

Global Multiple Sequence Alignment

- Given sequences s_1, \dots, s_k of lengths n_1, \dots, n_k
 seek s'_1, \dots, s'_k of length $l \geq \max\{n_i\}$ such that
- Obtain s_i from s'_i by removing gaps
 - No column contains all gaps
 - The score of the alignment is optimal

Global Multiple Sequence Alignment

```

HUMAN MKWVTFISLL FLSSAYSRG V..FRRDA.H KSEVAHRFKD LGEENFKALV
RABIT MKWVTFISLL FLSSAYSRG V..FRREA.H KSEIAHRFND VGEEHFGLV
PIG   ~-WVTFISLL FLSSAYSRG V..FRRDT.Y KSEIAHRFKD LGEQYFKGLV
CHICK MKWVTLISFI FLSSATSRN LQRFARDAEH KSEIAHRYND LKEETFKAVA
  
```

- Align k sequences, so that residues in each column share a property of interest:
- a common ancestor
 - a structural or functional role

Applications Global Multiple Alignment




```

HUMAN MKWVTFISLL FLSSAYSRG V..FRRDA.H KSEVAHRFKD LGEENFKALV
RABIT MKWVTFISLL FLSSAYSRG V..FRREA.H KSEIAHRFND VGEEHFGLV
PIG   ~-WVTFISLL FLSSAYSRG V..FRRDT.Y KSEIAHRFKD LGEQYFKGLV
CHICK MKWVTLISFI FLSSATSRN LQRFARDAEH KSEIAHRYND LKEETFKAVA
  
```

- Protein structure and function
- RNA structure
- Evolutionary tree reconstruction

Scoring function: Sum-of-Pairs

$$\text{Score} = \sum_{x=1}^k \sum_{y>x} d(s'_x[j], s'_y[j])$$


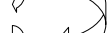
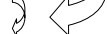
- (1) **AG_CT** 
- (2) **AG_CT** 
- (3) **ACT_T** 

$$\begin{aligned} \text{Score} &= p[s_1, s_2] + p[s_1, s_3] + p[s_2, s_3] \\ &= M + m + m = 2m + M \end{aligned}$$

Note: this example uses a similarity function. We can also use Sum-of-Pairs with distance scoring.

Scoring function: Sum-of-Pairs

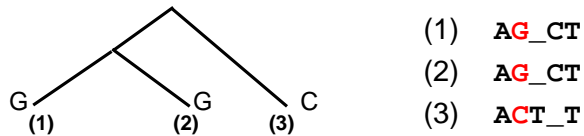
$$\text{Score} = \sum_{x=1}^k \sum_{y>x} d(s'_x[j], s'_y[j])$$

- (1) **AG_CT** 
- (2) **AG_CT** 
- (3) **ACT_T** 

$$\begin{aligned} \text{Score} &= p[s_1, s_2] + p[s_1, s_3] + p[s_2, s_3] \\ &= M + g + g = 2g + M \end{aligned}$$

Note: this example uses a similarity function. We can also use Sum-of-Pairs with distance scoring.

