MIMM4750G

Sequence alignment

What is an alignment?

- So far we have talked about comparing sequences residue-by-residue with a score matrix.
- The underlying assumption is that these sequences are aligned.
- An alignment is a hypothesis about how residues (nt, aa) in homologous sequences are related to residues in a common ancestor.
- This is not trivial because of insertions and deletions.

```
Query  1    CTRPNNTRKSVSITIGPGRASYATG---GQAHC  30
           ||||||||||  |||||||||||||   |||||
Sbjct  95   CTRPNNTRKS--ITIGPGRASYATGGIIGQAHC  125
```

Gap characters

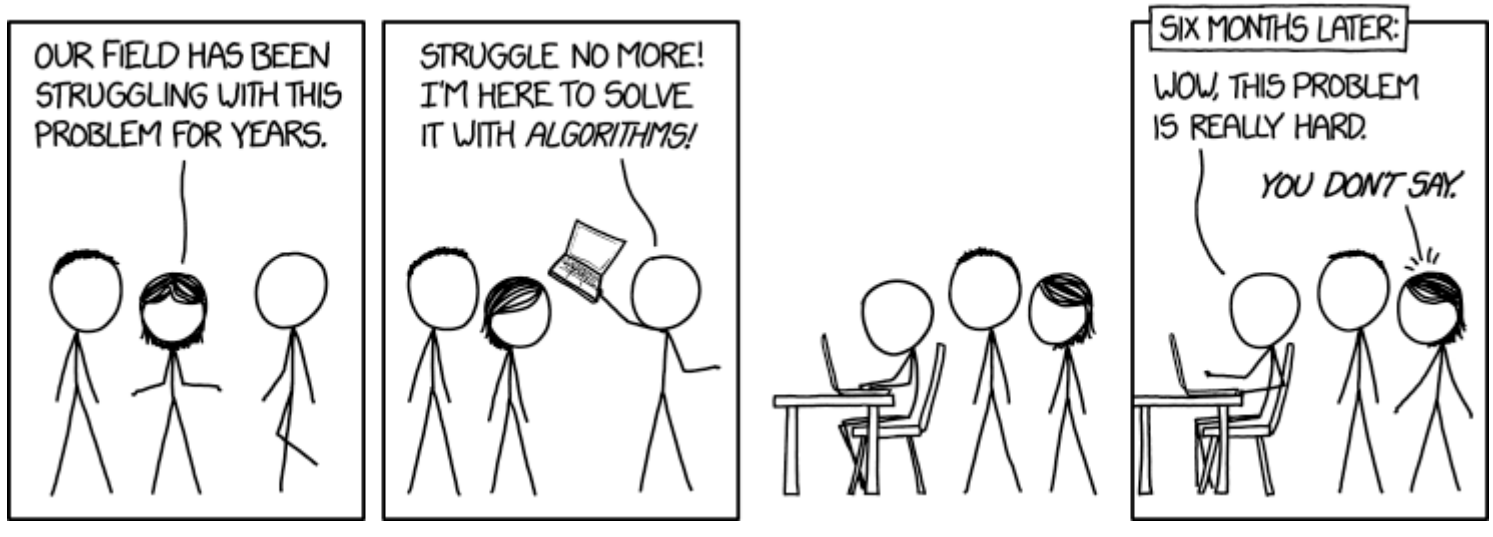
- The presence of an insertion or deletion is indicated by a gap character.
- By convention, we use a single dash "–" for each indel.
- Some programs use non-standard characters like "`.`", "`~`" or "`x`".
- Without additional information, we cannot tell whether a gap is the result of an insertion in the longer sequence or a deletion in the shorter.
- Hence we use the portmanteau *indel* (insertion/deletion).

Pairwise alignment

- Thorne, Kishino and Felsenstein proposed a simple model of indel evolution (TKF91).
  - Insertions and deletions at constant rates.
  - One nucleotide at a time.
- Solving for the maximum likelihood of TKF91 can be used to align sequences.
- However, TKF91 and subsequent extensions of the model are not feasible for many long sequences.
- There is an infinite number of possible evolutions - *e.g.*, unsampled insertions.

