(X_i = a)$ by $p_{ia} = \frac{c_i^a}{n}$
  - Column score is proportional to the entropy of X_i

Scoring an alignment: Entropy based score

- High entropy: More uniform distribution/more variability of characters
- Low entropy: Less uniform distribution/less variability of characters

$$S(m_i) = -\sum_a c_i^a \log(p_{ia})$$

Scoring of a column: Sum of Pairs

- Define the sum of the pairwise scores (SP scores) as:

$$S(m_i) = \sum_{k<l} s(m_i^k, m_i^l)$$

Iterate over all pairs of rows in the column

- Columns are scored by a 'sum of pairs' (SP) function using:

$s(m_i^k, m_i^l)$ Substitution score from a substitution/match matrix such as BLOSUM or PAM

Algorithms for performing a Multiple Sequence Alignment (MSA)

- Dynamic programming
  - Not practical
- Progressive alignment algorithms
  - Star alignment
  - Guide tree approach
- Iterative alignment algorithms

Generalizing Pairwise to Multiple Alignment

- Alignment of 2 sequences is a 2D matrix.
- Alignment of 3 sequences is a 3D matrix

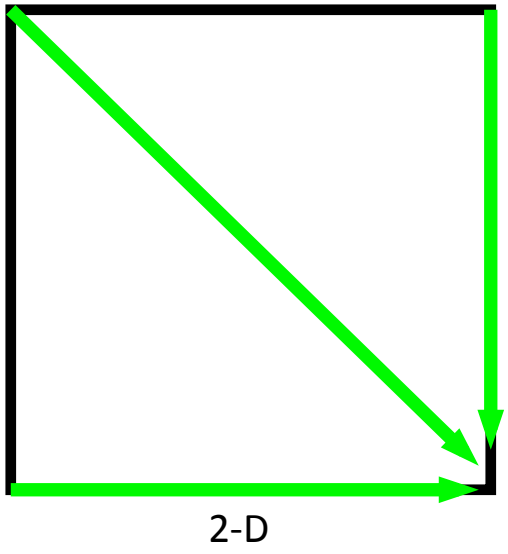
```
A T - G C G -
A - C G T - A
A T C A C - A
```

- Our scoring function should score alignments with conserved columns higher.

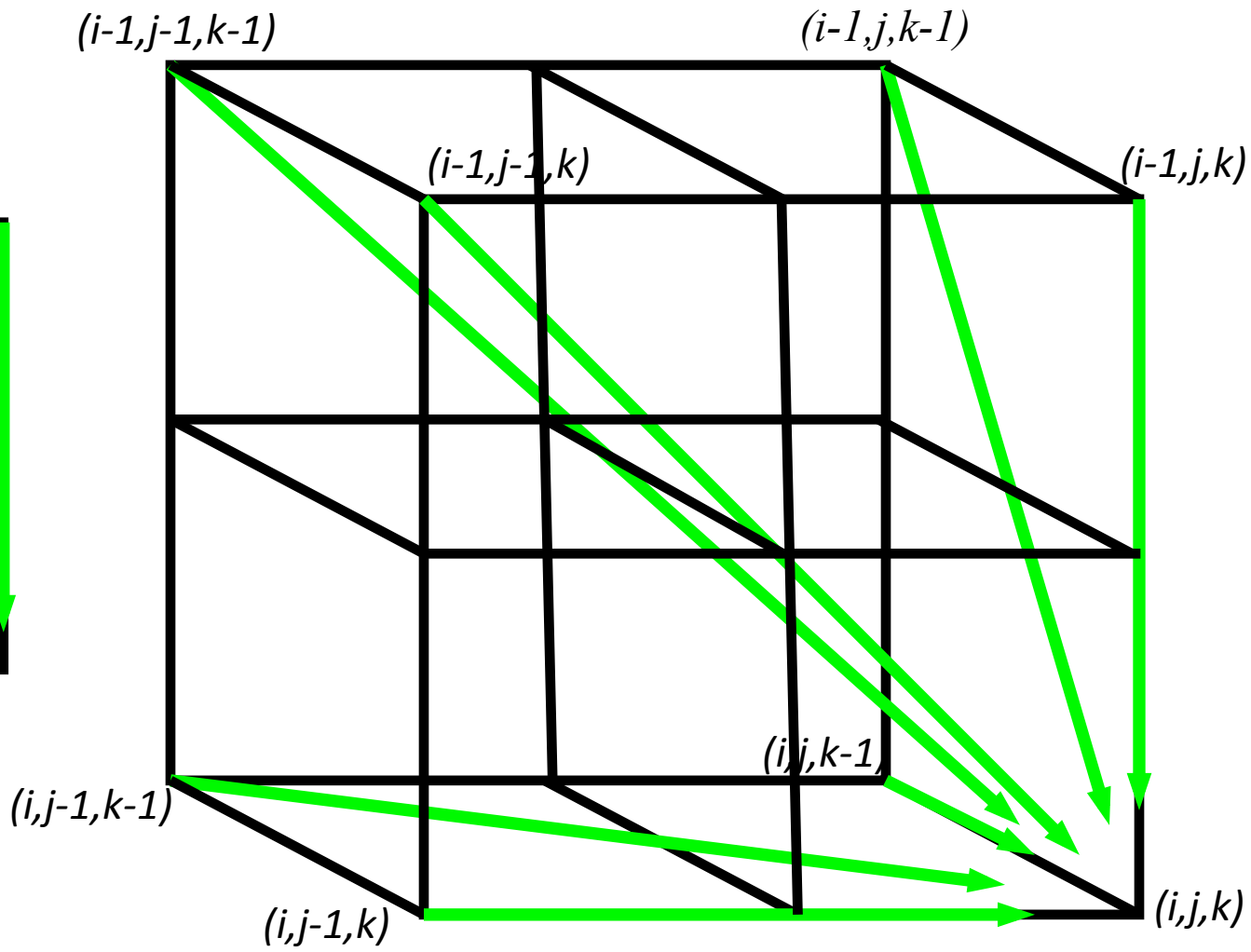
Dynamic Programming (DP) for global multiple sequence alignment

