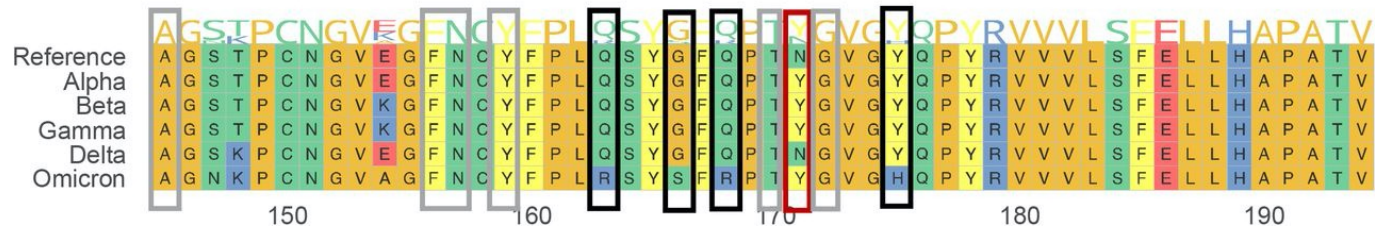
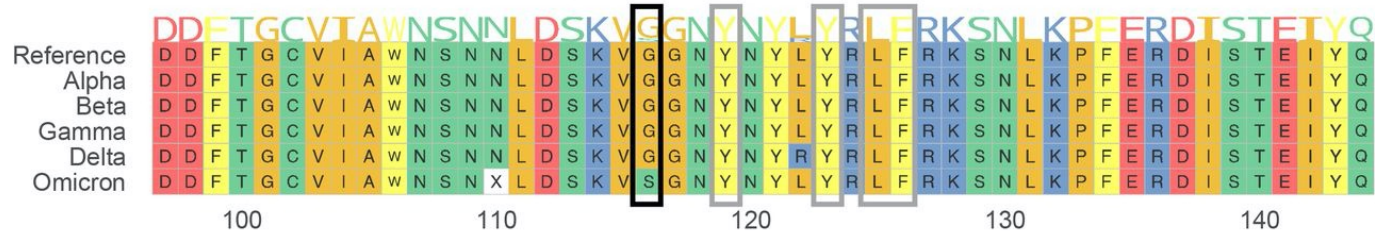
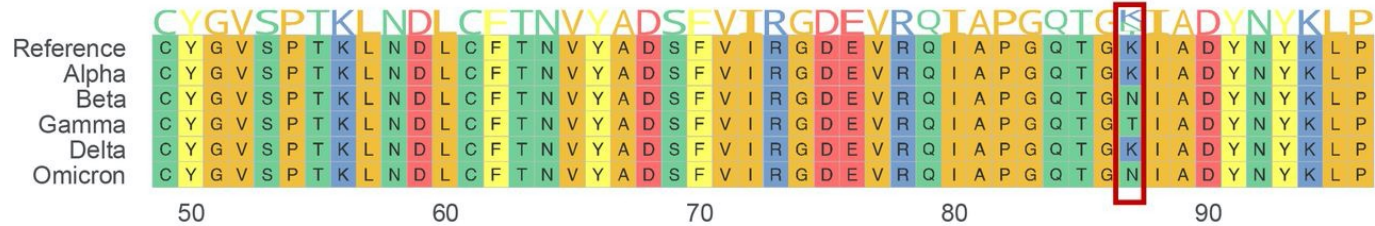
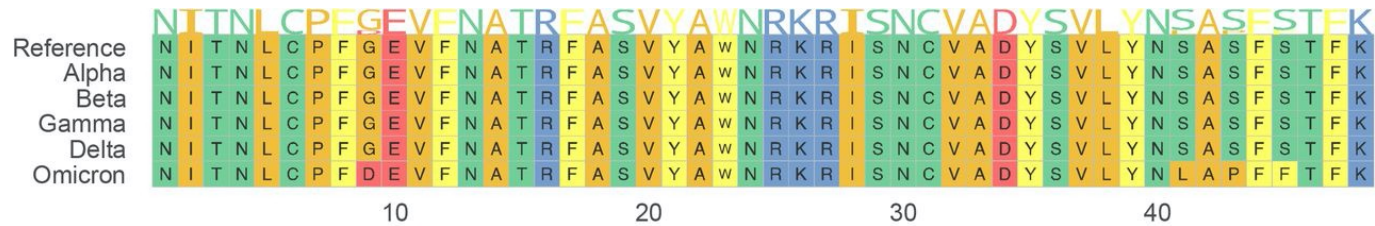