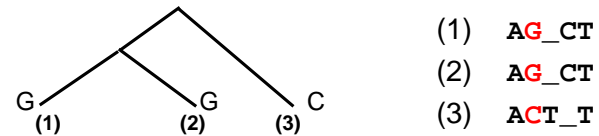
Tree Alignment



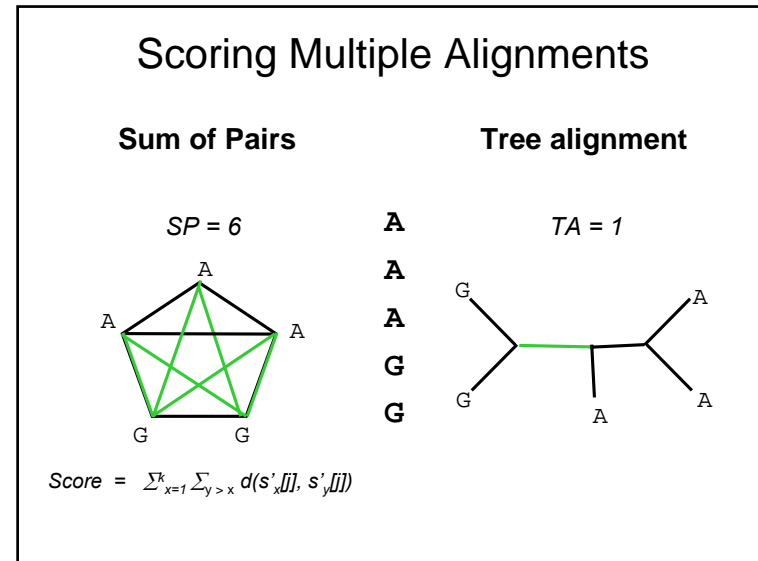
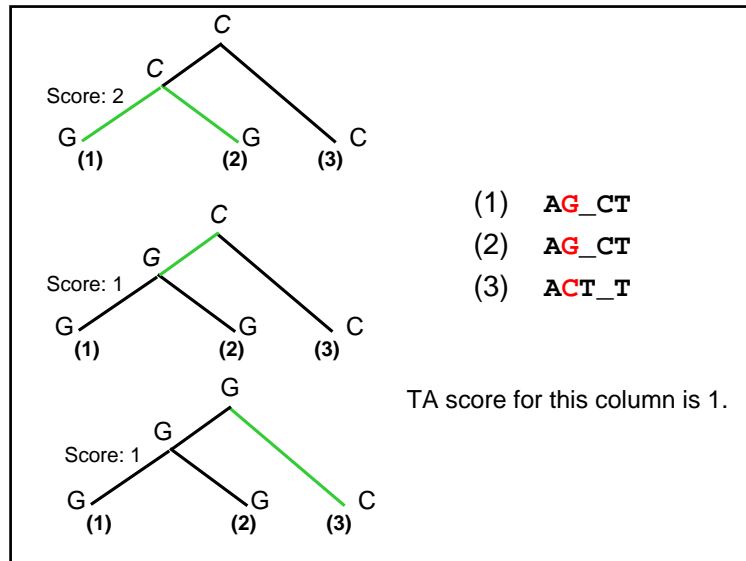
Assumptions:

- Sequences (and columns) evolved from a common ancestor
- Evolution is parsimonious: mutations are rare.

Tree Alignment



Score: Given a known tree, the score is the minimum number of mutations required to explain the data.

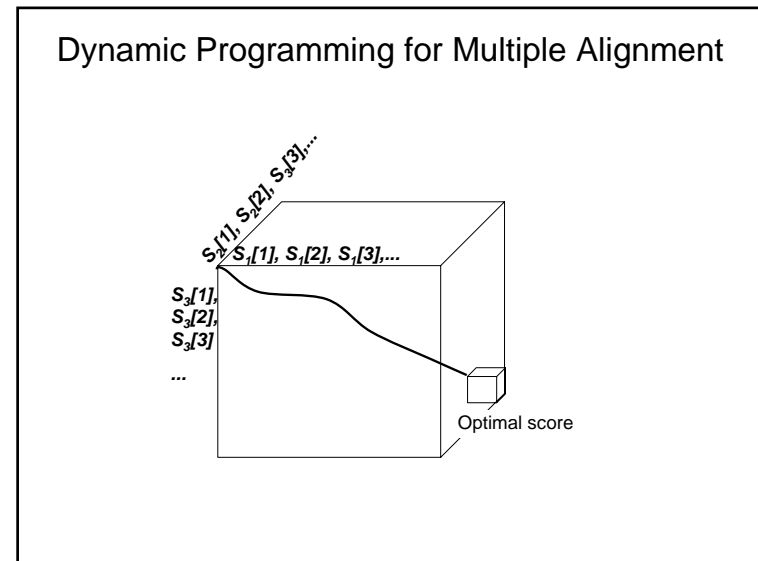


Tree Alignment

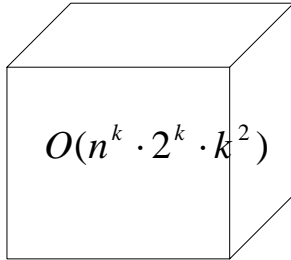
Thought to be “most biological” but

- We don't know the correct tree
- Need to infer sequences on internal nodes
- Columns may not share evolutionary history
- Data may not be parsimonious
- Not always relevant for structural or functional questions