- Assume columns are independent
  - Score of alignment is sum of column scores
- Generalization of methods for pairwise alignment
  - Consider d-dimensional matrix for k sequences (instead of 2-dimensional matrix)
  - each matrix element represents alignment score for k prefixes (instead of 2 prefixes)

2-D Alignment Cell versus 3-D Alignment Cell



2-D



Notation for DP

- Assume we have k sequences $x^1, \cdots, x^k$
- i_1 denotes the length of the prefix for sequence 1
- i_2 denotes the length of the prefix for sequence 2
- …
- i_k denotes the length of the prefix for sequence k
- $x^k_{i_k}$ denotes the character at i_k position of sequence x^k
- F: k-dimensional matrix where

$$F(i_1, i_2, \cdots, i_k)$$

denotes the score of the best alignment of the i_1, i_2.. i_k prefixes of the sequences

Recall the DP for the pairwise alignment

$$F(i_1, i_2) = \max \begin{cases} F(i_1 - 1, i_2 - 1) + S(x^1_{i_1}, x^2_{i_2}) \\ F(i_1, i_2 - 1) + S(-, x^2_{i_2}) \\ F(i_1 - 1, i_2) + S(x^1_{i_1}, -) \end{cases}$$

DP for Multiple sequence alignment

$$F(i_1,\cdots,i_k)=\max\begin{cases}F(i_1-1,\cdots,i_k-1)+S(x^1_{i_1},\cdots,x^k_{i_k})\\ F(i_1,i_2-1,\cdots,i_k-1)+S(-,x^2_{i_2},\cdots,x^k_{i_k})\\ F(i_1-1,i_2,\cdots,i_k-1)+S(x^1_{i_1},-,\cdots,x^k_{i_k})\\ \vdots\\ F(i_1,i_2-1,\cdots,i_k)+S(-,x^2_{i_2},\cdots,-)\\ \vdots\end{cases}$$

max score of alignment for the k prefixes

Every possible pattern for the gaps

How many items do we need to maximize over? $2^k - 1$

DP Multiple Alignment: Running Time

- For 3 sequences of length *n*, the run time is proportional to $7n^3$

- For a *k*-way alignment, each k of length n
  - n^k nodes
  - most nodes have $2^k - 1$ incoming edges.
  - Space complexity(n^k)
  - Runtime: O($2^k n^k$)
- Too slow

MSA Sum of Pairs: Example

- Sequences: GATTCA, GTCTGA, GATATT, GTCAGC.

- 6 pairwise alignments (premium for match +1, penalties for indels and mismatches -1)

```
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)
```

Heuristic algorithms to Multiple sequence alignment

- Progressive alignment
  - Build the alignment of larger number of sequences from partial alignments of subsets of sequences
- Iterative alignment
  - Possibly remove some of the aligned sequences and re-align to see if score improves

Progressive Alignment

- Key heuristic: Align the "most similar" sequences first
- Rely on pre-computed pairwise similarity/distance
  - Pairwise sequence alignments
  - Algorithms differ in the extent to which the pairwise similarity influences the final alignments
- Two strategies
  - Star alignment
  - Tree alignments
    - Simple (quick and dirty) tree
    - At each time combine two, possibly singleton, sets of sequences

Progressive Alignment

- Build a multiple sequence alignment up from pairwise alignments.

Start with an alignment between S_c and some other sequence:

```
SC YFPHFDLSHGSAQVKAHGKKVGDALTLAVGHLDDLPGAL
S1 YFPHFDLSHG-AQVKG--KKVADALTNAVAHVDDMPNAL
```

Add 3rd sequence, say S2, and use the SC - S2 alignment as a guide, adding spaces into the MSA as needed.

SC - S2 alignment:

```
SC YFPHF-DLS-----HGSAQVKAHGKKVGDALTLAVGHL----DDLPGAL
S2 FFPKFKGLTTADQLKKSADVRWHAERII----NAVNDAVASMDDTEKMS
```

New {SC, S1, S2} alignment (red gaps added in S1):

```
SC YFPHF-DLS-----HGSAQVKAHGKKVGDALTLAVGHL----DDLPGAL
S1 YFPHF-DLS-----HG-AQVKG--KKVADALTNAVAHV----DDMPNAL
S2 FFPKFKGLTTADQLKKSADVRWHAERII----NAVNDAVASMDDTEKMS
```

Continue with S3, S4, ...