Heuristic methods

- Until TKF91-type methods become feasible, we continue to use heurstic methods.
- A heuristic is an algorithm for solving a problem that has no theoretical guarantee of being accurate.
- In practice, heuristic is designed to quickly produce a solution that is "good enough".



Score matrices (again)

- A major feature of heuristic methods of alignment
- Remember a score quantifies the likelihood of a residue being replaced by another.
- Find which alignment of two sequences maximizes the score.
- A simple score matrix for nucleotides: $+1$ (match), -1 (mismatch).

	A	C	G	T
A	+1	-1	-1	-1
C	-1	+1	-1	-1
G	-1	-1	+1	-1
T	-1	-1	-1	+1

Gap penalties

- We need to penalize the score for gaps, or else an alignment gets gaps for free:

```
A-C-G-T   ACGT
-A-C-T-   AC-T
```

- The left option is obviously a terrible alignment!
- If we use match/mismatch scores of +1 / –1 and a gap penalty of –1, then **what are the scores** for these alignments?

Penalizing longer gaps

- If a gap spans 2 or more residues, we might want to enforce a milder or more severe penalty.
- For a gap of length l:
  - **Linear gap penalty** = $-ld$, where d is a constant per-gap penalty.
  - **Affine gap penalty** = $-d-(l-1)e$, where d is the gap *opening* penalty and e is the gap *extension* penalty.
- It is more common to use affine gap penalties.

Terminal gaps

- Terminal gaps are a contiguous run of gaps on either extreme left or right of a pairwise alignment.
- Also known as "leading" and "trailing" gaps.

```
ACTGATC    ACTGATC
---GATC    ACTG---
```

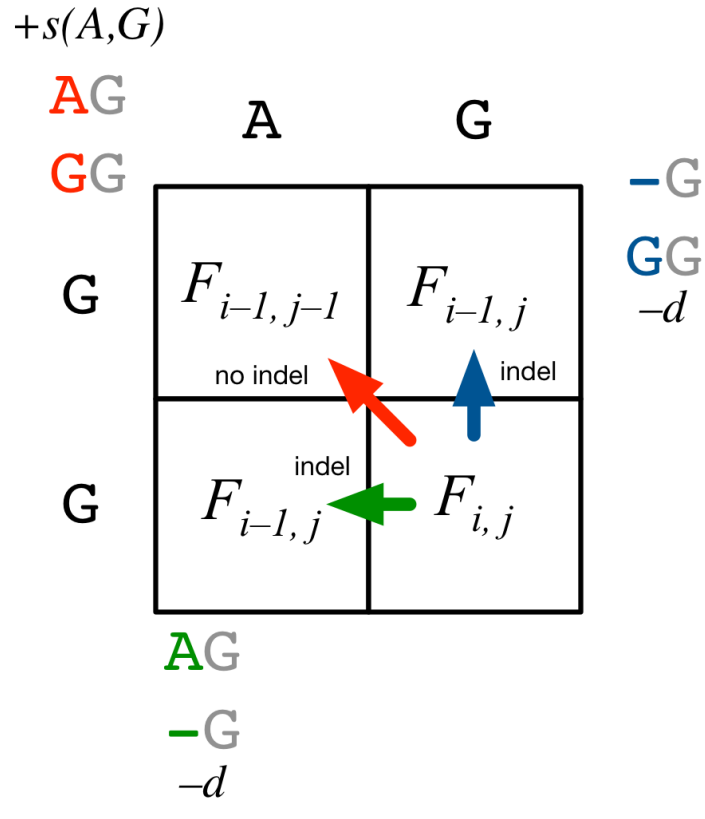
- We might not want to penalize these when aligning partial (incomplete) sequences.

Dynamic programming

- Dynamic programming is a fundmanetal concept in computer science.
- A complex problem can be broken down into a sequence of much smaller, simpler *recursive* problems.
- "Recursive" means that the problems are nested within each other. Solving one is part of solving another.
- Retrieving the solutions to problems nested within the next problem saves work!

