From pairwise to multiple alignment

- Alignment of 2 sequences is represented as a 2-row matrix
- In a similar way, we represent alignment of 3 sequences as a 3-row matrix

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

- Score: more conserved columns, better alignment

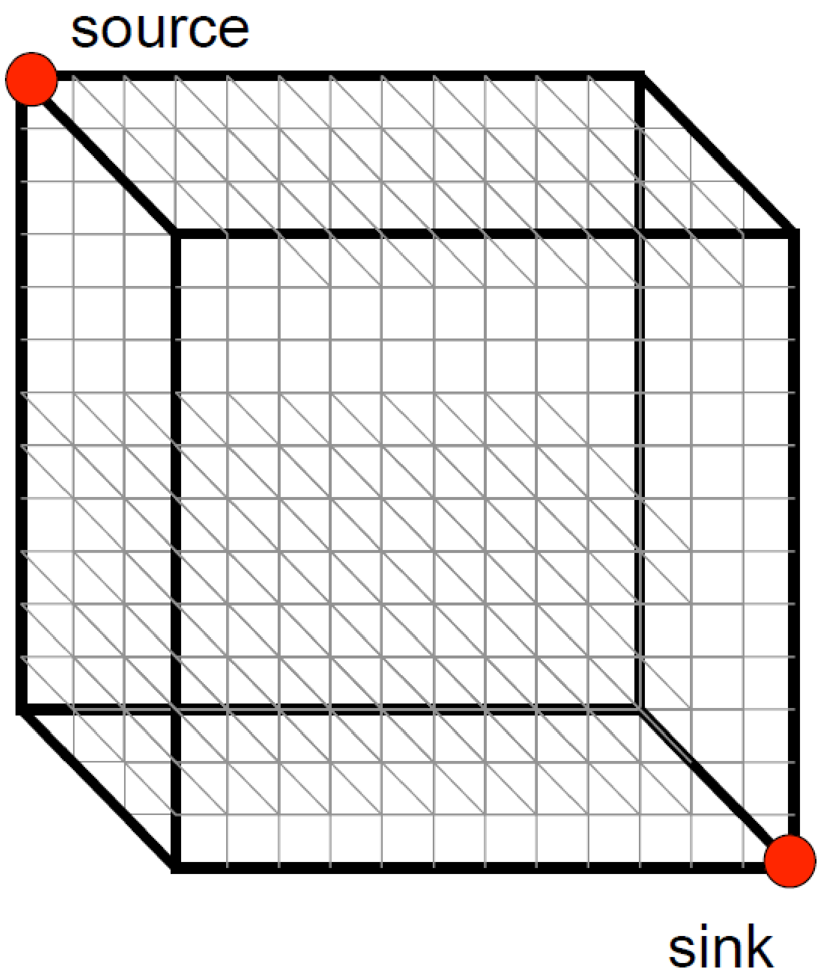
What is a multiple sequence alignment (MSA)

- A model
- Indicates relationship between residues of multiple sequences
- Reveals similarity/dissimilarity

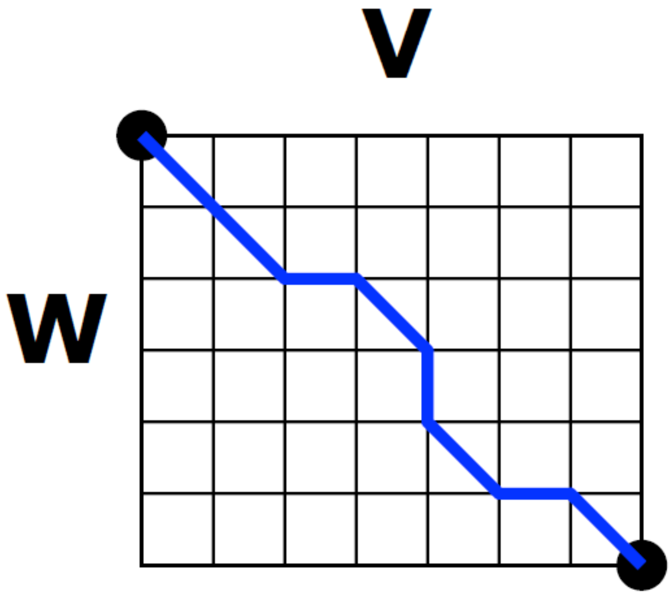
- Central to many bioinformatics applications
  - Patterns (Motifs)
  - Structure prediction (RNA, protein)