Sum of Pairs is almost always used in practice



Dynamic Programming for Multiple Alignment



MSA is NP-complete for both
 – Sum-of-Pairs
 – Tree alignment

Limits:

- ~ k = 8 - 10 sequences
- ~ n = 500 residues

Progressive Alignment Heuristic

- Compute all pairwise alignments
- Construct a distance matrix, $D[s_i, s_j], \forall i, j > i$
- Construct an ordering from $D[]$, e.g., a “guide tree”
- Use ordering to progressively merge partial MSA’s, using the “once a gap, always a gap” rule.

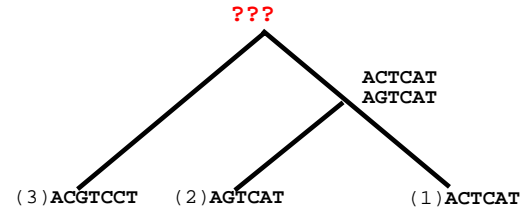
The results depend on the **ordering** and on how partial MSA’s are **merged**.

Optimal Pairwise Alignments

(1) ACTCAT	(1) ACTCAT	3
(2) AGTCAT	(2) AGTCAT	
(3) ACGTCCT	(2) A_GTCAT	5
	(3) ACGTCCT	
	(1) AC_TCAT	5
	(3) ACGTCCT	

$d(x, y) = 3$
 $d(x, _ _) = 2$

Progressive Alignment



- Use *profile alignment* to merge sequences according to a guide tree.
- Typically, most closely related sequences are merged first.

Merging strategy:

Align the profile (1,2) with sequence (3)

