

Lecture 08

sequence alignment



Course: Practical Bioinformatics (BIOL 4220)
Instructor: Michael Landis
Email: michael.landis@wustl.edu



Lecture 08 outline

Last time: sequence data

This time: sequence alignment

- repository anatomy
- stage (add) and commit
- branch and merge
- local and remote

Sequence variation

Many questions in genome biology
are fundamentally ***comparative***

- where is this gene located in the genome?
- what amino acid differences cause two proteins to differ in function?
- how are two genes evolutionarily related?

Sequence variation

Any two sequences can differ
in length and/or content

TCCAAGCGTTATC

same length,
same content



TCCAAGCGTTATC

AATCAGTGGTATC



same length,
diff. content

diff. length,
diff. content



TAGTGGTATC

Sequence alignment

An ***alignment*** defines which parts of the sequence are evolutionary or functionally comparable (***homologous***)

(unaligned sequences)

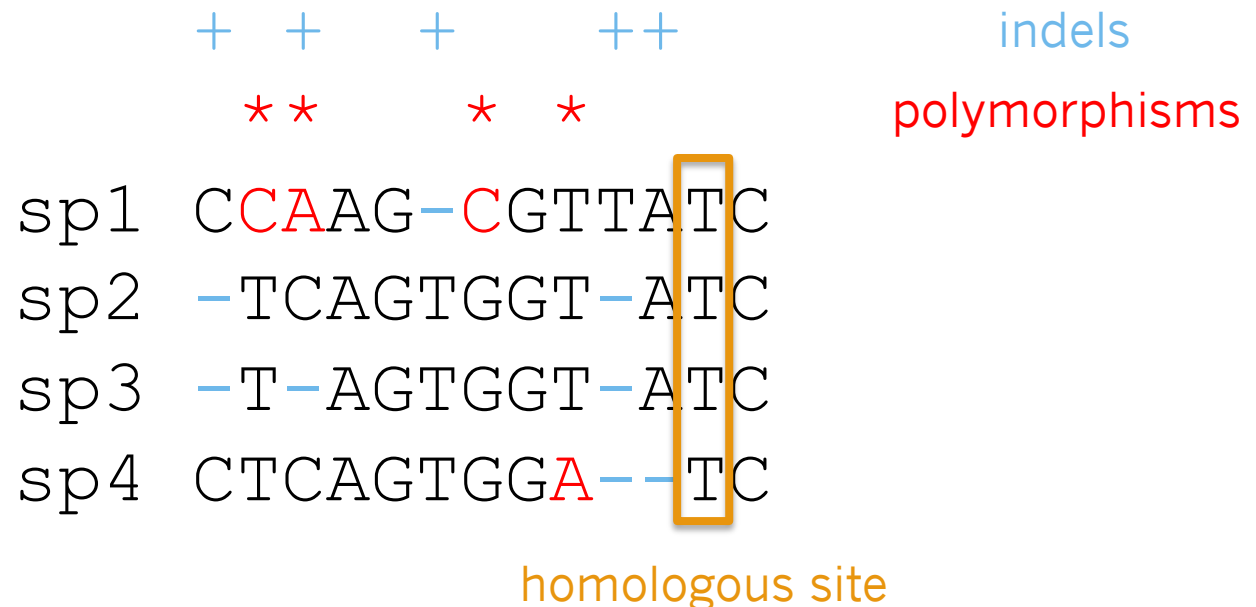
```
sp1  CCAAGCGTTATC
sp2  TCAGTGGTATC
sp3  TAGTGGTATC
sp4  CTCAGTGGATC
```

What creates sequence variation?



Sequence alignment

An ***alignment*** defines which parts of the sequence are evolutionary or functionally comparable (***homologous***)



substitutions are common
indels are rare

+ + + ++
* * * *

sp1 CCAAG-CGTTATC
sp2 -TCAGTGGT-ATC
sp3 -T-AGTGGT-ATC
sp4 CTCAGTGGAA--TC

5 indels
3 mismatches

substitutions are rare
indels are common

+++ + ++
*

sp1 C-CAAGCGGTTATC
sp2 -TCA-GTGGT-ATC
sp3 -T-A-GTGGT-ATC
sp4 CTCA-GTGG--ATC

6 indels
1 mismatches

Alignment methods

Alignment algorithms find the matrix for which:

- rows are different sequences/genes
- columns are homologous characters
- some optimization criterion is maximized

Two dominant method families:

- ***heuristic methods*** optimize “match scores” for alignment matrix
- ***generative methods*** reconstruct the most probable history that generated the alignment matrix

Heuristic alignment

Example: minimize cost of alignment

TCA**A**— — —GTAT**C**GACCT
TCA**T**GCGGTAT**T**—ACCT

- +1 Match (11)
- 1 Mismatch (2)
- 2 Gap open (2)
- 1 Gap extension (2)

$$+1 \times 11 + -1 \times 2 + -2 \times 2 + -1 \times 2 = +3$$

Heuristic alignment

Example: minimize cost of alignment

TCAA— — —GTAT—CGACCT
TCA—TGCGGTATT—ACCT

+1 Match (11)
-1 Mismatch (0)
-2 Gap open (4)
-1 Gap extension (4)

$$+1 \times 11 + -1 \times 0 + -2 \times 4 + -1 \times 4 = -1$$

Needleman-Wunsch example

	S1	T	G	A	C
S2	0				
T					
C					
A					
T					
C					

Type	Score
Match	+1
Mismatch	-1
Gap	-2

S1 : ?

S2 : ?

Needleman-Wunsch example

Recursive algorithm

- compute local cost to add a match, mismatch, or gap to the pairwise alignment for each cell
- construct path based on best cost
- continue until you reach the final site
- traverse path in reverse order to obtain alignment

Needleman-Wunsch example

	S1	T	G	A	C
S2	0	-2			
T	-2				
C					
A					
T					
C					

Type	Score
Match	+1
Mismatch	-1
Gap	-2

S1 : ?

S2 : ?

Needleman-Wunsch example

	S1	T	G	A	C
S2	0	-2	-4	-6	-8
T	-2				
C	-4				
A	-6				
T	-8				
C	-10				

Type	Score
Match	+1
Mismatch	-1
Gap	-2

S1 : ?

