Comparison of sequences

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY **DELHI**

Dr. Jaspreet Kaur Dhanjal
Assistant Professor, Center for Computational Biology
Email ID: jaspreet@iiitd.ac.in

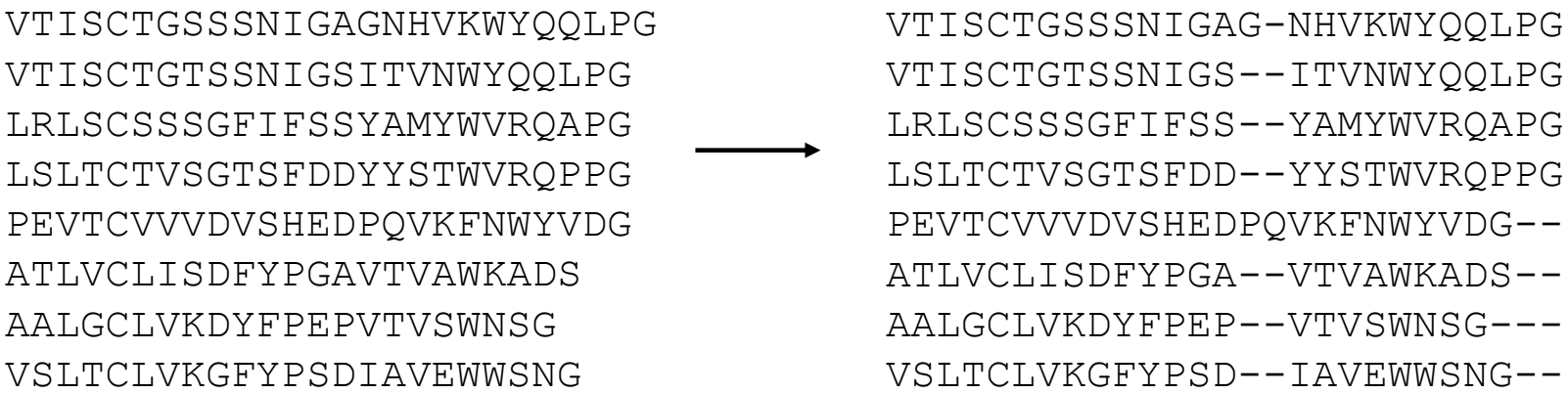
August 26, 2025

Multiple sequence alignment

Multiple sequence alignment (MSA)

- Aligning more than 2 sequences together to find conserved residues or regions
- Input for phylogenetic reconstruction

- Given k sequences, find an alignment that maximizes the alignment score



Scoring multiple sequence alignment

```
a: VSLSCTGSSSNIGAG-NHVKWYQQLPG
b: VTISCTGTSSNIG--SITVNWYQQLPG
c: ATLVCLISDFYPGA-SVTVAWKADS--
d: AALGCLVKDYFPEP--VTVSWNSG---
e: --LTCLVKGFYPSD--IAVEWESNG--
```

Sum of pairs (SP) score

V
V
A
A
–

Score of column = sum of all pairs

$= S_{VV}+S_{VA}+S_{VA}+S_{V-}$

$+ S_{VA}+S_{VA}+S_{V-}$

$+ S_{AA}+S_{A-}$

$+ S_{A-}$

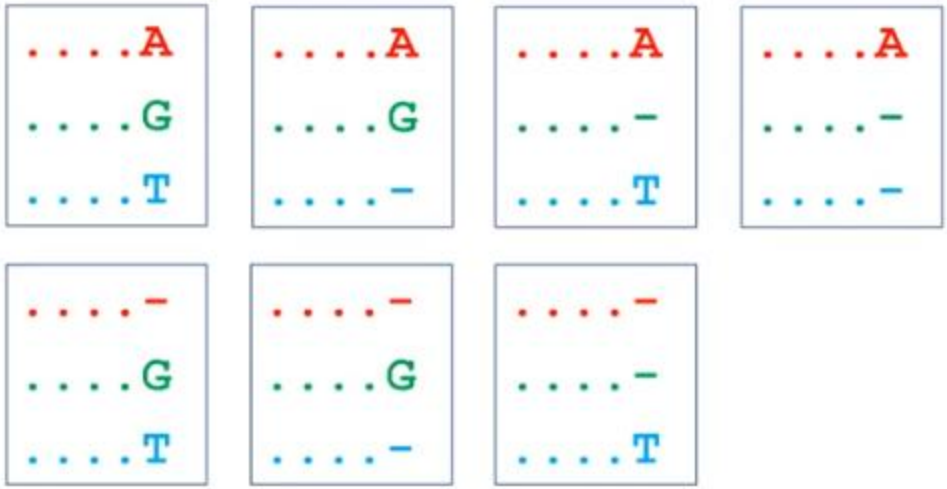
Multiple alignment score = sum of all column scores