Multiple Sequence Alignment

Multiple Alignment Example

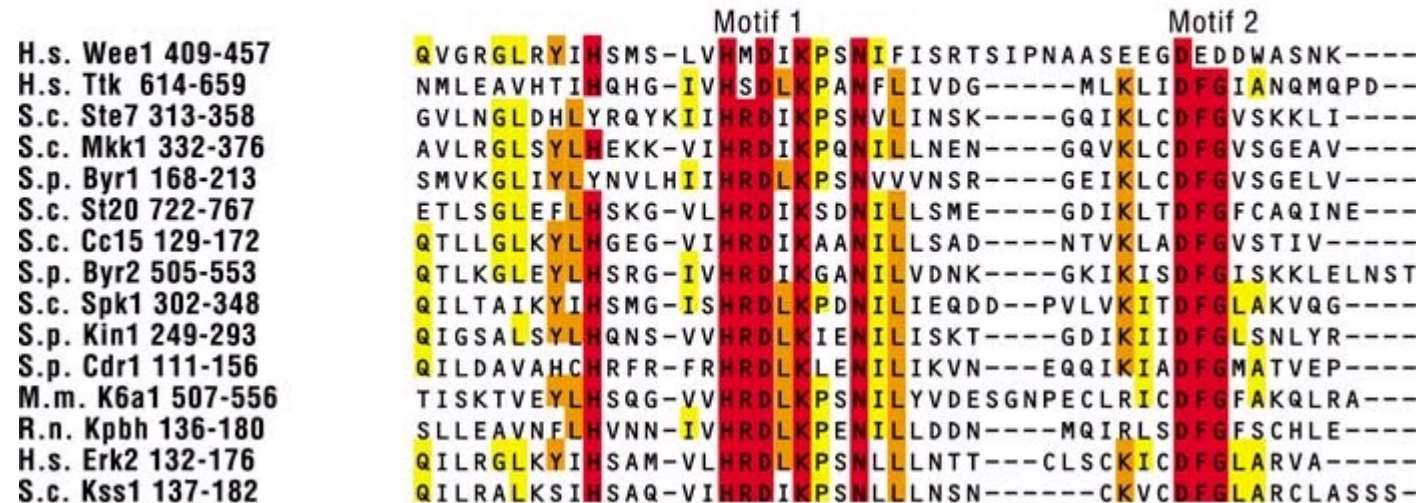


Contacting residues between SARS-CoV-2 and ACE2. Boxes denote the contacting residues. Black boxes denote mutations unique to omicron, red boxes denote mutations occurring in multiple variants, and grey boxes denote no mutations in any variant.

Image credit: Li et al.
doi: <https://doi.org/10.1101/2021.12.08.471688>

From pairwise to multiple

- A multiple sequence alignment of k sequences is an insertion of gaps in the positions of the sequences, just like a pairwise alignment.



© 1999–2004 New Science Press

- “Two homologous sequences whisper, a multiple alignment shouts loudly” -- Arthur M. Lesk

Heuristic Algorithm for Multiple Alignment

- There is a simple algorithm that can merge pairwise alignments to a multiple alignment.
- The algorithm does not guarantee the optimality of the result. But runs relatively fast.

Merging Two Pairwise Alignments

- Suppose we have the following two pairwise alignments
- t : A-GAGC
- s_1 : ATGAGC
- and
- t : AGA-GC
- s_2 : AGTTGC
- Our main idea is to use the shared sequence t to construct a multiple alignment of s_1 , t , and s_2 .

Merging Two Pairwise Alignments



*pairwise alignment
"induced" by the multiple
alignment.*

- Algorithm: Insert gaps to the two alignments, so that the superstrings for t become the same in the two alignments. Then put the two alignments together.
- Property: Maintains the pairwise alignment unchanged if ignoring the all-gap columns of the pairwise alignment.