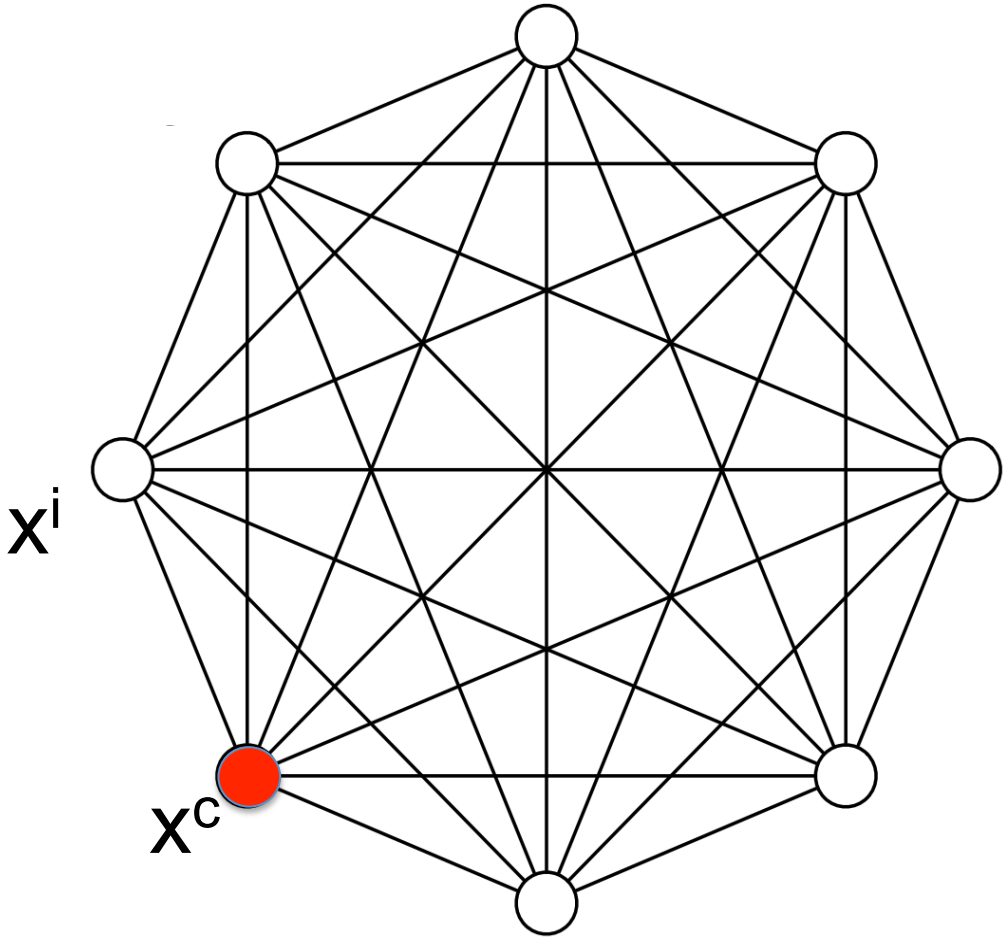
Star Alignment Approach

- Given: k sequences to be aligned

$$x^1, \cdots, x^k$$

  - pick one sequence x^c as the "center"
  - for each $x^i \neq x^c$ determine an optimal alignment between x^i and x^c
  - Aggregate pairwise alignments
    - Shift entire columns when incorporating gaps
- return: multiple alignment resulting from aggregate

Star Alignment

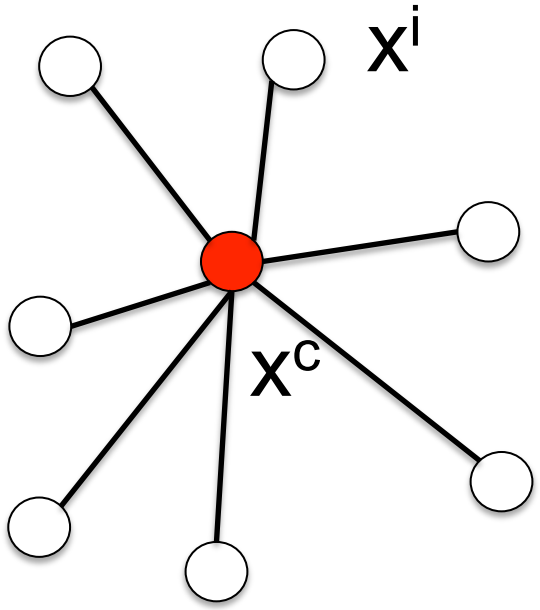


Picking the center in star alignments

Two possible approaches:

1. try each sequence as the center, return the best multiple alignment
2. compute all pairwise alignments and select the string x^c that maximizes:

$$\sum_{i \neq c} \text{sim}(x^i, x^c)$$



Aligning to an existing partial alignment

- Need to treat each "partial alignment" as a single entity
  - Partial alignment should not be changed other than gap insertions
- Shift entire columns when incorporating gaps

Star Alignment Example

Given:

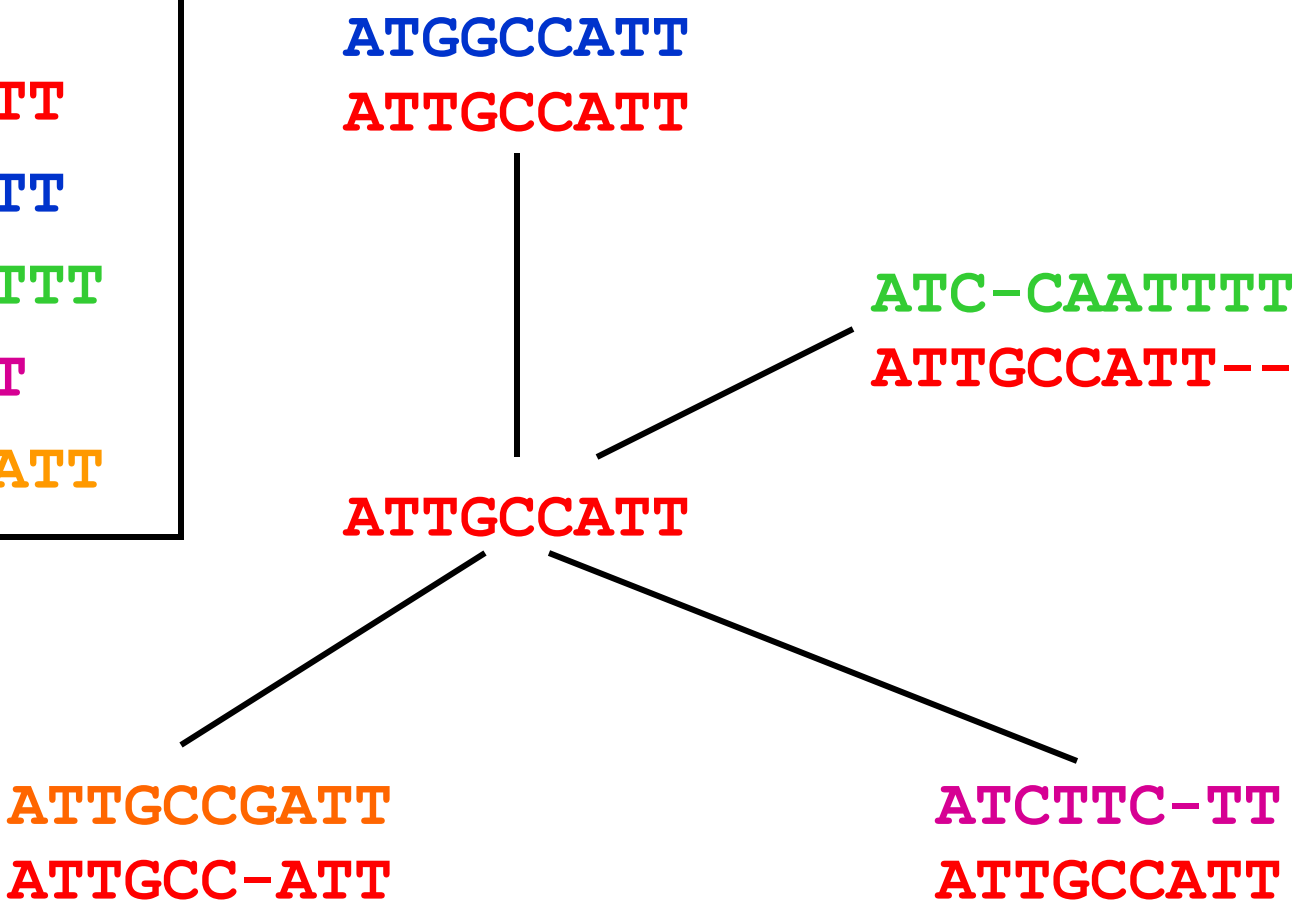
ATTGCCATT

ATGGCCATT

ATCCAATTTT

ATCTTCTT

ATTGCCGATT



Star Alignment Example

- Aggregate pairwise alignments

	present pair	Current multiple alignment
1.	ATGGCCATT ATTGCCATT	ATTGCCATT ATGGCCATT
2.	ATC-CAATTTT ATTGCCATT--	ATTGCCATT-- ATGGCCATT-- ATC-CAATTTT

Star Alignment Example

present pair

Current multiple alignment

3.

```
ATCTTC-TT
ATTGCCATT
```

```
ATTGCCATT--
ATGGCCATT--
ATC-CAATTTT
ATCTTC-TT--
```

4.

```
ATTGCCGATT
ATTGCC-ATT
```

```
ATTGCC- A TT--
ATGGCC- A TT--
ATC-CA- A TTTT
ATCTTC- - TT--
ATTGCCG A TT--
```

shift entire columns
when incorporating a gap

Comments about Star alignment

- Conceptually simple
- Dependent only upon pairwise alignments
- Does not consider any position-specific information of the partial multiple sequence alignment while aligning a new sequence to it

Tree-based progressive alignments

- Align sequences according to a *guide tree*
  - leaves represent sequences
  - internal nodes represent alignments
- Determine alignments from bottom of tree upward
  - return multiple alignment represented at the root of the tree
- One common variant: the CLUSTALW algorithm [Thompson et al. 1994]

Tree-based progressive alignment

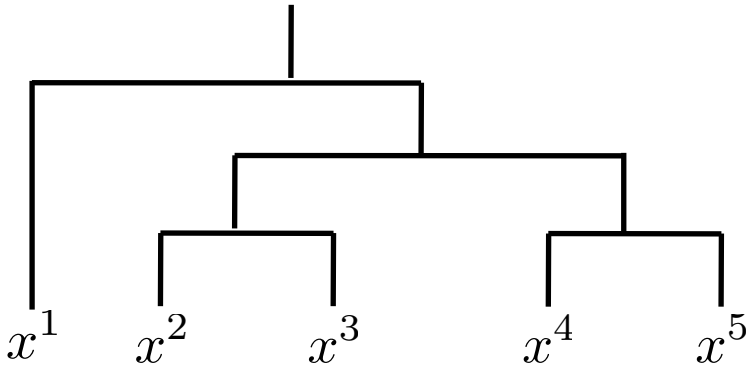
- Depending on the internal node in the tree, we may have to align a
  - a sequence with a sequence
  - a sequence with a partial alignment
  - a partial alignment with a partial alignment
- In all cases we have the option of inserting gaps or substitutions
  - For aligning alignments, we will use sum of pairs scoring
  - To choose between options we will use an idea similar to the pairwise sequence alignment case