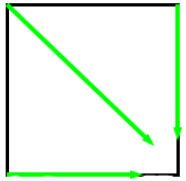
Pairwise sequence alignment score: $S_{ab} = S_{VV}+S_{ST}+S_{LI}+S_{SS}+\ldots\ldots\ldots\ldots$

Multiple alignment score = sum of all pairwise alignments = $S_{ab}+S_{ac}+S_{ad}+S_{ae}+S_{bc}+S_{bd}+S_{be}+S_{cd}+S_{ce}+S_{de}$

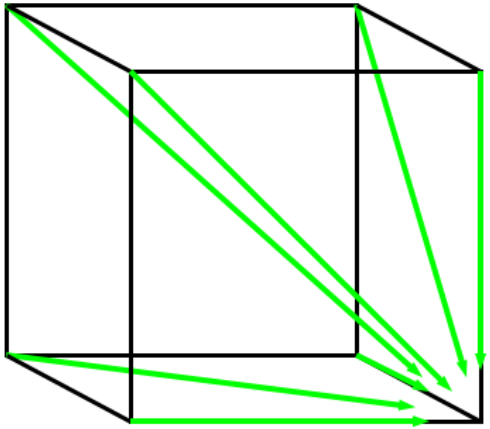
Dynamic programming for MSA

Aligning three sequences:



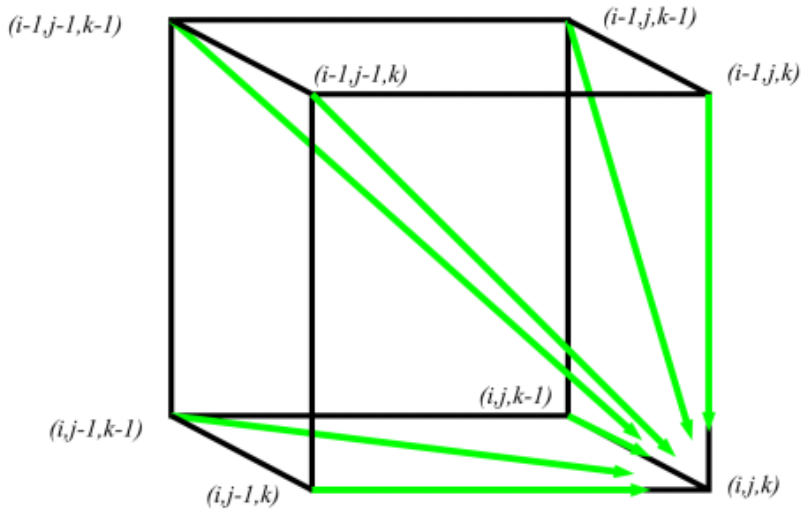


Pairwise: 3 possible paths (match/mismatch, insertion, and deletion)



In 3-D,
7 edges in each unit cube

Dynamic programming for MSA



$$s_{i,j,k} = \max \begin{cases} s_{i-1,j-1,k-1} + \delta(a_i, b_j, c_k) & \text{cube diagonal: no indels} \\ s_{i-1,j-1,k} + \delta(a_i, b_j, \_) & \\ s_{i-1,j,k-1} + \delta(a_i, \_, c_k) & \text{face diagonal: one indel} \\ s_{i,j-1,k-1} + \delta(\_, b_j, c_k) & \\ s_{i-1,j,k} + \delta(a_i, \_, \_) & \\ s_{i,j-1,k} + \delta(\_, b_j, \_) & \text{edge diagonal: two indels} \\ s_{i,j,k-1} + \delta(\_, \_, c_k) & \end{cases}$$