- (1) ACTCAT
- (2) AGTCAT
- (3) ACGTCCT

$d(x, y) = 3$
 $d(x, _) = 2$

(1)	ACTCAT	3
(2)	AGTCAT	

- (2) A_GTCAT 5
- (3) ACGTCCT 5

- (1) AC_TCAT 5
- (3) ACGTCCT

_ A C T C A T
 _ A G T C A T

4

_ A
 C
 G
 T
 C
 C
 T



$d(x, y) = 3$
 $d(x, _) = 2$

_ A C T C A T
 _ A G T C A T

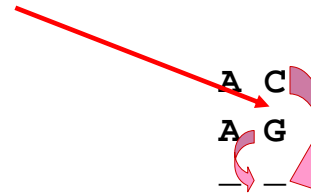
4 8

_ A
 C
 G
 T
 C
 C
 T

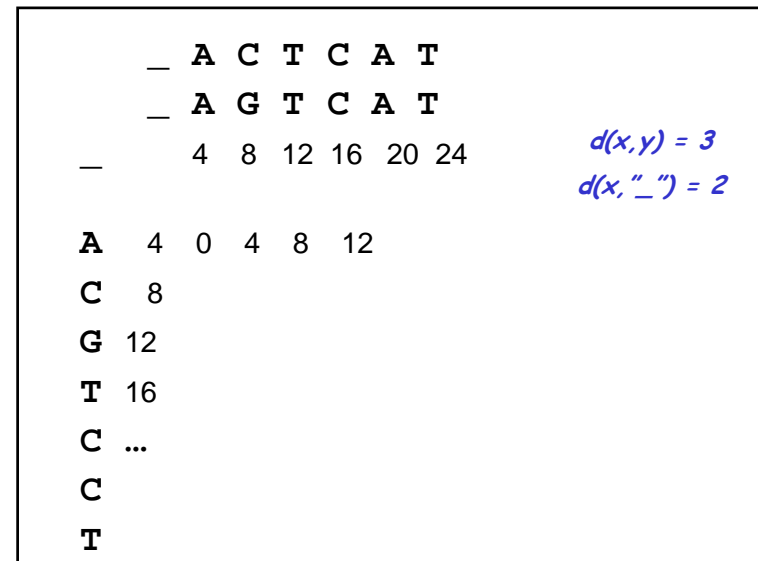
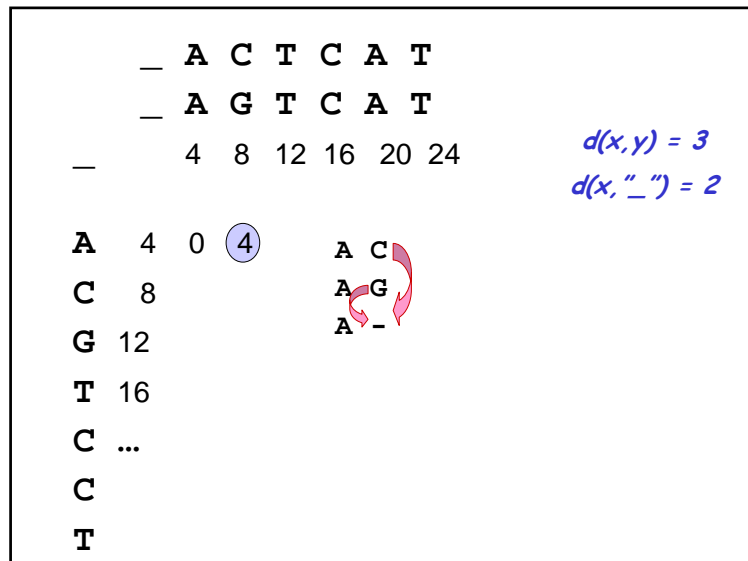
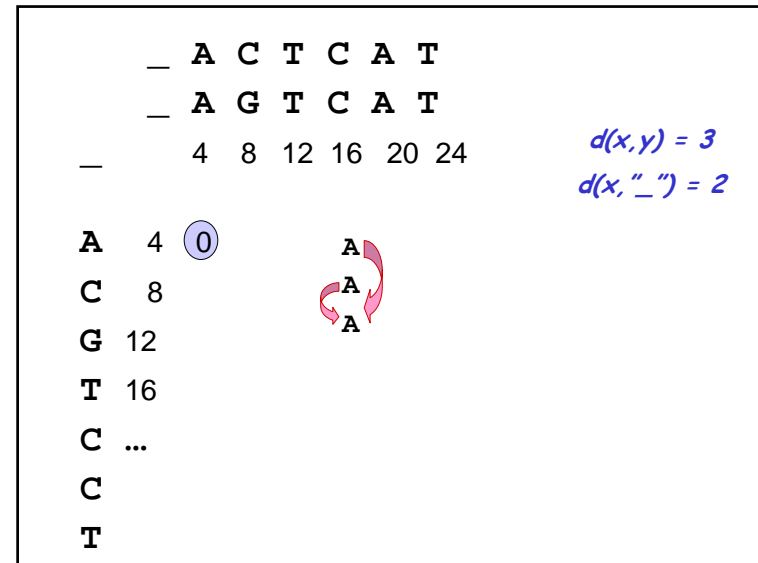
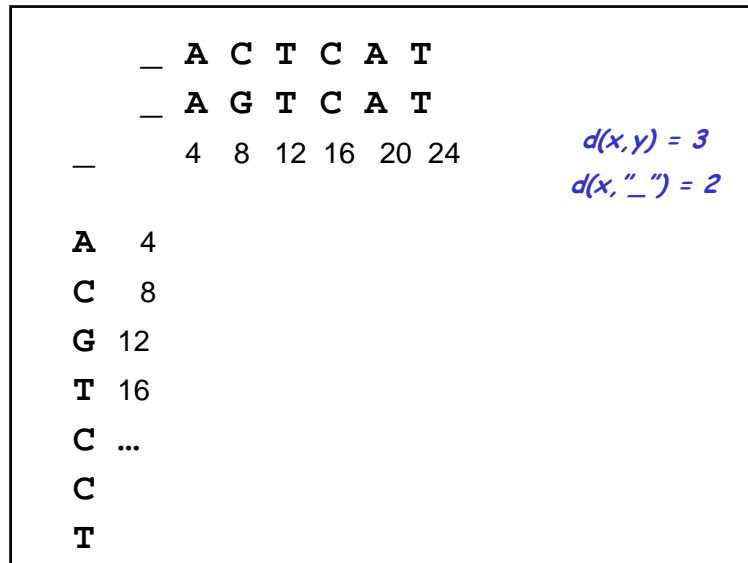