Filling the F matrix

- Most heuristics for sequence alignment operate on a dynamic programming matrix (F).
- Like a *dot plot*, it is a matrix where one sequence labels the columns, and the second labels the rows.
- Each entry in F is calculated from the entries above, to the left, or diagonally up and left.



Sequence 1 GATTACA

Sequence 2 GTCGACG

Match Score	Mismatch Score	Gap Score
1	-1	-2

Compute Optimal Alignment | Clear Path | Custom Path

```
G A T T A C A
G T C G A C G
Score = -1
```

		G	T	C	G	A	C	G
	0	-2	-4	-6	-8	-10	-12	-14
G	-2	1	-1	-3	-5	-7	-9	-11
A	-4	-1	0	-2	-4	-4	-6	-8
T	-6	-3	0	-1	-3	-5	-5	-7
T	-8	-5	-2	-1	-2	-4	-6	-6
A	-10	-7	-4	-3	-2	-1	-3	-5
C	-12	-9	-6	-3	-4	-3	0	-2

JS adapted from https://github.com/drdrsh

Local versus global alignment

- *Global* alignment (*e.g.*, Needleman-Wunsch) requires the sequences to be aligned end-to-end — terminal gaps are penalized.
- *Local* alignment (*e.g.*, Smith-Waterman) relaxes this requirement — it does not penalize terminal gaps.
- Use local alignment when you know that the query is shorter than the reference, or vice versa.

Multiple sequence alignment

- It is not trivial to extend pairwise alignment to more than two sequences.

```
1 AC-GT    1 ACGT    1 ACGT
  || ||      | ||    2 A-GT
3 ACAGT    2 A-GT    3 ACAGT
```

- Most alignment programs use a *progressive* algorithm to propagate information from pairwise alignments to all sequences.

Progressive sequence alignment

- A *guide tree* determines which pair of sequences are the most closely related.
- Three types of actions:
    1. Align closely related sequence pair.
    2. Align a sequence to group of sequences.
    3. Align two groups of sequences.
- CLUSTALW averages scores across residues for each position between groups.
- Preserve all gaps as we proceed down to the root.

The paradox of guide trees

- Alignment is more accurate when the guide tree is closer to the actual tree.
- Most tree-building methods require an alignment.
- We have to use an alignment-free clustering method to build the guide tree.
- For example, *MUSCLE* builds a guide tree by counting k-mers.

Iterative alignment

- After we build an alignment, we can reconstruct a tree.
- That tree can be plugged back into the alignment process as *the next guide tree*.
- This method should incrementally improve the accuracy of alignment.
- Seldom used in practice!

Software

- This is an incomplete list:

Name	Publication	Description
CLUSTALW	1994	One of the first MSA programs to achieve widespread popularity. Less accurate than more recent programs.
T-coffee	2000	Initially performs pairwise alignments of the sequences, but uses a mix of local and global alignments.
MAFFT	2002	Uses a fast Fourier transform to rapidly identify homologous regions between sequences.
MUSCLE	2004	Uses an alignment-free *k*-mer based distance to generate a guide tree, and iteratively refines the alignment by partitioning the tree into subtrees.

More software

Name	Publication	Description
BAli-Phy	2006	Uses Bayesian sampling to jointly estimate the alignment and the phylogeny. We nearly always assume the alignment is a known, fixed quantity when reconstructing the phylogeny. BAli-Phy infers the alignment and the tree *at the same time*. Computationally challenging.
PRANK	2008	Assumes that sequence insertions usually lack evolutionary homology to other insertions. Tends to spread insertions out to such an extreme that the resulting alignment becomes a sparse scaffold of isolated insertions.
SATé	2009	A pipeline for iterative alignment to estimate both the alignment and tree.

Manually editing your alignment

- **Always look at your data!**
- There are several programs available to visually inspect and manually edit a sequence alignment.
- AliView
- SeaView