Computational complexity

- For 3 sequences of length n, the run time: $O(7n^3)$
- For 4 sequences of length n: $O(15n^4)$
- For k sequences of length n: $O((2^k-1)n^k) = O(2^kn^k)$
- Dynamic programming approach for alignment between two sequences is easily extended to k sequences but it is impractical due to exponential running time
- Computing exact MSA is computationally almost impossible, and in practice heuristics are used (progressive alignment)

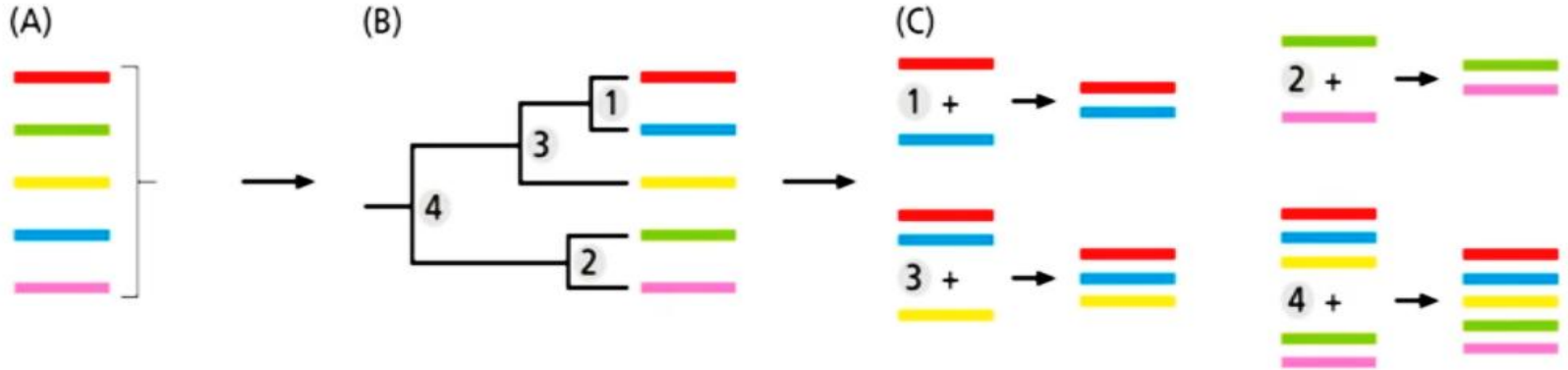
Progressive alignment for MSA

Three-step process:

1) Construct pairwise alignments

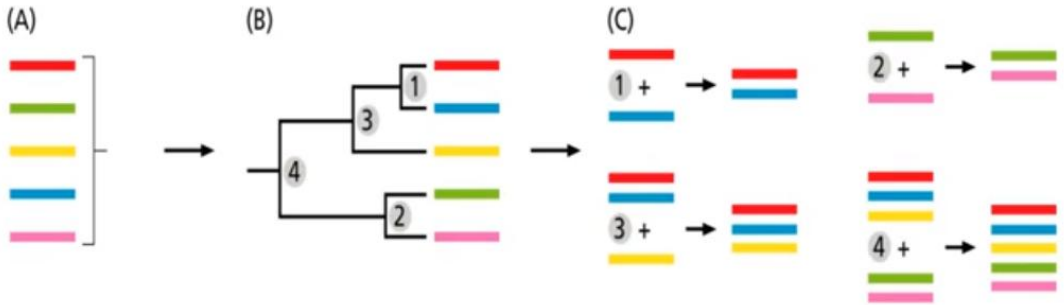
2) Build guide tree

3) Progressive alignment guided by the tree



Computational complexity

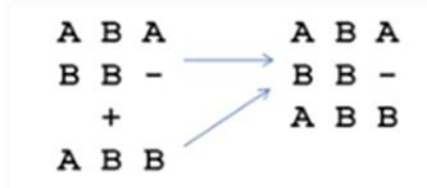
- k sequences, each with length n
- Each pairwise alignment: $O(n^2)$
- Building the phylogenetic tree:
  - $O(k^2)$ pairwise comparisons
  - $O(k^2n^2)$ time
  - Done once to construct a phylogenetic tree
- Number of merge steps:
  - $O(k)$ steps
  - $O(kn^2)$ time
- Overall time:
  - $O(k^2n^2+kn^2) = O(k^2n^2)$