S2 : ?

Needleman-Wunsch example

match?
 $0 + 1$
 S1 T
 S2 T

	S1	T	G	A	C
S2	0	-2	-4	-6	-8
T	-2	???			
C	-4				
A	-6				
T	-8				
C	-10				

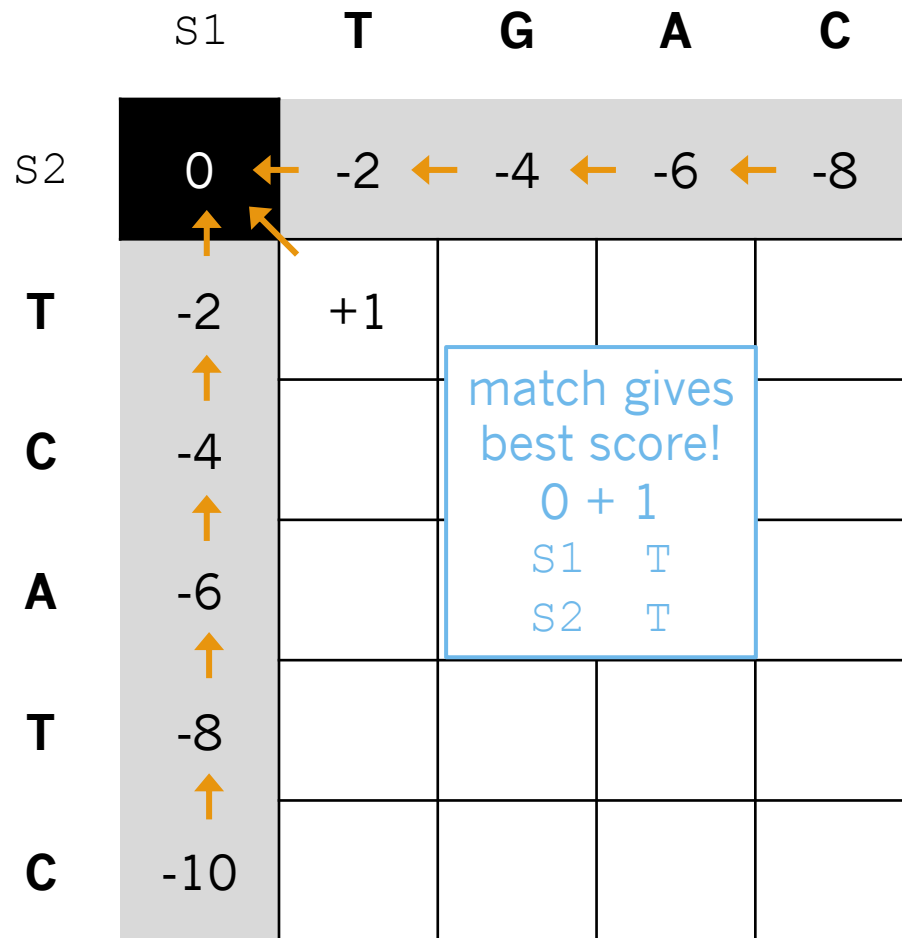
gap?
 $-2 + -2$
 S1 -
 S2 T

gap?
 $-2 + -2$
 S1 T
 S2 -

Type	Score
Match	+1
Mismatch	-1
Gap	-2

S1 : ?
 S2 : ?

Needleman-Wunsch example



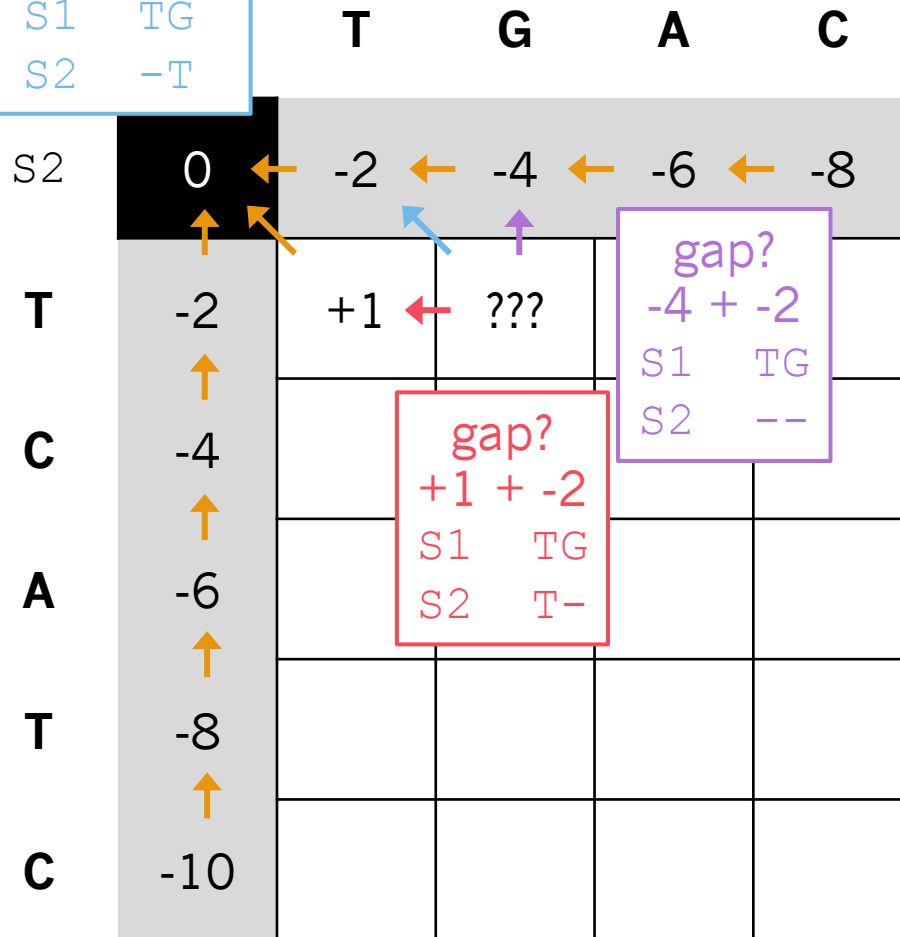
Type	Score
Match	+1
Mismatch	-1
Gap	-2

S1 : ?