Aligning three sequences

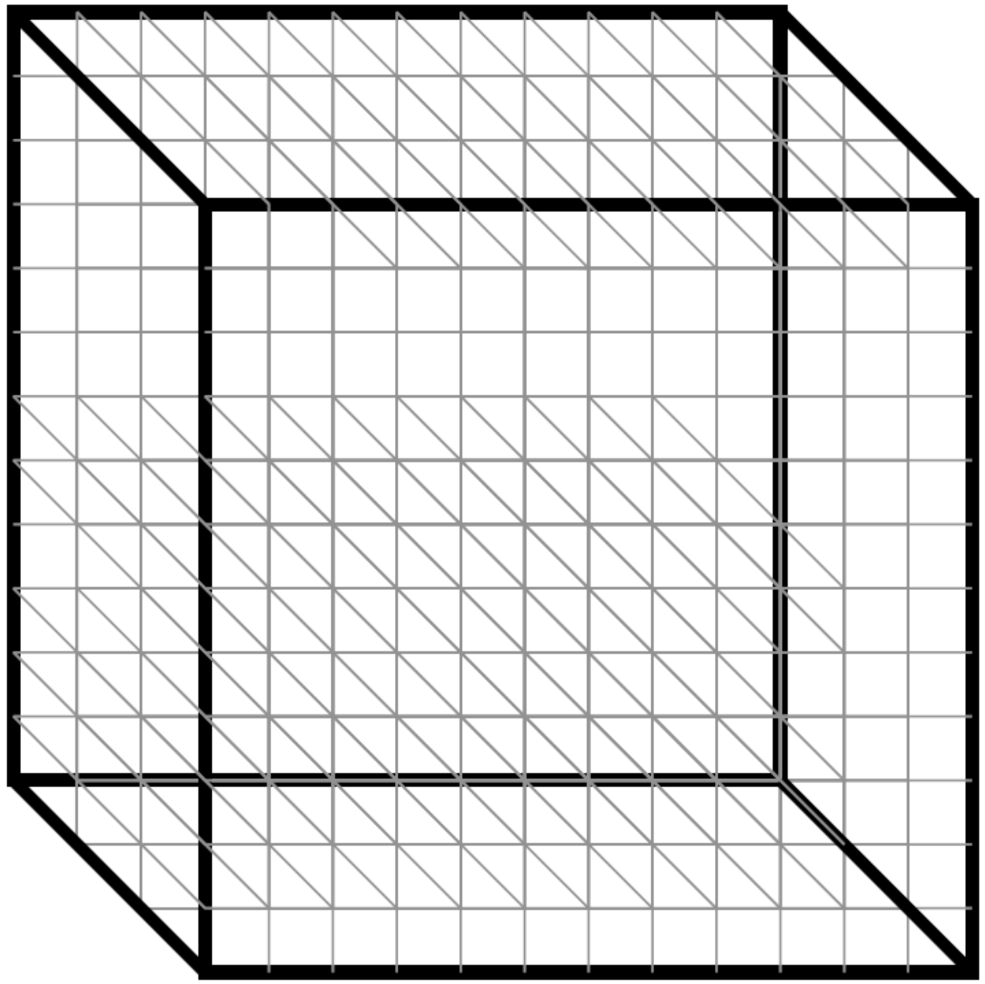
- Same strategy as aligning two sequences
- Use a 3-D "Manhattan Cube", with each axis representing a sequence to align
- For global alignments, go from source to sink