Tree alignment example

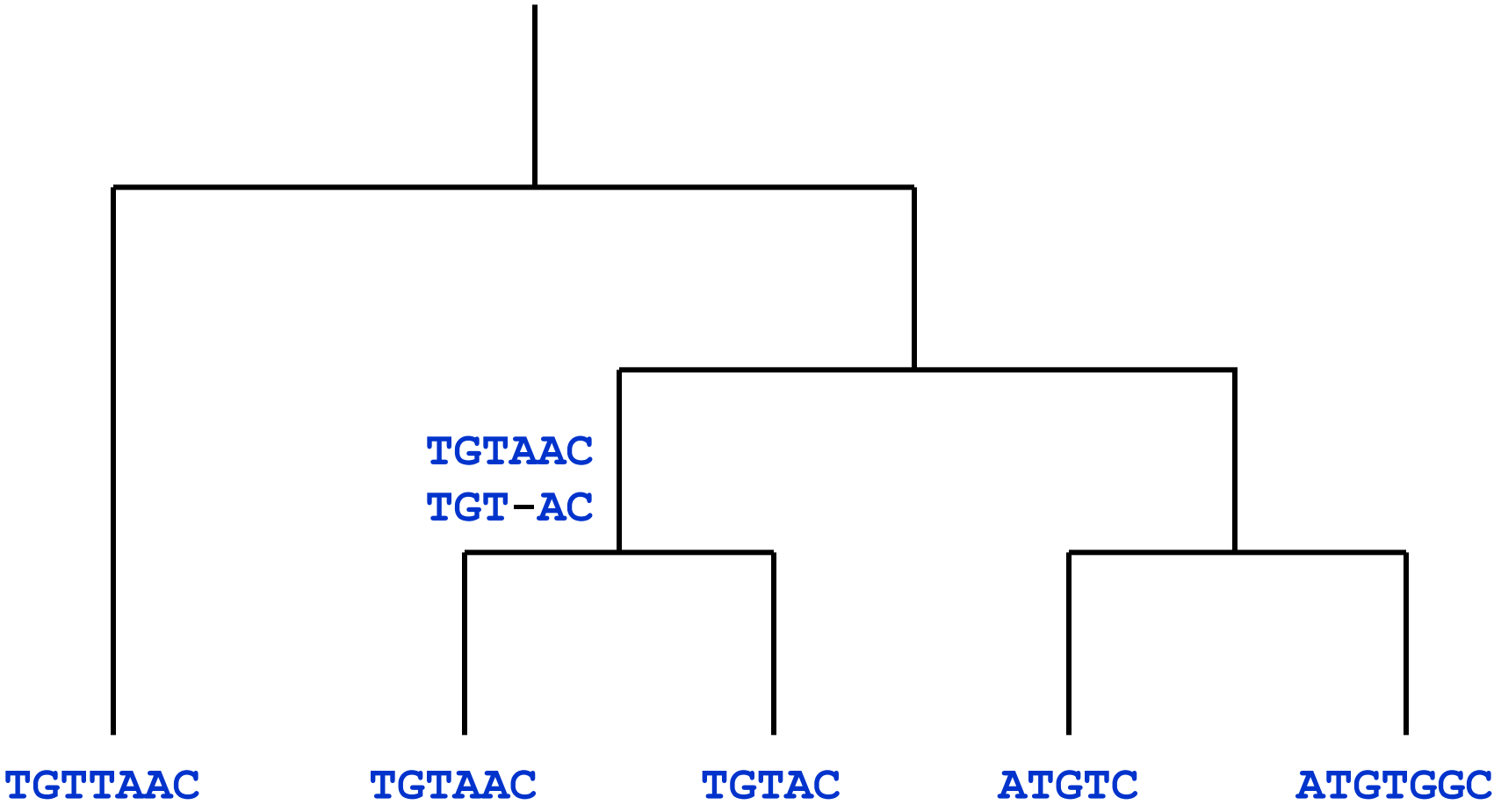
- Starting sequences

x^1 **TGTTAAC**
x^2 **TGTAAC**
x^3 **TGTAC**
x^4 **ATGTC**
x^5 **ATGTGGC**

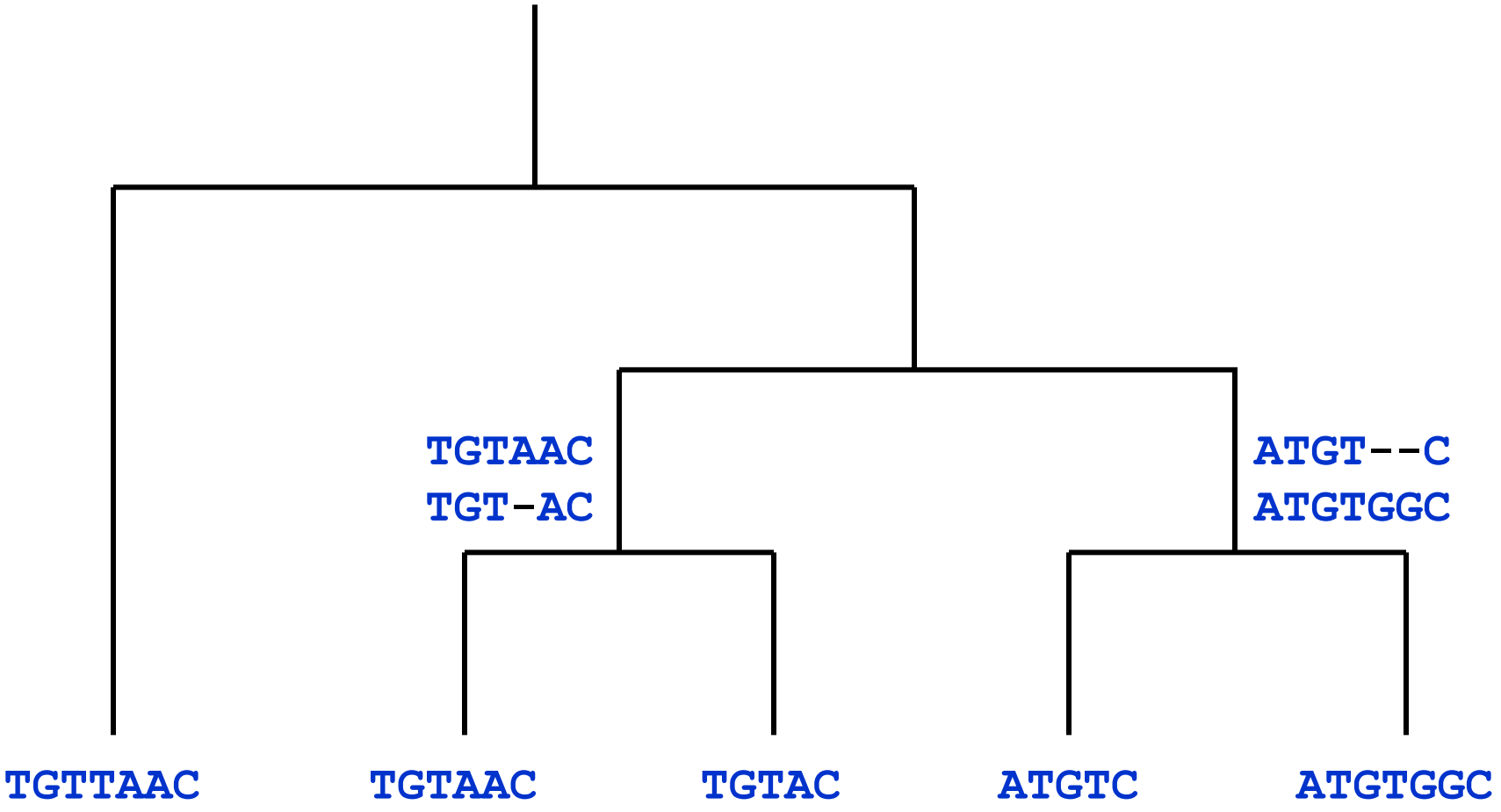
- Create a guide tree
  - Using pairwise distances (we will cover this in subsequent lectures)
  - Approach similar to but simpler than phylogenetic trees