S2 : ?

Needleman-Wunsch example

mismatch?
 $-2 + -1$
 S1 TG
 S2 -T



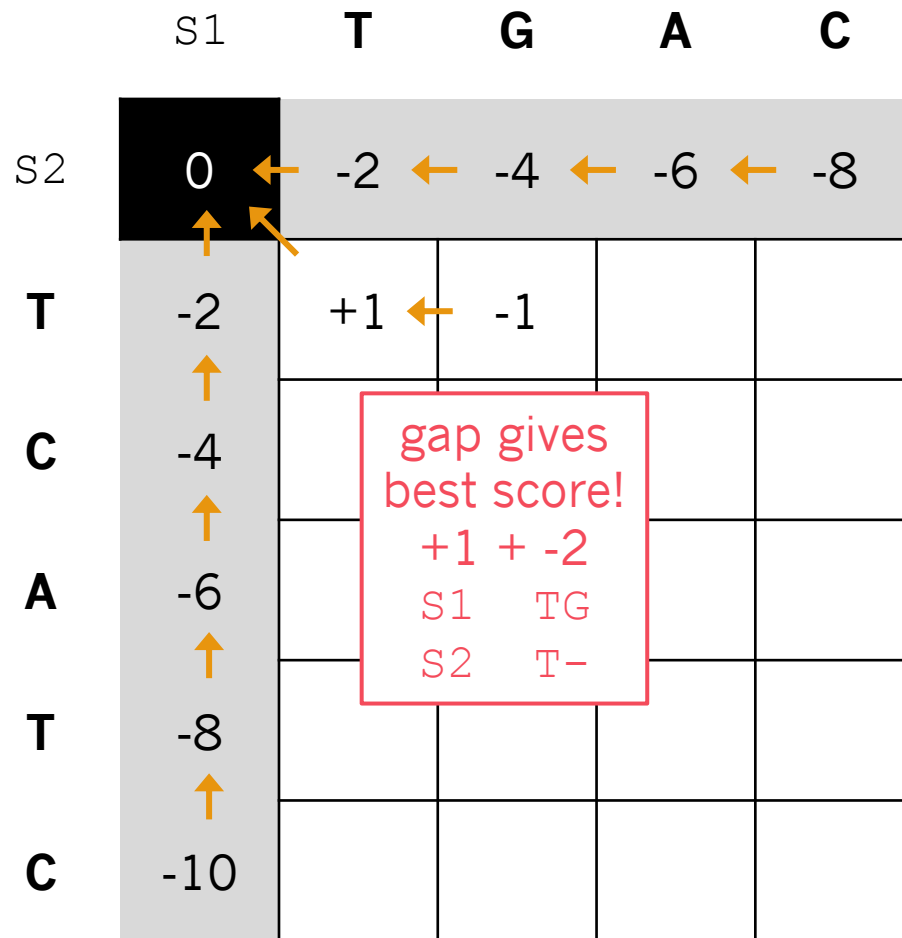
gap?
 $-4 + -2$
 S1 TG
 S2 --

gap?
 $+1 + -2$
 S1 TG
 S2 T-

Type	Score
Match	+1
Mismatch	-1
Gap	-2

S1 : ?
 S2 : ?

Needleman-Wunsch example



Type	Score
Match	+1
Mismatch	-1
Gap	-2

S1 : ?

S2 : ?

Needleman-Wunsch example

	S1	T	G	A	C
S2	0	-2	-4	-6	-8
T	-2	+1	-1	-3	-5
C	-4				
A	-6				
T	-8				
C	-10				

Orange arrows indicate the alignment path from the top-left cell (0,0) to the cell (1,1) containing +1. The path starts at (0,0), moves down to (1,0) containing -2, and then moves right to (1,1) containing +1.

Type	Score
Match	+1
Mismatch	-1
Gap	-2

S1 : ?

S2 : ?

Needleman-Wunsch example

	S1	T	G	A	C
S2	0	-2	-4	-6	-8
T	-2	+1	-1	-3	-5
C	-4	-1	0	-2	-2
A	-6				
T	-8				
C	-10				

Type	Score
Match	+1
Mismatch	-1
Gap	-2

S1 : ?

S2 : ?

Needleman-Wunsch example

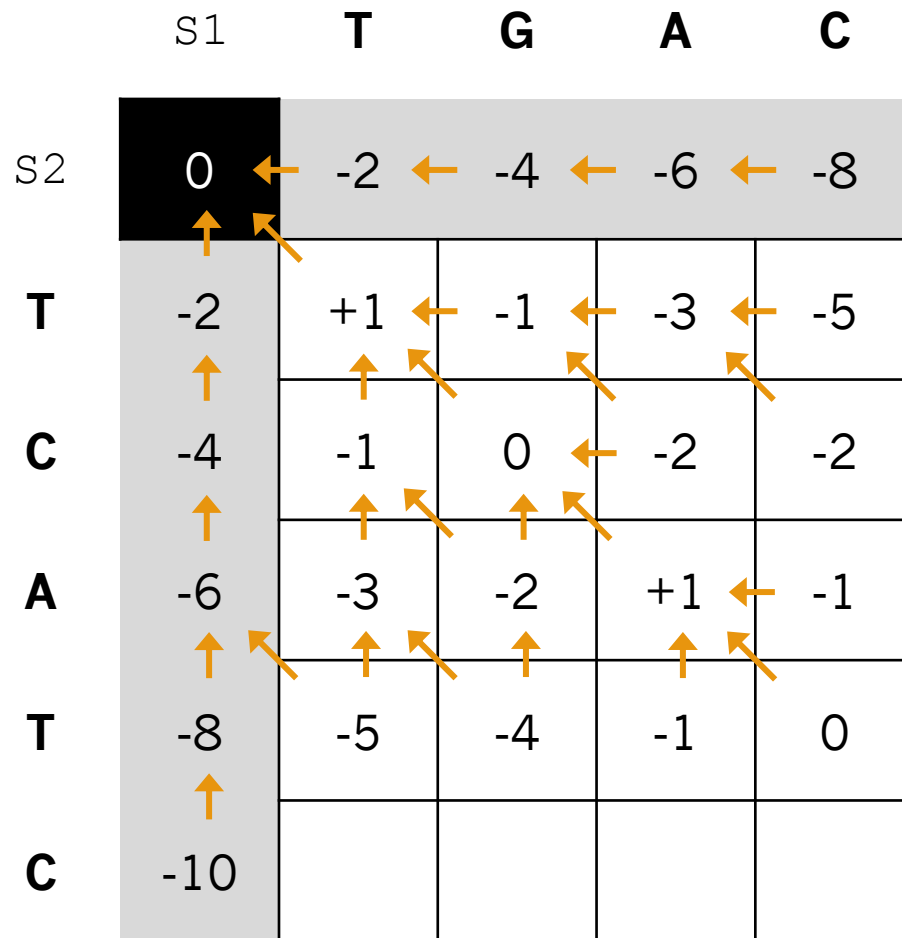
	S1	T	G	A	C
S2	0	-2	-4	-6	-8
T	-2	+1	-1	-3	-5
C	-4	-1	0	-2	-2
A	-6	-3	-2	+1	-1
T	-8				
C	-10				