2D vs 3D alignment grid

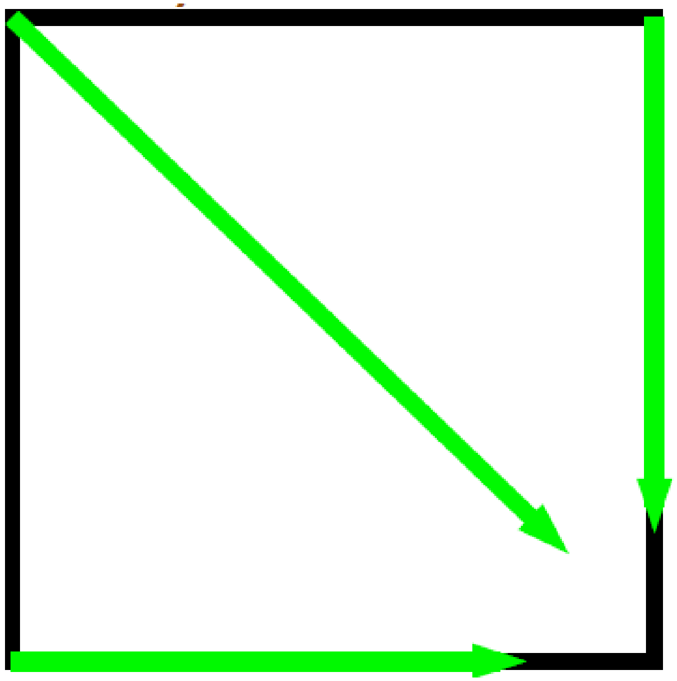


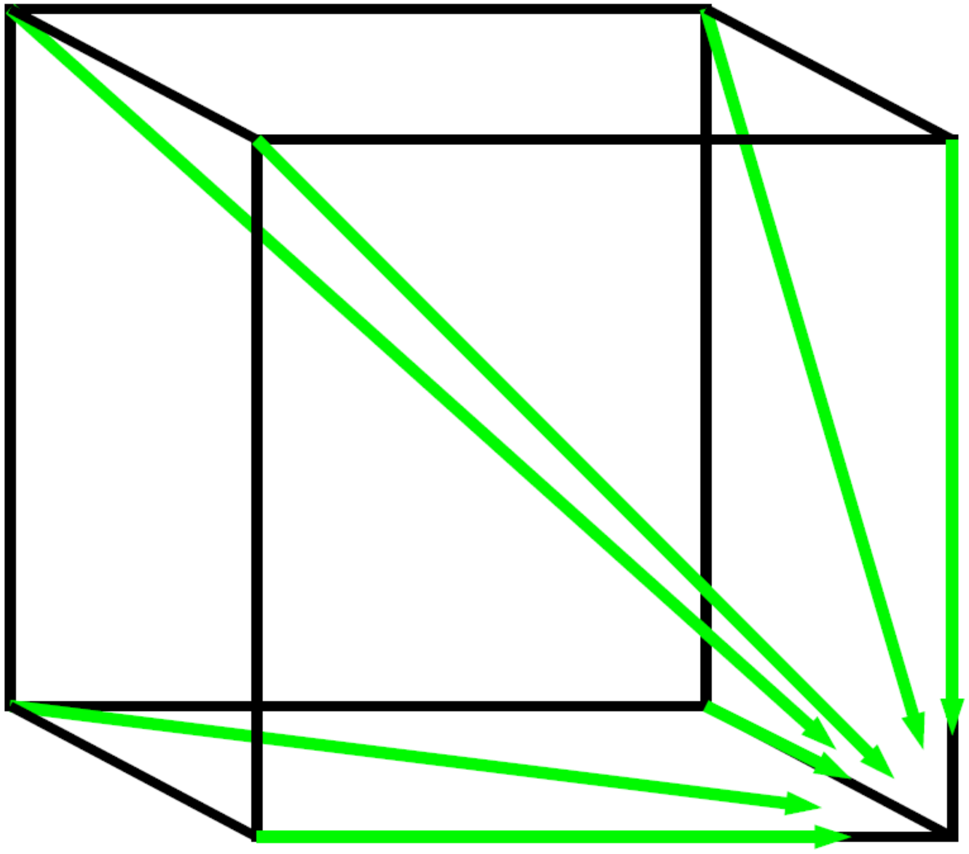
2D grid



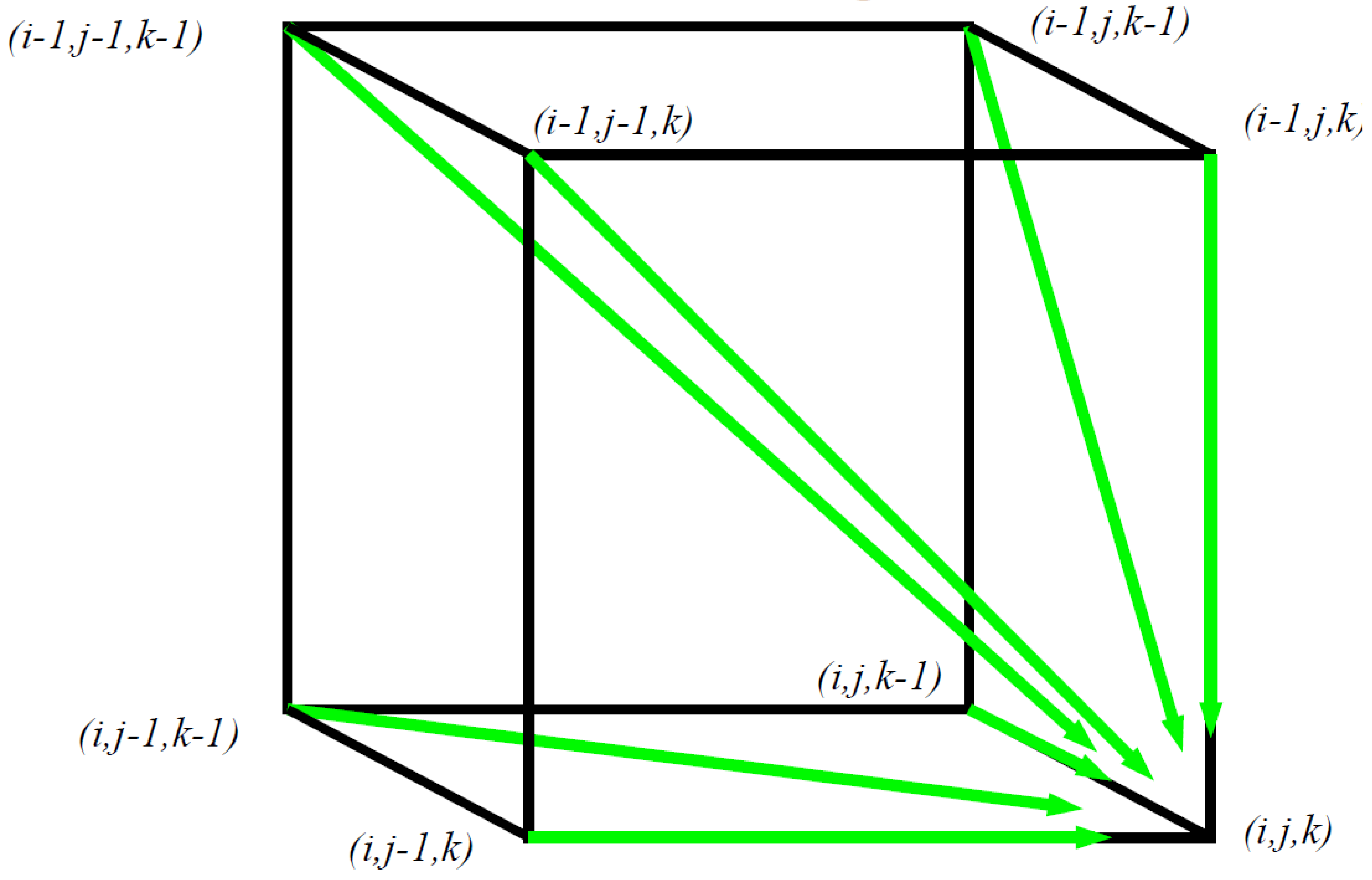
3D grid

DP recursion (3 edges vs 7)


Pairwise: 3 possible paths (match/mismatch, insertion, and deletion)


In **3-D**, 7 edges in each unit cube

Architecture of 3D alignment cell



Multiple alignment: dynamic programming

$$s_{i,j,k} = \max \begin{cases} s_{i-1,j-1,k-1} + \delta(v_i, w_j, u_k) \\ s_{i-1,j-1,k} + \delta(v_i, w_j, \_) \\ s_{i-1,j,k-1} + \delta(v_i, \_, u_k) \\ s_{i,j-1,k-1} + \delta(\_, w_j, u_k) \\ s_{i-1,j,k} + \delta(v_i, \_, \_) \\ s_{i,j-1,k} + \delta(\_, w_j, \_) \\ s_{i,j,k-1} + \delta(\_, \_, u_k) \end{cases}$$