Limitation of progressive alignment

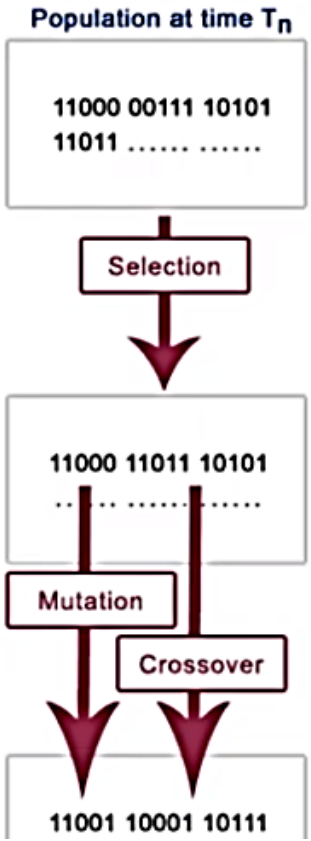
The order of alignment matters

If we align ABA and BB first:



But if we align ABA and ABB first:

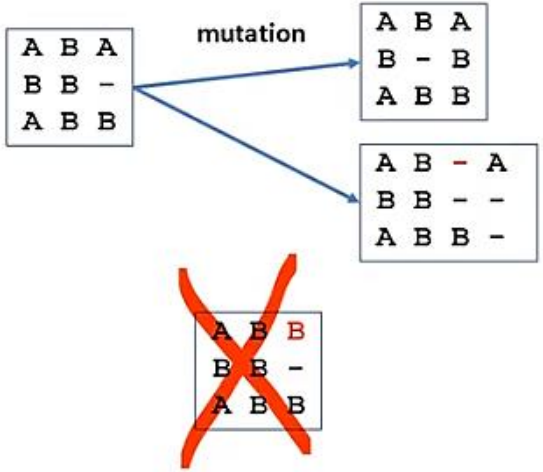
MSA using Genetic algorithm



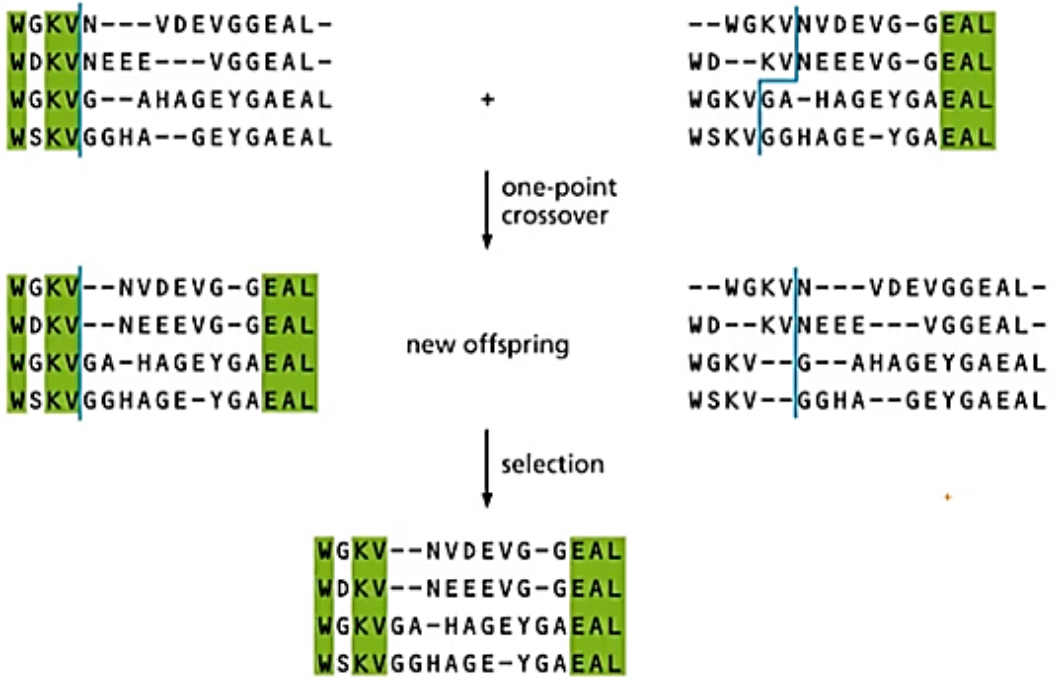
Fitness values:

F(11000) = 0.8
F(00111) = 0.2
F(10101) = 0.6
F(11011) = 0.7

MSA using Genetic algorithm



Mutations



Crossover

Database searching

Database searching

- Search query sequence against all sequence in database
- Calculate score and select top sequences
- Dynamic programming is best
- Approximation Algorithms : FASTA BLAST

FASTA

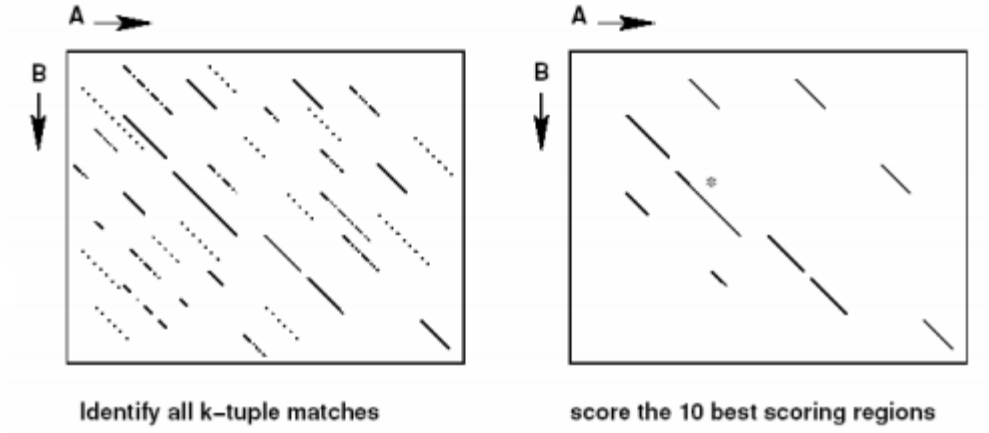
- Fast sequence search
- Based on dot plot
- Identify identical words (k-tuples)
- Search significant diagonals
- Use PAM 250 for further refinement
- Dynamic programming for narrow region

BLAST

- Heuristic method to find the highest scoring based on local optimal alignments
- Based on ungapped sequence alignments
- Allow multiple hits to the same sequence

FASTA

Localize the 10 best regions of similarity between the two seq. Each identity between two "word" is represented by a dot