Type	Score
Match	+1
Mismatch	-1
Gap	-2

S1 : ?

S2 : ?

Needleman-Wunsch example



Type	Score
Match	+1
Mismatch	-1
Gap	-2

S1 : ?

S2 : ?

Needleman-Wunsch example

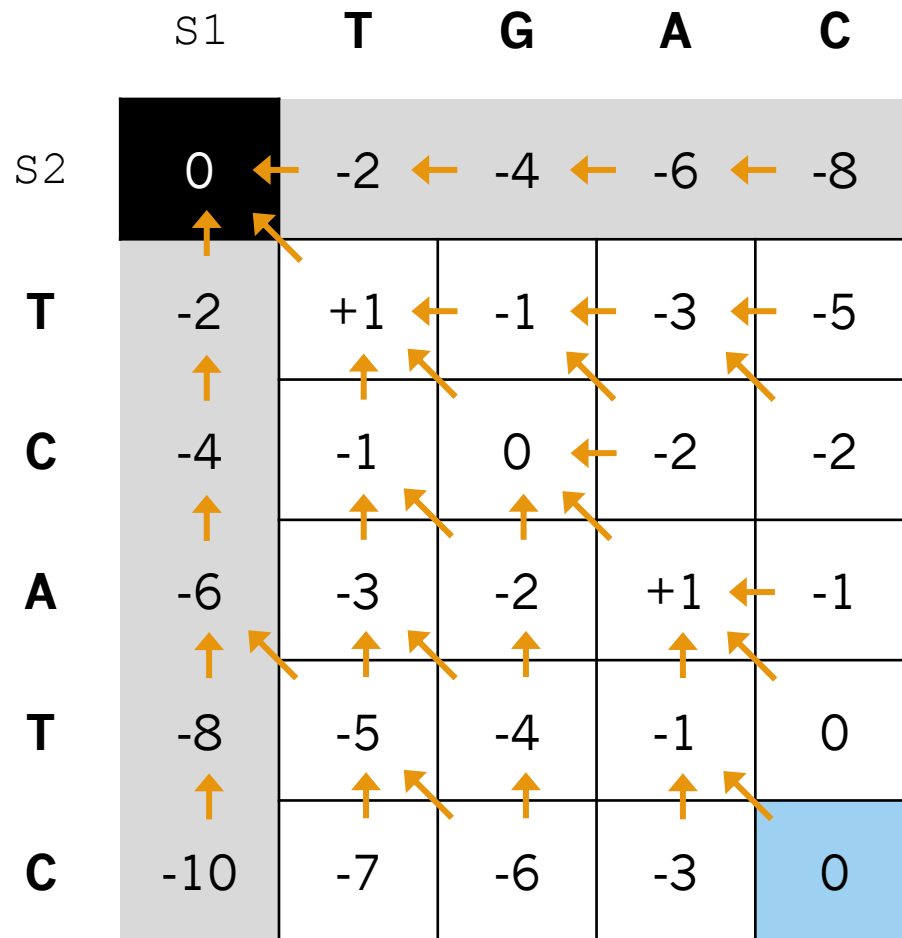
	S1	T	G	A	C
S2	0	-2	-4	-6	-8
T	-2	+1	-1	-3	-5
C	-4	-1	0	-2	-2
A	-6	-3	-2	+1	-1
T	-8	-5	-4	-1	0
C	-10	-7	-6	-3	0

Type	Score
Match	+1
Mismatch	-1
Gap	-2

S1 : ?

S2 : ?