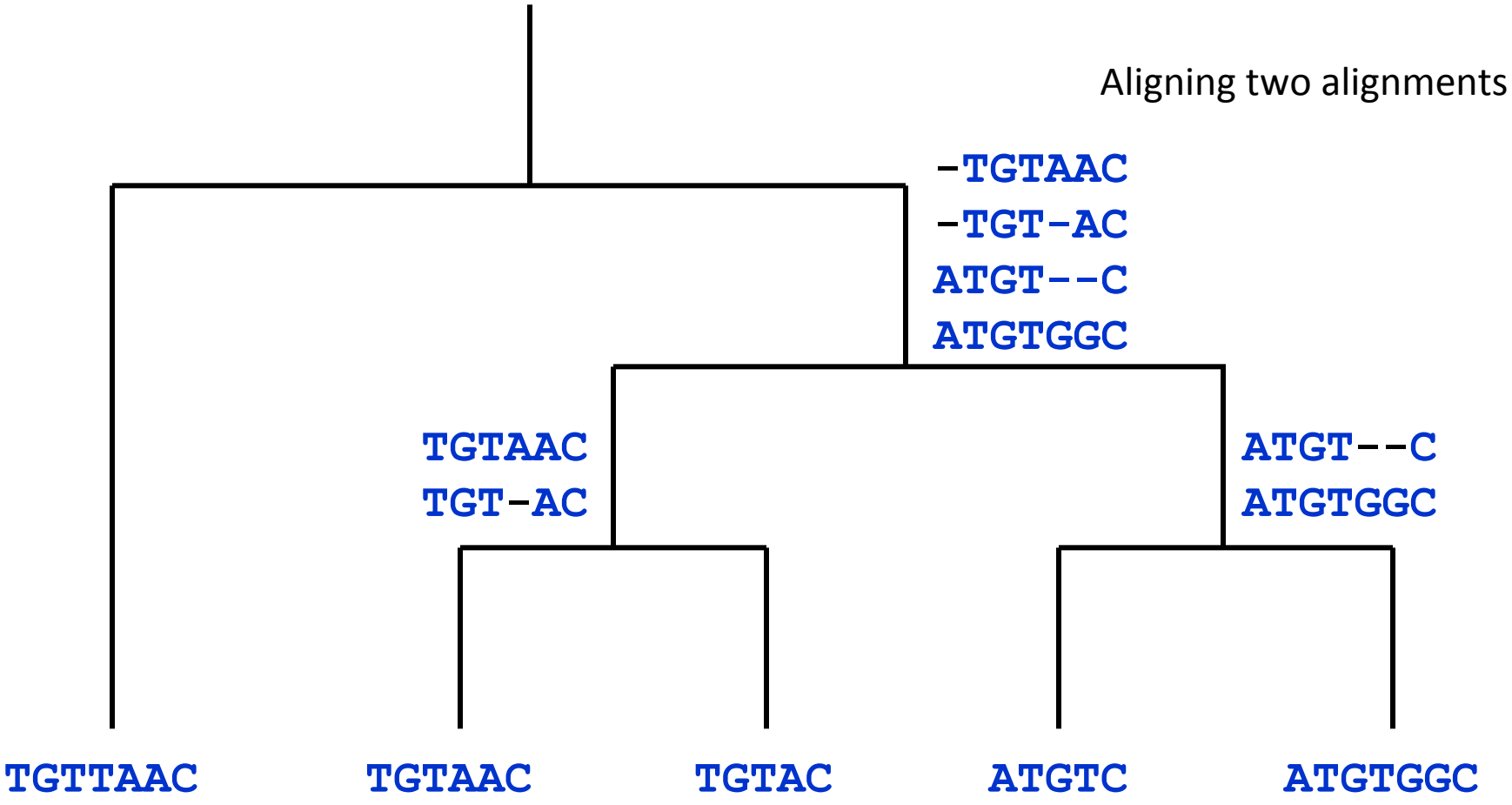
Tree Alignment Example



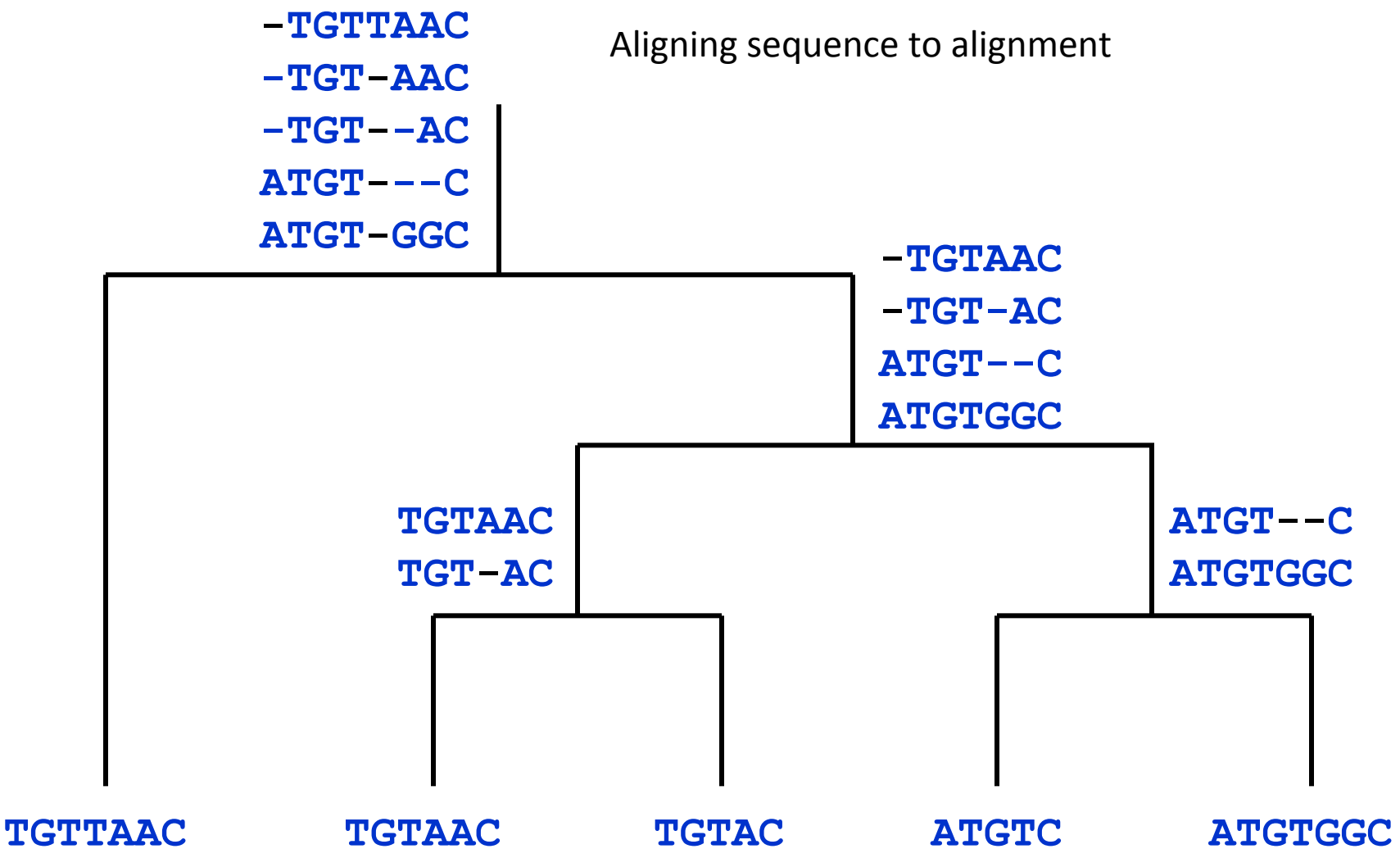
Tree Alignment Example



Tree Alignment Example



Tree Alignment Example



Scoring an alignment of partial alignments

- Recall the sum of pairs score for a column i

$$S(m_i) = \sum_{k<l} s(m_i^k, m_i^l)$$

- Let 1 to n represent sequences from the first alignment
- Let $n+1$ to N represent sequences from the second alignment, N denotes total number of sequences
- Alignment at column i can be written as

$$S(m_i) = \sum_{k<l\leq n} s(m_i^k, m_i^l) \quad \text{Within first alignment}$$

$$+ \sum_{n<k<l\leq N} s(m_i^k, m_i^l) \quad \text{Within second alignment}$$

$$\sum_{k\leq n, n<l\leq N} s(m_i^k, m_i^l) \quad \text{Between two alignments}$$

Computing the sum of scores for two alignments

- Assume we have two alignments corresponding to intermediate nodes of the guide tree

Alignment A1	Alignment A2
AAAC	AGC
-GAC	ACC

- Alignment of two alignments = pairwise alignment of sequences of *columns*
- Filling entry *(i, j)* of the DP matrix we maximize over
  - aligning column i in A1 to a column j in A2
  - aligning column i in A1 to gaps in A2
  - aligning column j in A2 to gaps in A1

Comments about tree-based progressive alignment

- Exploits partial alignment information
- But, greedy
  - The tree might not be correct - may reflect an incorrect ordering of how sequences should be stacked up in the alignment