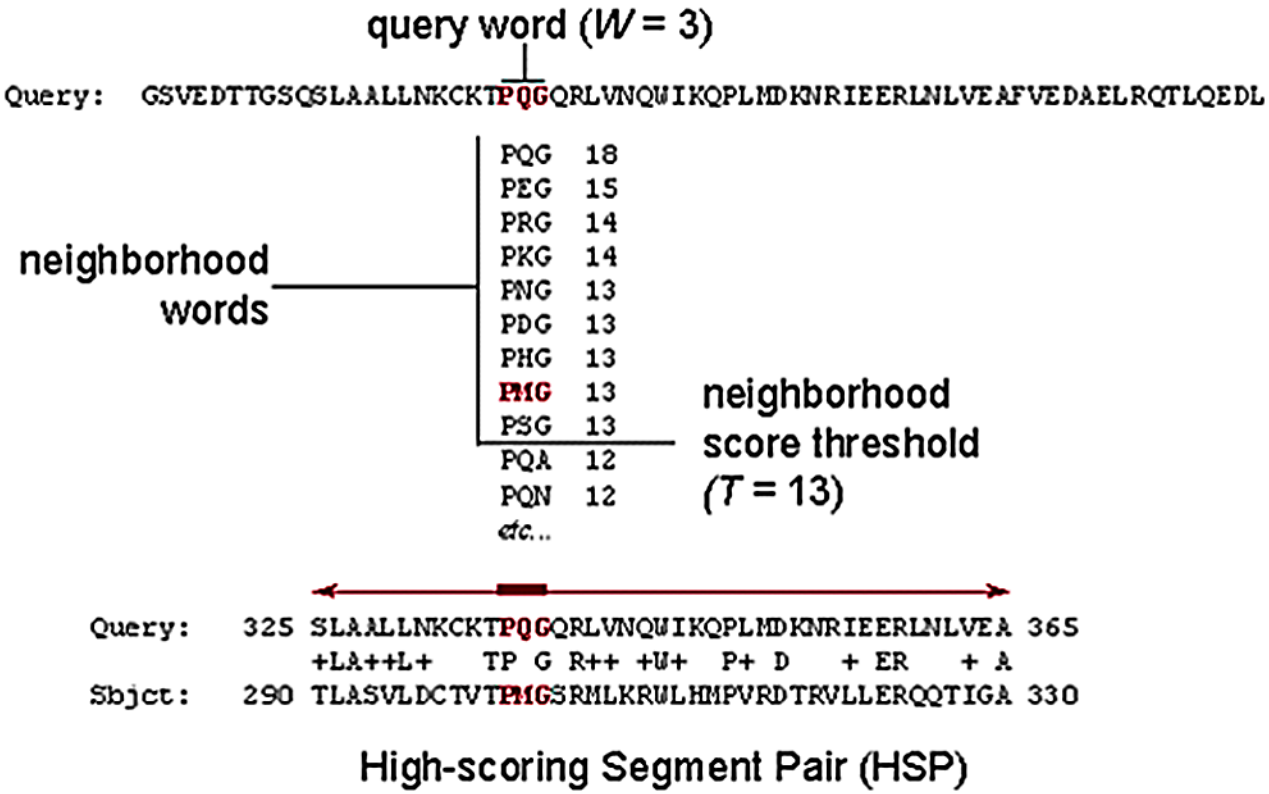


Each diagonal: ungapped alignment.
Smaller the k, sensitive the method but slower

Find the best combination of the diagonals-> compute a score.
Only those sequences with a score higher than a threshold will go to the fourth step

DP applied around the best scoring diagonal.

BLAST


query word (W = 3)
Query: GSVEDTTGSQSLAALLNKCKTPQGQRLVNQWIKQPLMDKNRIEERLNLVEAFVEDAELRQTLQEDL
neighborhood words
PQG 18
PEG 15
PRG 14
PKG 14
PNG 13
PDG 13
PHG 13
PMG 13
PSG 13
PQA 12
PQN 12
etc...
neighborhood score threshold (T = 13)
Query: 325 SLAALLNKCKTPQGQRLVNQWIKQPLMDKNRIEERLNLVEA 365
+LA++L+ TP G R++ +W+ P+ D + ER + A
Sbjct: 290 TLASVLDCTVTPMGSRMLKRWLHMPVRDTRVLLERQQTIGA 330
High-scoring Segment Pair (HSP)

BLAST

Scoring parameters: Score, E-value, Query coverage, Identity

Formula for the E-value (likely that a hit of a given score could occur just by chance in the database) in BLAST:

$$E = K \cdot m \cdot n \cdot e^{-\lambda S}$$

Where,

E = expected number of alignments with score ≥ S
m = length of the query sequence
n = total length of the database (sum of all sequences)
S = raw alignment score
K = a statistical constant, depends on the scoring system and sequence composition
λ (lambda) = scaling factor (related to the scoring matrix and gap penalties) that scales the raw score into a probability scale.

Interpreting the E-value:
E = 10 → you'd expect ~10 hits with this score or better to occur just by chance.
E = 0.01 → only 1 in 100 searches of this size would produce a hit this good by chance.
E ≈ 0 (e.g., 1e-100) → essentially certain that the hit is real.

BLAST variants