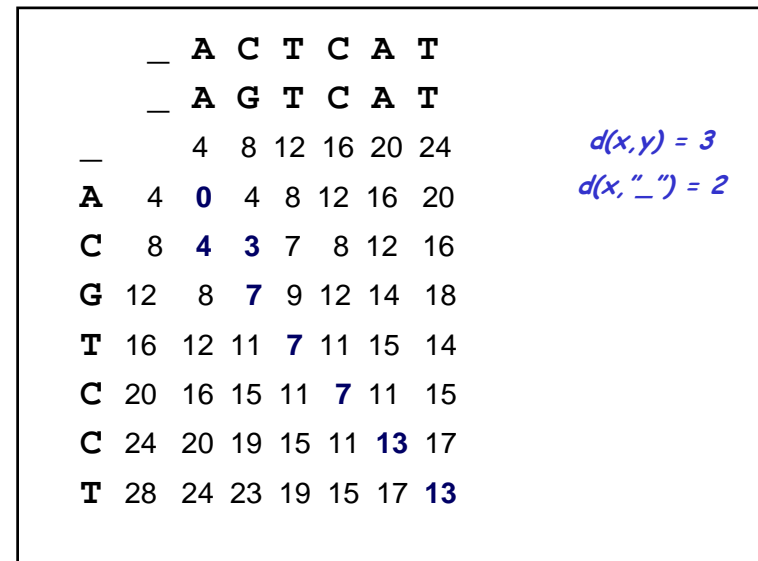
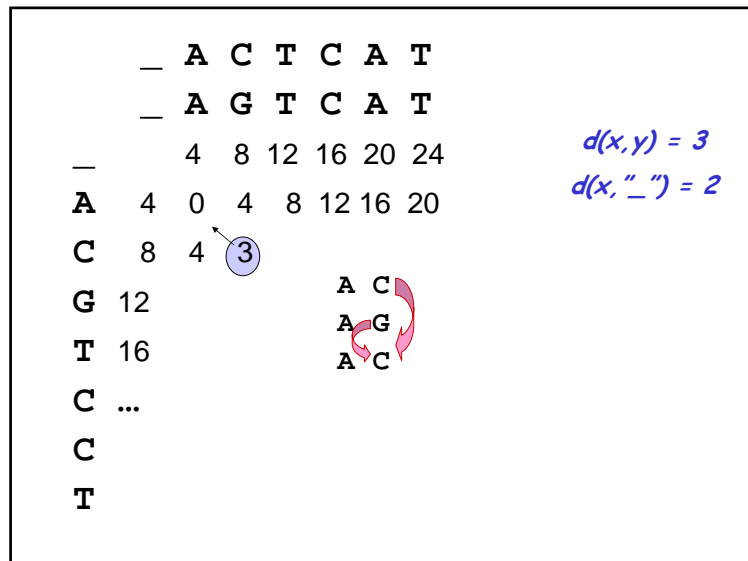
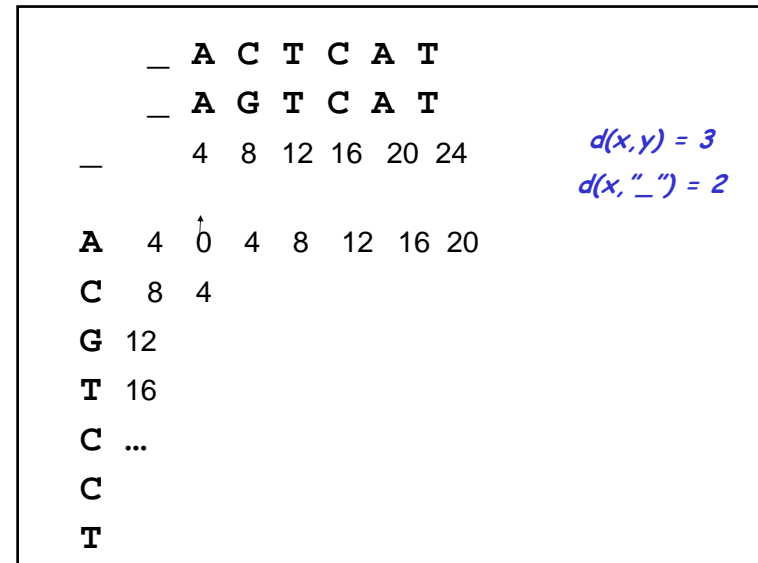
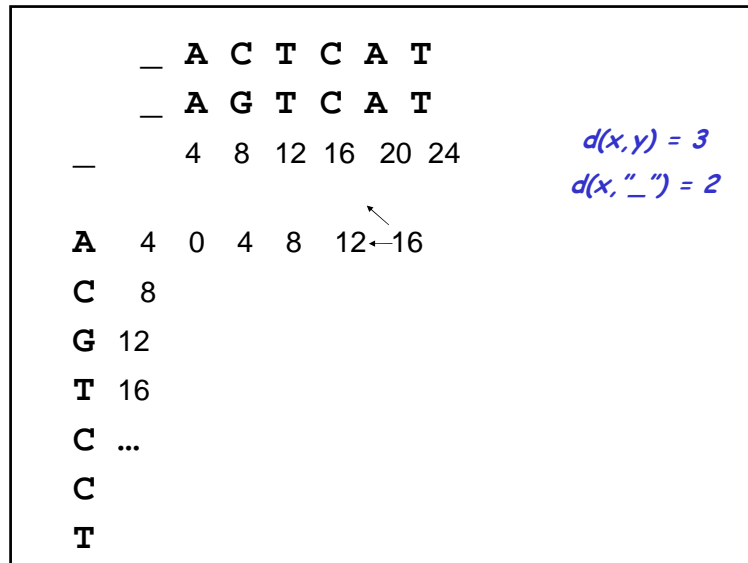
$d(x, y) = 3$
 $d(x, _) = 2$