Needleman-Wunsch example

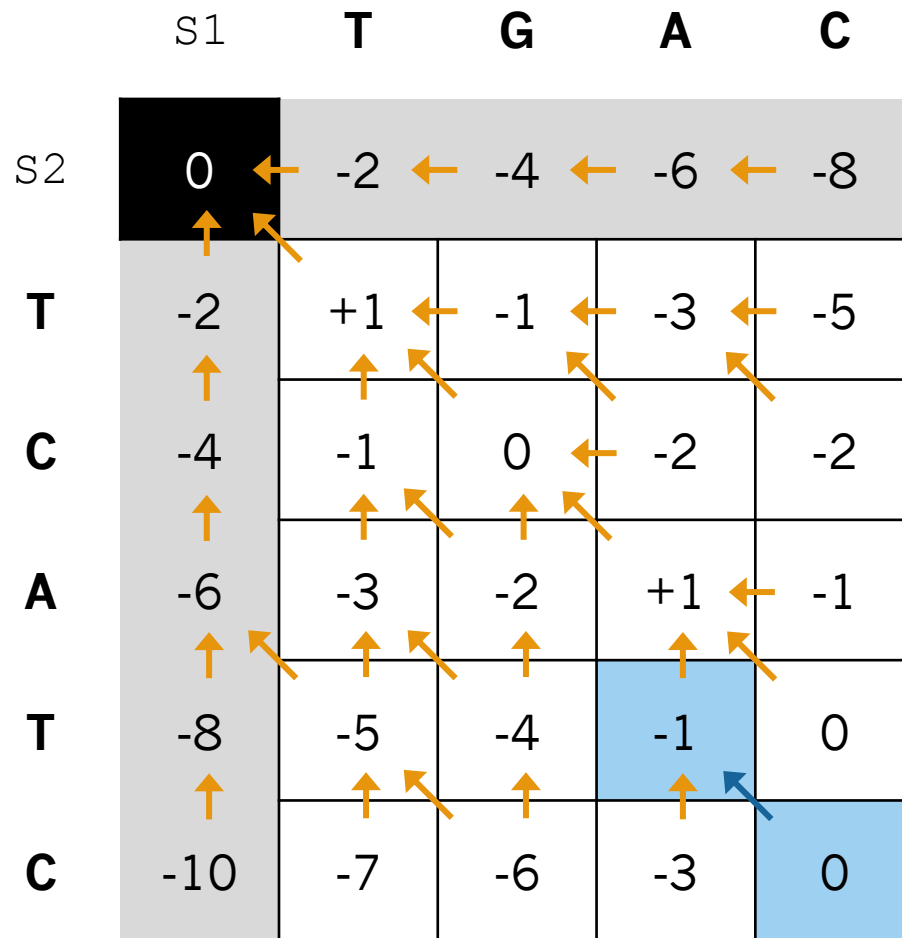


Type	Score
Match	+1
Mismatch	-1
Gap	-2

S1 : C

S2 : C

Needleman-Wunsch example

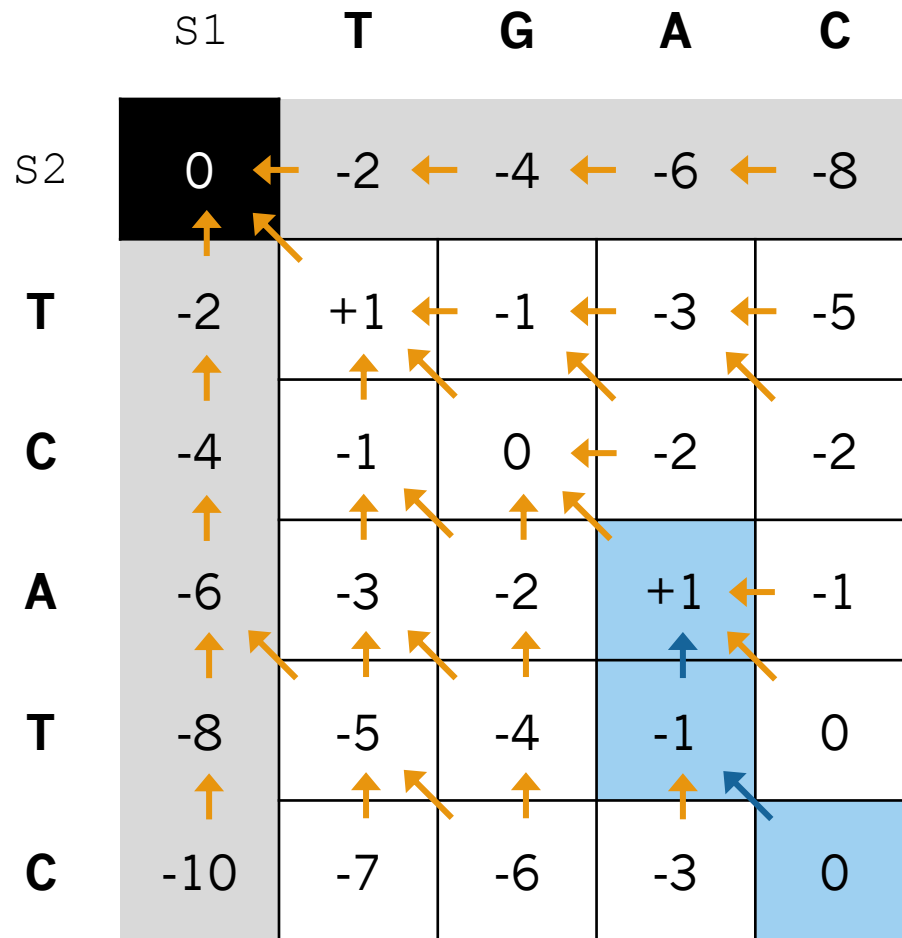


Type	Score
Match	+1
Mismatch	-1
Gap	-2

S1 : AC

S2 : TC

Needleman-Wunsch example

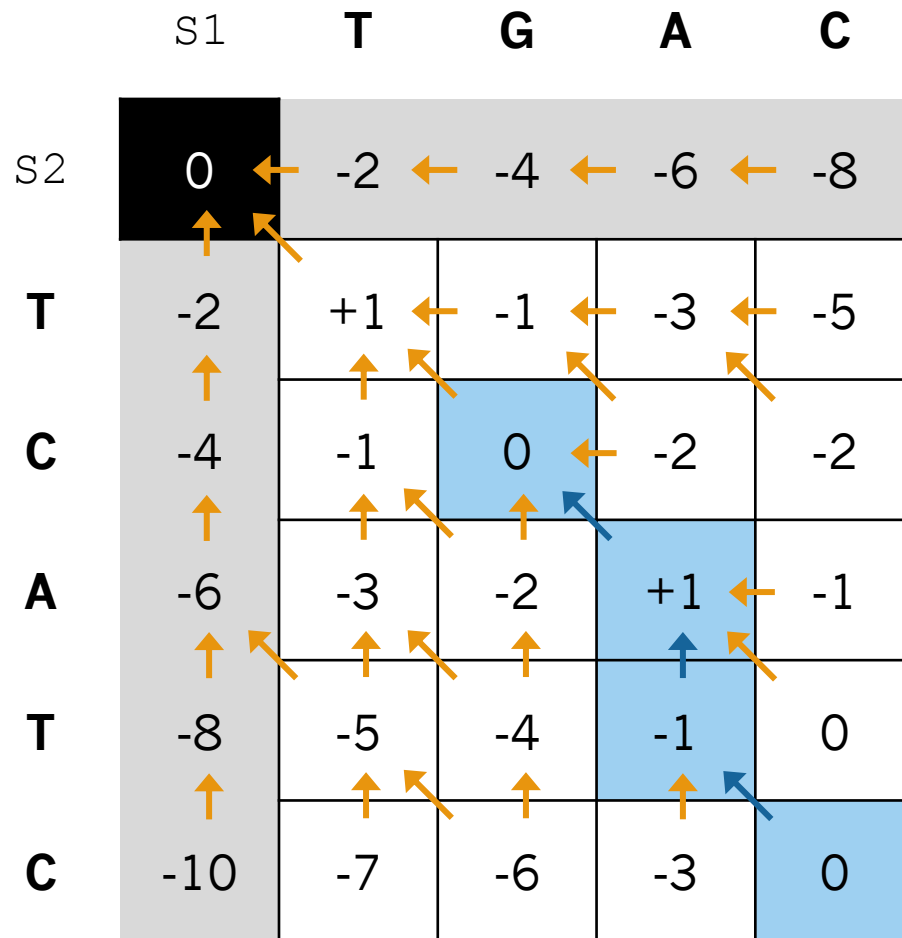


Type	Score
Match	+1
Mismatch	-1
Gap	-2

S1 : A-C

S2 : ATC

Needleman-Wunsch example

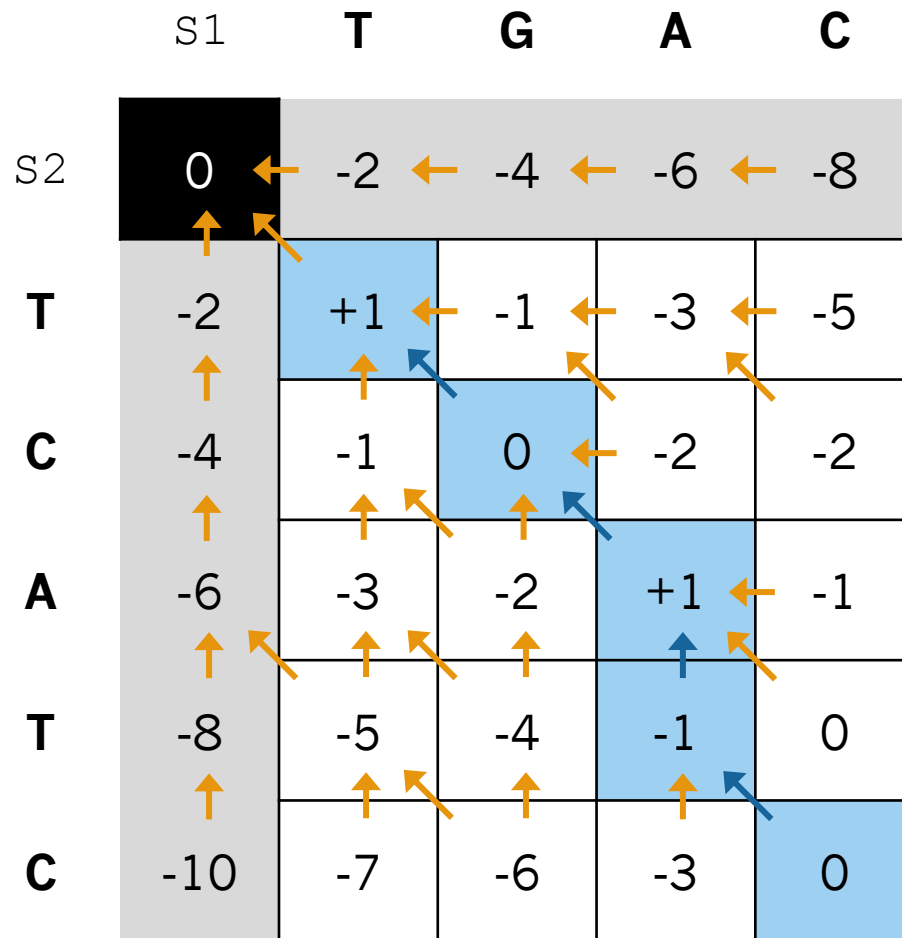


Type	Score
Match	+1
Mismatch	-1