  - Final results prone to errors in alignment
    - Some positions might be misaligned (have a lower score than if a different ordering is used).

Ordering matters

Consider aligning GG, DGG and DGD

1	2
D G D	D G D
– G G	G G –

Both are equally good. But when we include DGG

1	2
D G D	D G D
– G G	G G –
D G G	D G G

1 is better than 2, assuming a match score of 2, mismatch score =1, gap penalty=-2

Profiles

- Another way to summarize MSA

x^1 ACG-TT-GA
x^2 ATC-GTCGA
x^3 ACGCGA-CC
x^4 ACGCGT-TA

Column in the alignment

Character	1	2	3	4	5	6	7	8	9
A	1	0	0	0	0	0.25	0	0	0.75
C	0	0.75	0.25	0.5	0	0	0.25	0.25	0.25
G	0	0	0.75	0	0.75	0	0	0.5	0
T	0	0.25	0	0	0.25	0.75	0	0.25	0
-	0	0	0	0.5	0	0	0.75	0	0

Fraction of time given column had given character

Profile based alignment

gap in profile introduced to better fit sequence

R =

	1	2	3	4	5	6	7	8	9
A	1	0	0	0	0	0.25	0	0	0.75
C	0	0.75	0.25	0.5	0	0	0.25	0.25	0.25
G	0	0	0.75	0	0.75	0	0	0.5	0
T	0	0.25	0	0	0.25	0.75	0	0.25	0
-	0	0	0	0.5	0	0	0.75	0	0

A C C - A G A C G A

Score of matching character x with column j of the profile:

$$P(x,j) = \sum_{c \in \Sigma} \text{sim}(x,c) \times R[c,j]$$

sim(x,c) = how similar character x is to character c.

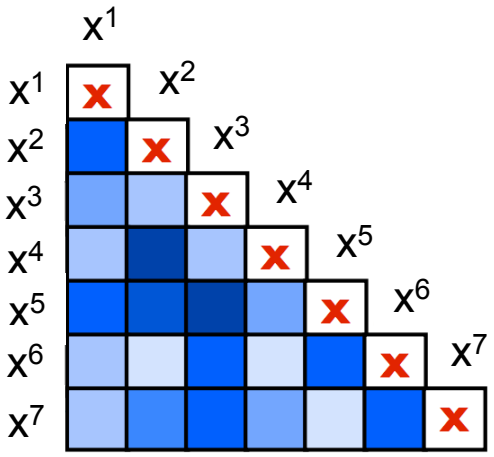
$$F(i,j) = \max \begin{cases} F(i-1, j-1) + P(x_i, j) & \text{align } x_i \text{ to column } j \\ F(i-1, j) + \text{gap} & \text{introduce gap to profile} \\ F(i, j-1) + P(-, j) & \text{introduce gap to x} \end{cases}$$