Merging Two ~~Pairwise~~ ^{multiple} Alignments



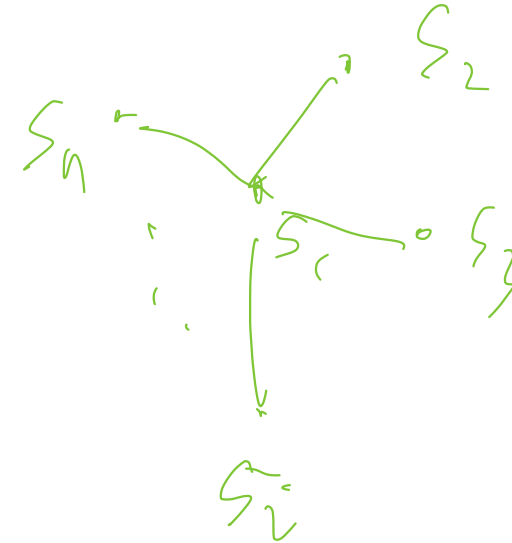
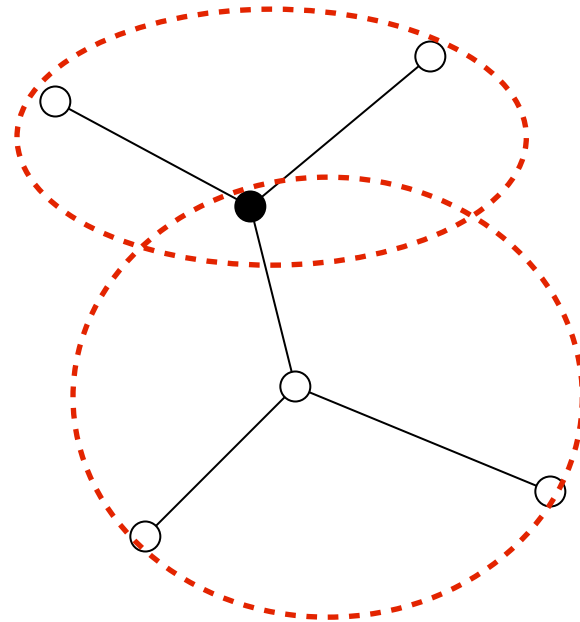
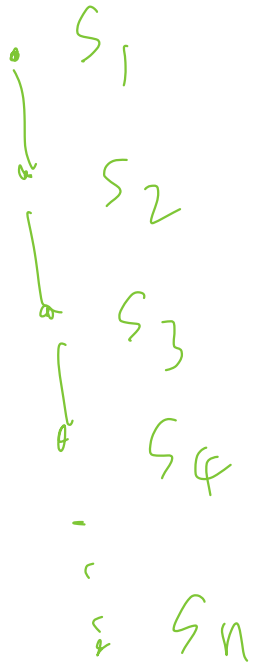
- The process can be continued to merge two multiple alignments together as a bigger one.
- Note that the obtained multiple alignment may not be optimal.

Heuristic Algorithm for Multiple Alignment

- Input: s_1, s_2, \dots, s_n .
- Algorithm:
 - Let A = pairwise alignment of s_1 and s_2 .
 - For i from 3 to n .
 - Construct pairwise alignment P between s_1 and s_i .
 - Let $A = \text{merge}(A, P, s_1)$, i.e., merging A and P using s_1 as the template.
 - Return A .

Progressive Alignment

- The order of the merging is important to get good (but not optimal) multiple alignment.
- We want to merge similar sequences first.
- One way is to construct a minimum spanning tree, and then merge using the shared vertices.

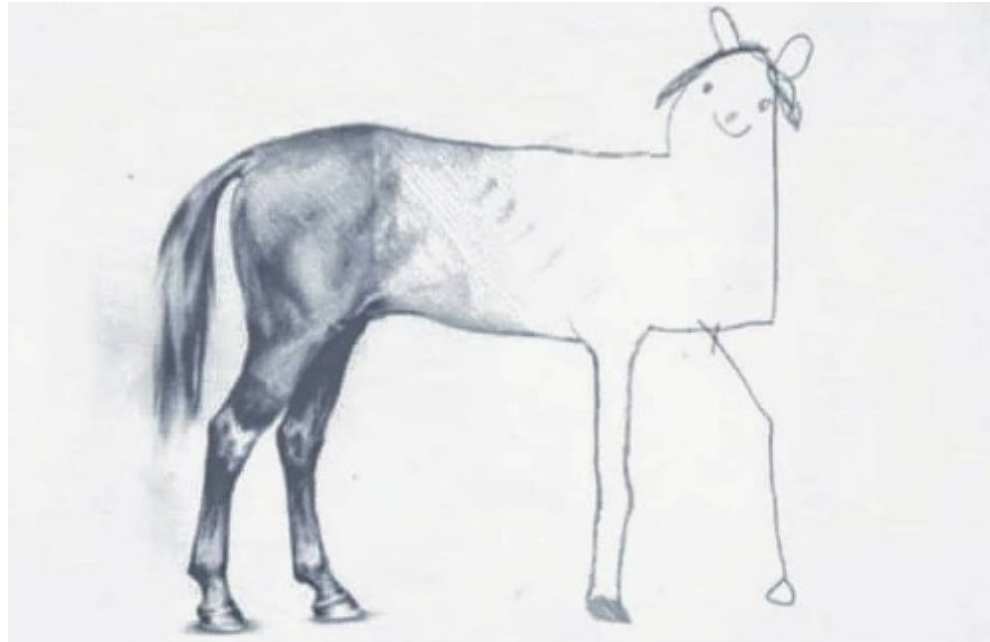


Heuristic Algorithm

- The exact algorithm for multiple alignment is super-polynomial. Before we study it, let's examine a heuristic algorithm.
- A heuristic algorithm is an algorithm that gives up quality for speed. It usually does not offer any performance guarantee in terms of quality.
- Well... We already sacrificed the quality because of a simple scoring function any way.
- If we cannot afford exponential time, the best we can ask for is a suboptimal solution.
- But practically, it might work sometimes.

Heuristic Algorithm

- We do not want to spend super-polynomial (e.g. exponential) time.



“You get what you pay for”.

Exact Algorithm for Multiple Alignment

- When the optimal alignment is needed. There is an exact algorithm as well.