Gap	-2

S1 : GA-C

S2 : CATC

Needleman-Wunsch example

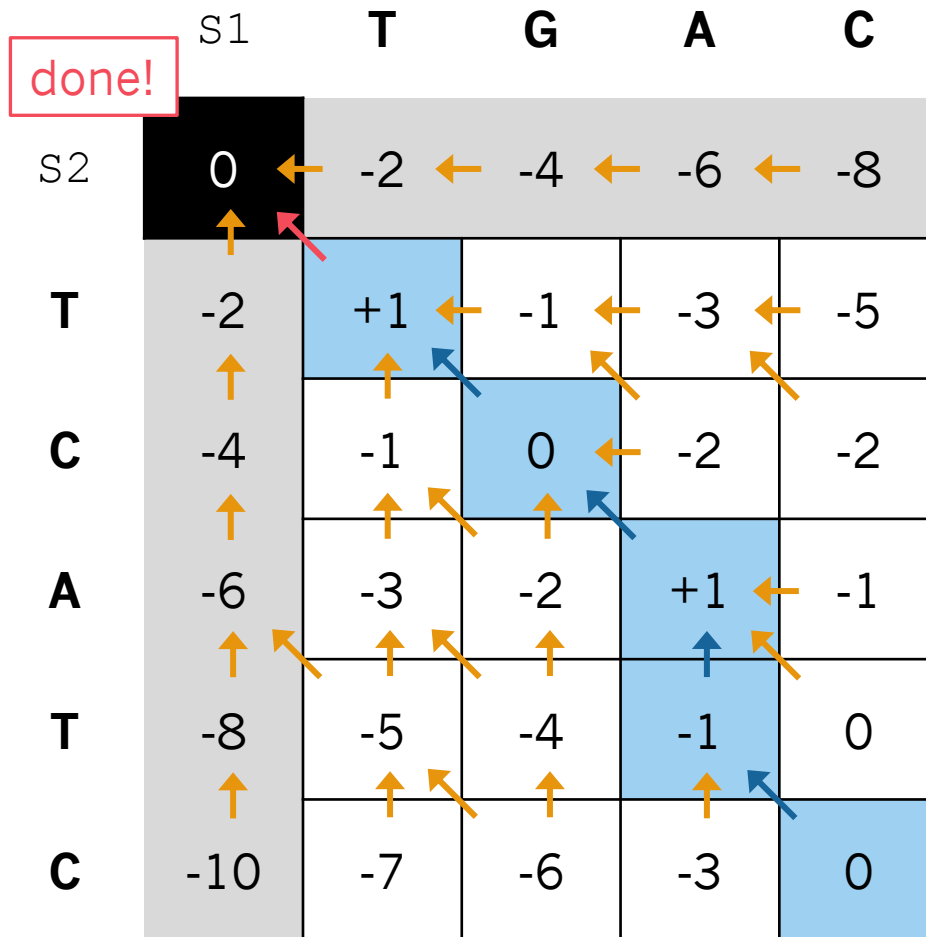


Type	Score
Match	+1
Mismatch	-1
Gap	-2

S1 : TGA-C

S2 : TCATC

Needleman-Wunsch example

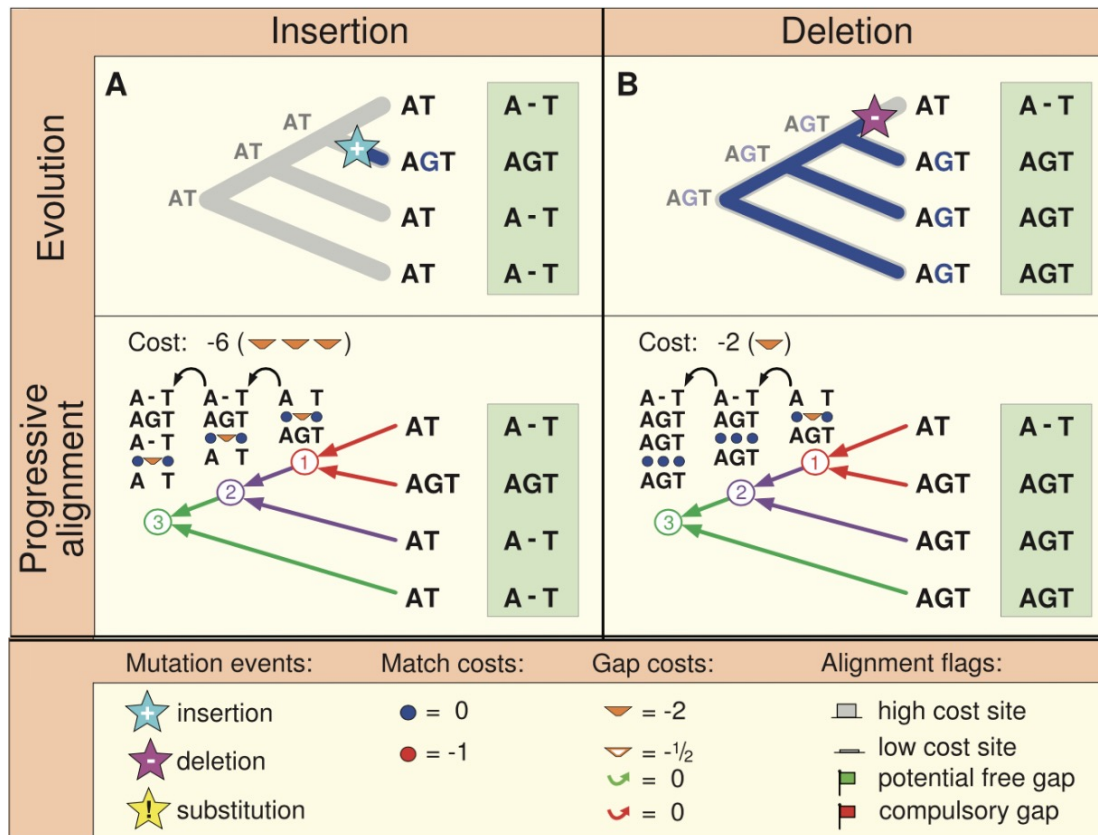


Type	Score
Match	+1
Mismatch	-1
Gap	-2

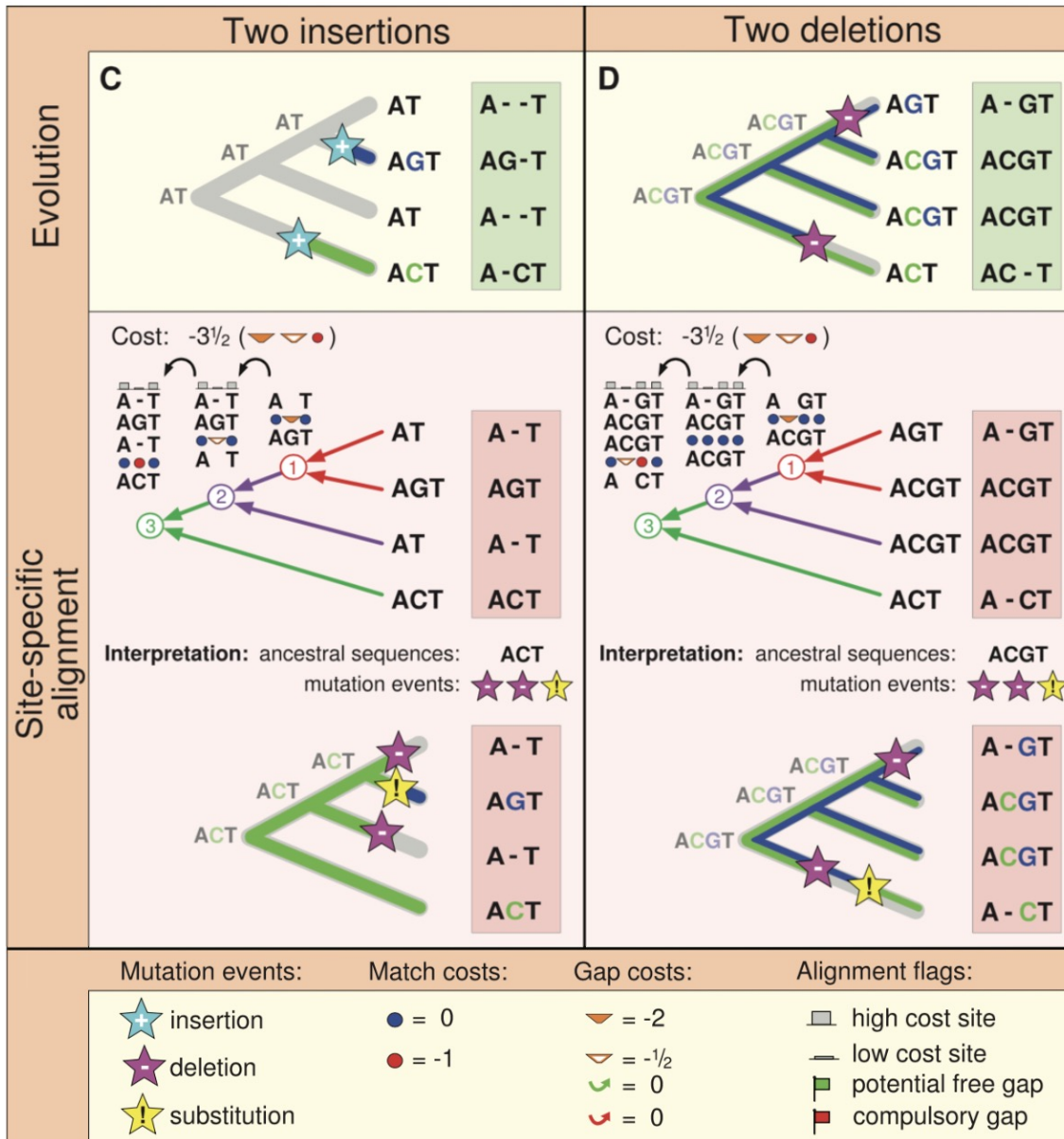
S1 : TGA-C

S2 : TCATC

Progressive alignment



Aligns multiple sequences by *progressively* adding new sequences to alignment based on a **guide tree** (*phylogeny*)