Note: no penalty for mutations in the profile.
 We paid for those in a previous step



4+4



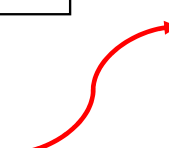


Optimal Pairwise Alignments		Progressive alignment
(1) ACTCAT	(2) AGTCAT	(1,2) + (3)
(2) A_GTCAT	(3) ACGTCCT	(3) ACGTCCT
(1) AC_TCAT	(3) ACGTCCT	(1) AC_TCAT $4m+2g$
(2) A_GTCAT	(2) AG_TCAT	(2) AG_TCAT
(3) ACGTCCT		
		An alternate alignment
(1) AC_TCAT	(2) A_GTCAT	(1) AC_TCAT
(2) A_GTCAT	(3) ACGTCCT	(2) A_GTCAT $2m+4g$
(3) ACGTCCT		(3) ACGTCCT

Optimal Pairwise Alignments		Progressive alignment
(1) ACTCAT	(2) AGTCAT	(1,2) + (3)
(2) A_GTCAT	(3) ACGTCCT	(3) ACGTCCT
(1) AC_TCAT	(3) ACGTCCT	(1) AC_TCAT 16
(2) A_GTCAT	(2) AG_TCAT	(2) AG_TCAT
(3) ACGTCCT		
		An alternate alignment
(1) AC_TCAT	(2) A_GTCAT	(1) AC_TCAT
(2) A_GTCAT	(3) ACGTCCT	(2) A_GTCAT 14
(3) ACGTCCT		(3) ACGTCCT

Note also...

The pairwise alignments induced by the *optimal multiple alignment* are *not* the same as the *optimal pairwise alignments*.

Optimal Pairwise Alignments		Optimal Multiple Alignment
(1) ACTCAT		(1) AC_TCAT
(2) AGTCAT	m	(2) A_GTCAT $2g$
		(3) ACGTCCT

Although this costs more, it may be a biologically more realistic alignment

Summary:
Progressive alignment heuristics

- are not guaranteed to give the optimal MSA
- bad choice of gaps propagates
- Complexity
 - Progressive: $O(k^2n^2)$
 - versus DP: $O(n^k \cdot 2^k \cdot k^2)$
- differ in
 - the order in which partial multiple alignments are merged.
 - how partial alignments are merged
- typically, merge the most closely related sequences first.