Iterative refinement methods

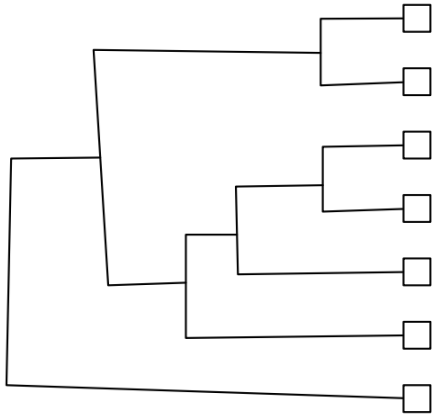
- The order of selection of sequences can influence the alignment
  - ClustalW overcomes some of these issues but has many heuristics and parameters
- How to avoid committing to a non-optimal pairwise decision?
  - Revisit alignments
  - This is the focus of iterative alignments
- Basic iterative refinement algorithm
  - Remove a sequence from the current multiple alignment
  - Realign the removed sequence back to the multiple alignment
  - Repeat until removal and realignment of any sequence does not improve the alignment score

CLUSTALW

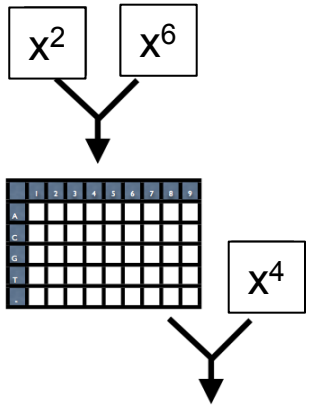
- CLUSTALW is a widely used heuristic multiple aligner
- Not the fastest, nor the most accurate, but pretty good
- Many heuristics used:



Step (1): Build pairwise distance matrix



Step (2): Build guide tree



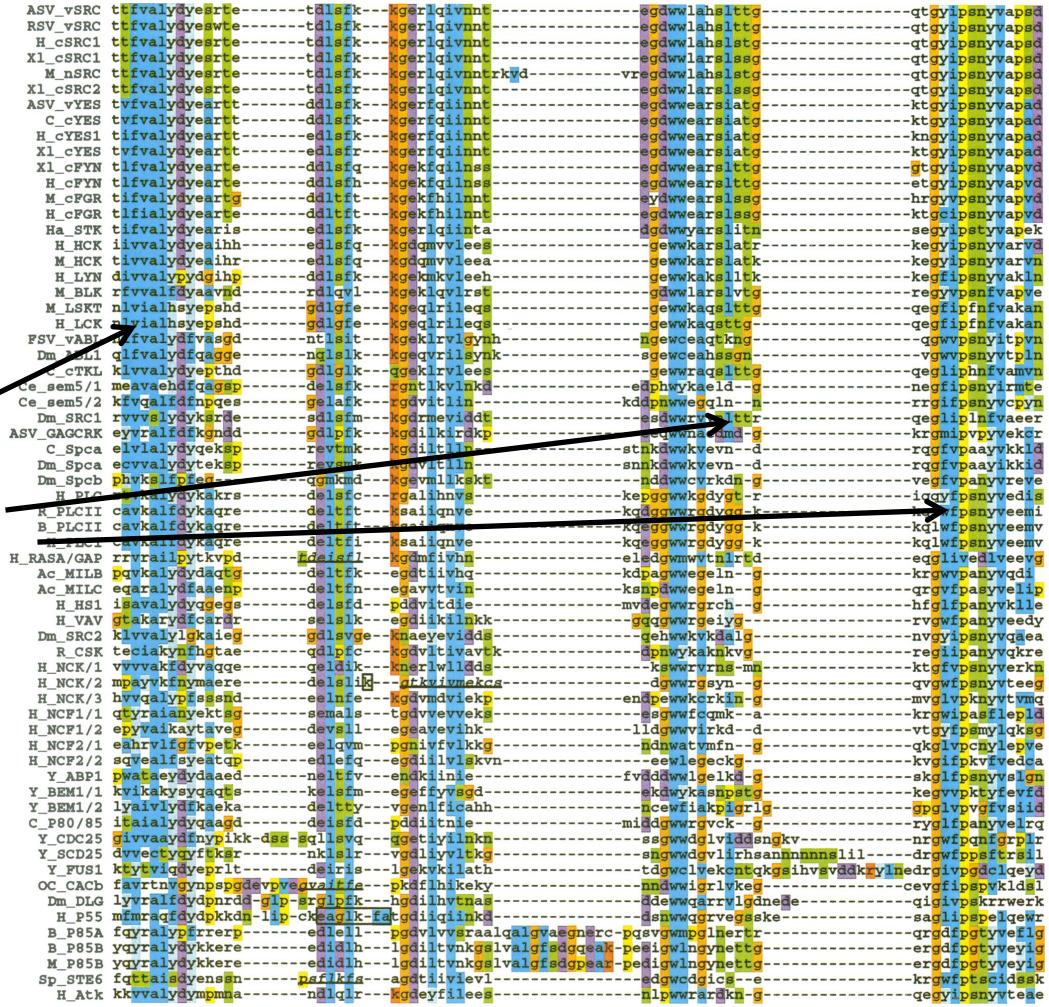
Step (3): Align sequences / **sets of sequences** from the most similar to least similar

Additional notes about the ClustalW algorithm

- Tailored to handle very divergent sequences: 25-30% similarity
- Dynamically varies the gap penalties in a position and residue specific manner
- Weight different sequences differently
  - Closely related sequences need to be down-weighted
  - Divergent sequences are up-weighted
- Dynamically switch between substitution matrices depending upon the average similarity between sequences being aligned

Applying ClustalW to SH3 domain proteins

Alignment blocks correspond to beta strand secondary structures



Proteins share <12% sequence identity

Thomson et al, 1994

Summary

- Multiple sequence alignment is the problem of finding corresponding positions among more than two sequences
- Scoring function:
  - Entropy based
  - Sum of pairs
- Algorithms
  - Progressive
    - Star
      - Dependent upon a center
      - Keep adding all pairs of aligned sequences with the current alignment
    - Tree
      - Create an approximate guide tree
      - Use tree to align the sequences
  - Iterative
    - Don't commit to the fixed ordering, revisit the alignment until score doesn't change