cube diagonal: no indels

face diagonal: one indel

edge diagonal: two indels

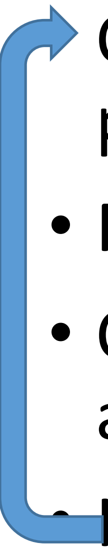
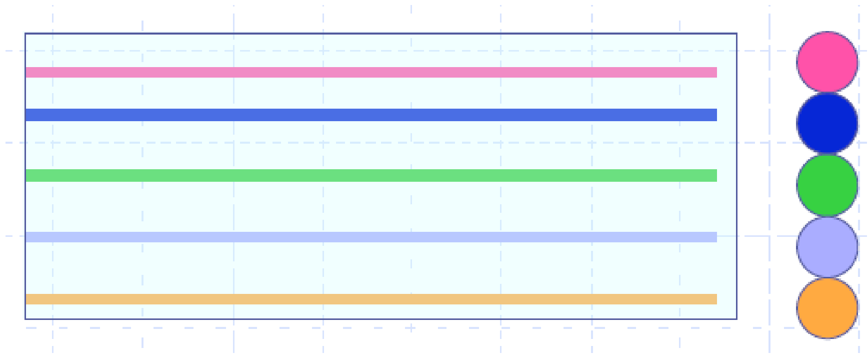
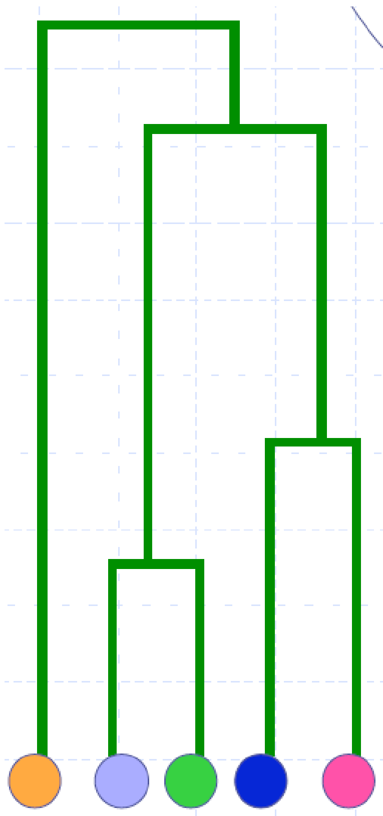
$\delta(x, y, z)$ is an entry in the 3D scoring matrix

MSA: runtime

- For 3 sequences of length ***n***, the run time is $7n^3$; O($\mathbf{n}^3$)
- For ***k*** sequences, build a ***k***-dimensional Manhattan, with run time $(2^k-1)(n^k)$; $O(2^k n^k)$
- Conclusion: dynamic programming approach for alignment between two sequences is easily extended to ***k*** sequences (simultaneous approach) but it is impractical due to exponential running time
- Computing exact MSA is computationally almost impossible, and in practice heuristics are used (progressive alignment)

Basic progressive alignment strategy

- Compute *D*, a matrix of distances between all pairs of sequences
- From *D*, construct a "guide tree" T
- Construct MSA by pairwise alignment of partial alignments ("profiles") guided by T
- Improve alignment by iterations, etc.