Even mildly complex evolutionary scenarios can cause progressive alignments to produce inaccurate homology statements

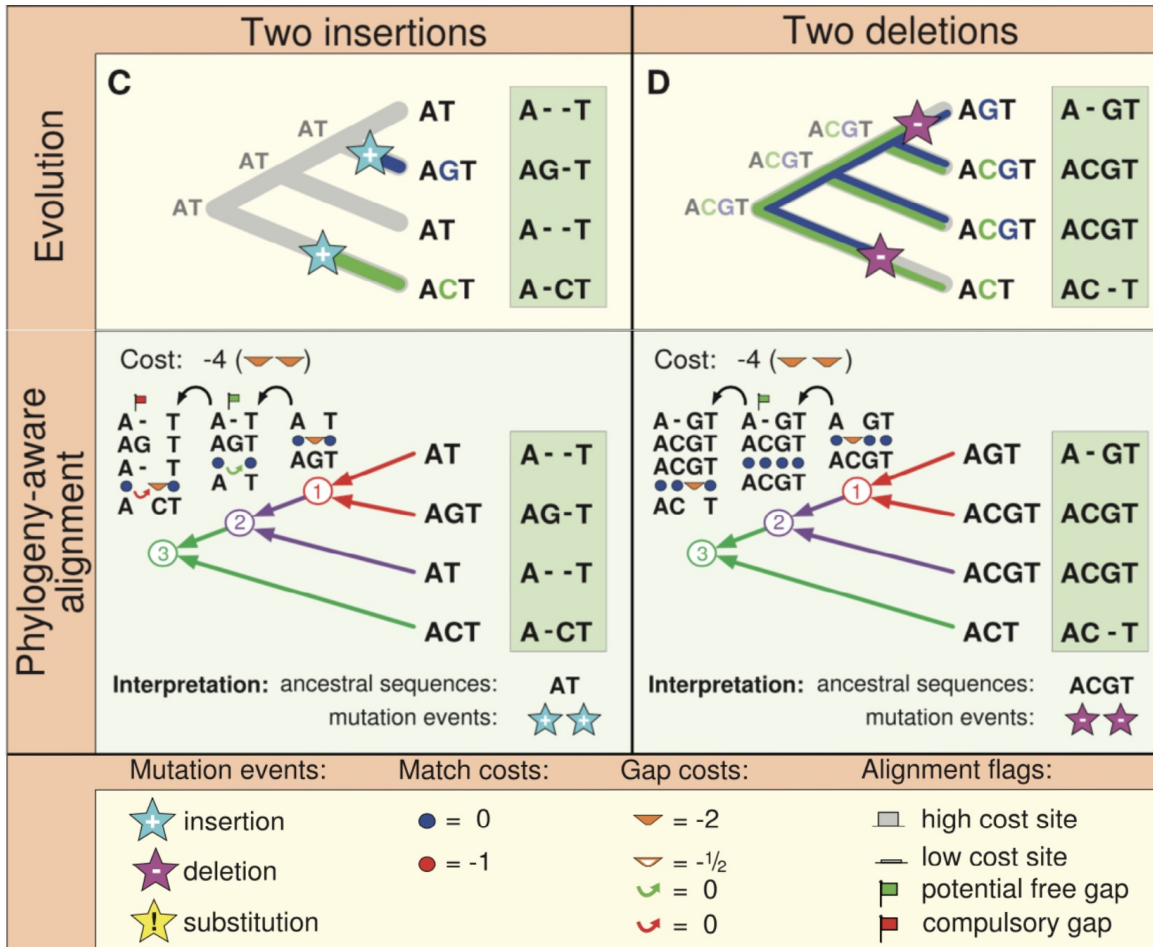
See the two-insertion scenario (left)

Generative alignment

Generative methods uses guide tree to reconstruct the series of substitution, insertion, and deletion events that most likely generated a sequence alignment

Alignment score depends on rates of:

- substitution
- deletion
- insertion



This software (PRANK; left) “flags” events as scored so they’re not double-counted

Generative alignments tend to be “gappier”

More evolutionarily accurate statements of homology

list exact matches, *grep -o*

lists all
instances
that match
“ore”

```
$ cat limerick.txt
A UNIX sales lady, Lenore,
Enjoys work, but she likes the beach more.
She found a good way
To combine work and play:
She sells C shells by the seashore.
$ grep -o ore limerick.txt
ore
ore
ore
```

Overview for Lab 08