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

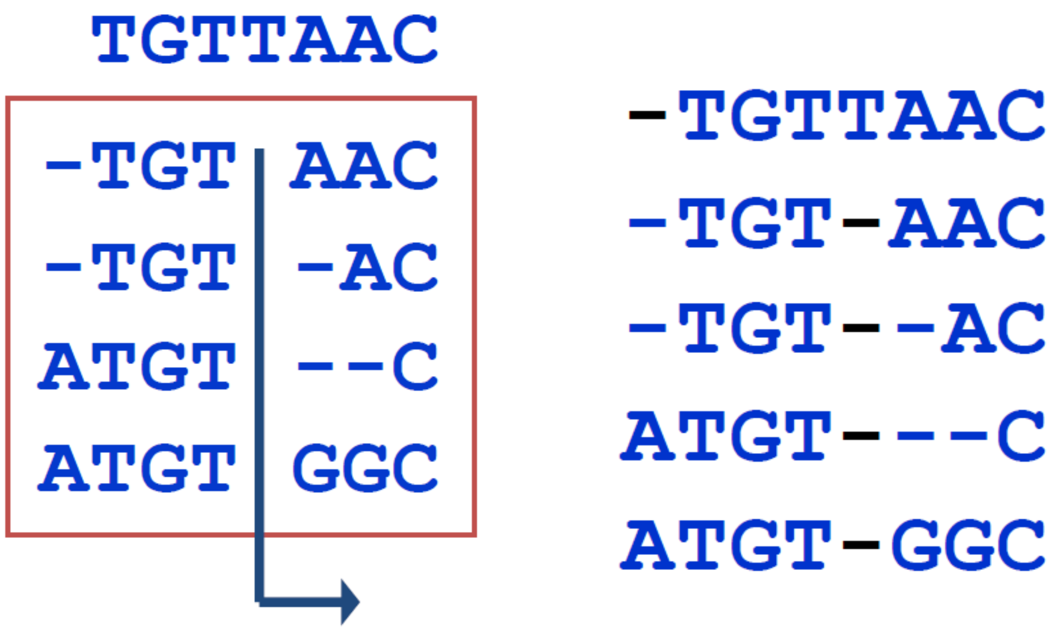
Picking the center in star alignments

- Two possible approaches:

1. try each sequence as the center, return the best multiple alignment
2. compute all pairwise alignments and select the string $\sum_{i \neq c} \mathrm{sim}(x^i, x^c)$

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



```
   TGTTAAC
                    -TGTTAAC
 -TGT AAC           -TGT-AAC
 -TGT -AC           -TGT--AC
 ATGT --C           ATGT---C
 ATGT GGC           ATGT-GGC
```

Star Alignment Example

Given:

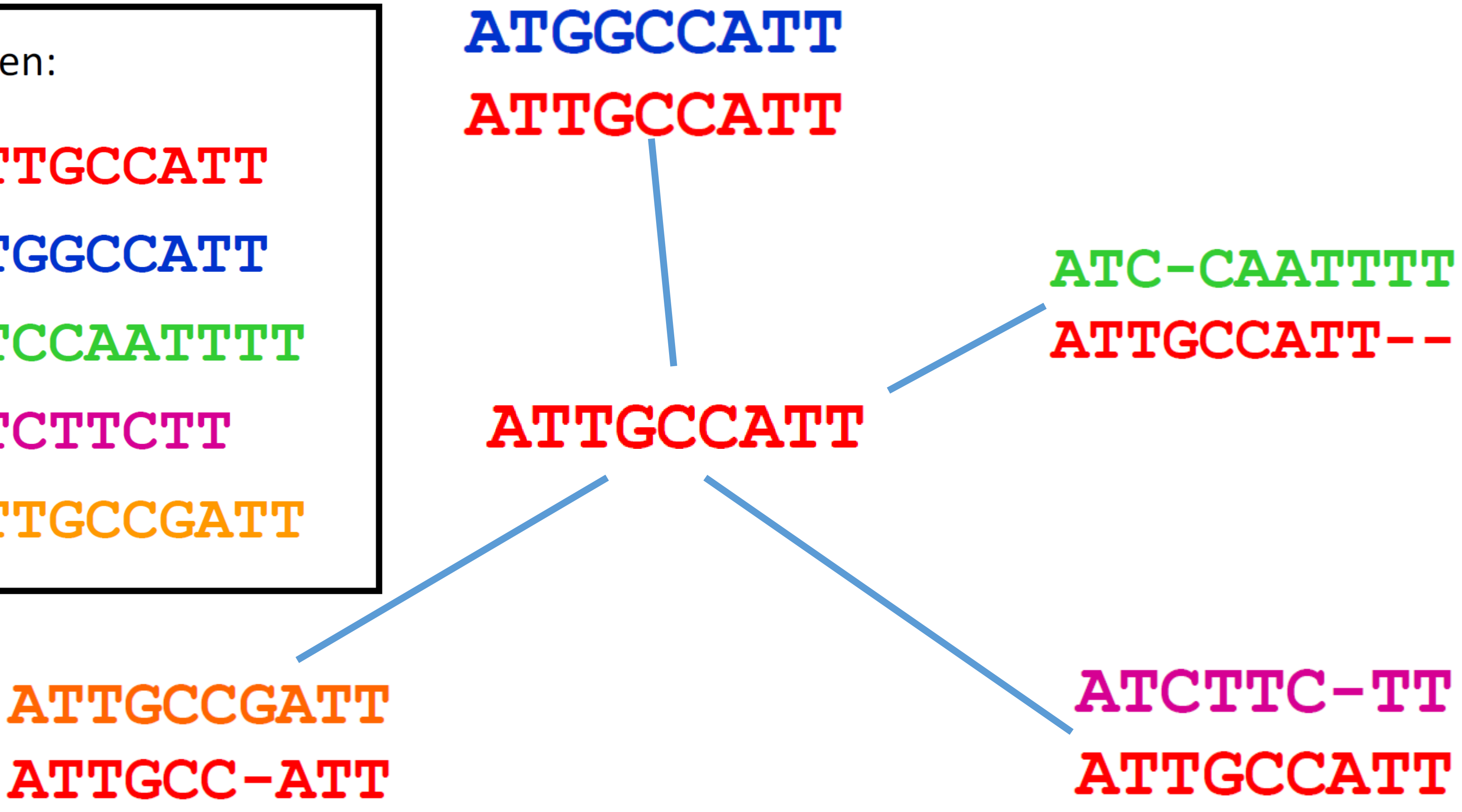
ATTGCCATT

ATGGCCATT

ATCCAATTTT

ATCTTCTT

ATTGCCGATT



Star Alignment Example

- Aggregate pairwise alignments

present pair alignment

1.
```
ATGGCCATT
ATTGCCATT
```

2.
```
ATC-CAATTTT
ATTGCCATT--
```

3.
```
ATCTTC-TT
ATTGCCATT
```

Current multiple

```
ATTGCCATT
ATGGCCATT
```

```
ATTGCCATT--
ATGGCCATT--
ATC-CAATTTT
```

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

Star Alignment Example

present pair

4.
```
ATTGCCGATT
ATTGCC-ATT
```

Current multiple alignment

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

shift entire columns

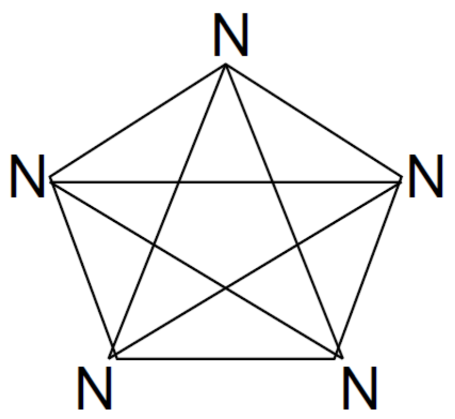
when incorporating a gap

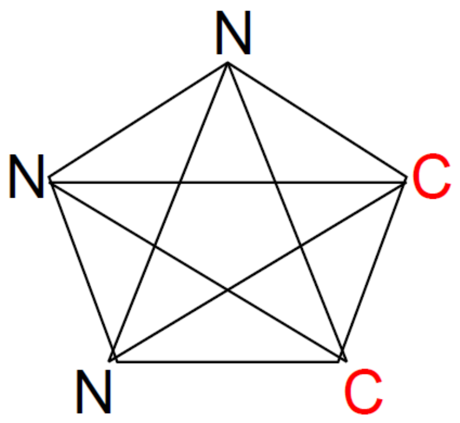
Comments about Star alignment

- 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

Sum of pairs score (SP score)

```
Seq   Column-A -B
1        ......N..............N......
2        ......N..............N......
3        ......N..............N......
4        ......N..............C......
5        ......N..............C......
```





S(N,N) = 6
S(N,C) = -3
S(C,C) = 9

Score= 10 * S(N,N)
= 10 * 6 = 60

Score= 3 * S(N,N) + 6 * S(N,C) + S(C,C)
= 3 * 6 + 6 * (-3) + 9 = 9

Complexity of progressive alignment

- Distance matrix
  - Each pairwise alignment O(n^2)
  - Number of pairwise alignments O(k^2)
- Iterative construction of MSA
  - Number of merge steps O(k)
  - Each pairwise alignment O(k^2n^2)

- Entire method O(k^2n^2)

Summary: Progressive alignment heuristics

- Not guaranteed to give the optimal MSA
- Bad choice of gaps propagates (because we never remove a gap)
- Complexity
  - Progressive: O($k^2 n^2$)
  - DP: O($n^k\ 2^k\ k^2$)
- Typically, merge the most closely related sequences first.

- https://en.wikipedia.org/wiki/List_of_sequence_alignment_software

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	