Local Alignment: Smith-Waterman algorithm

- Example: a shared common domain of two protein sequences; extended sections of genomic DNA sequence.
- Sensitive to detect similarity in highly diverged sequences.
- Algorithm: similar to global alignment with modified boundary conditions and recurrence rules.
  - The top row and left column are now filled with 0.
  - If the (sub-)alignment score becomes negative, restart the search:

$$F(i,j) = max \begin{cases} 0, \\ F(i-1,j-1) + s(x_i, y_j), \\ F(i-1,j) - d, \\ F(i,j-1) - d. \end{cases}$$

  - Traceback: from the maximum of $F(i,j)$ **in the whole matrix** to the first 0.
- Example: the optimal local alignment between HEAGAWGHEE and PAWHEAE is AWGHE::AW-HE.
- Issue: In gapped alignments, the expected score for a random match may be positive.

Repeated matches

- Example: repeated domains in proteins.
- Asymetric alignment: finding multiple (non-overlapping) matches of one sequence in the other.
- We want each match's score to be greater than T. The alignment score is the sum of the match scores substracted by T: $score(alignment) = \Sigma_i(score(match_i) - T)$.
- Boundary conditions: $F(0,0) = 0$.

$$F(i,0) = max \begin{cases} F(i-1,0), \\ F(i-1,j) - T, j = 1, ..., m; \end{cases}$$

- Recurrence rules:

$$F(i,j) = max \begin{cases} F(i,0), \\ F(i-1,j-1) + s(x_i, y_j), \\ F(i-1,j) - d, \\ F(i,j-1) - d. \end{cases}$$

- $F(i,0)$ is the score of optimal alignment of $y_{1..m}$ to $x_{1...(i-1)}$.

- The best alignment score is $F(n+1,0)$.
- Traceback from cell $(n+1,0)$ until meet 0.
- Complexity is still $O(nm)$.
- Increasing T may exclude matches. Decreasing T may split them.
- Dilemma: a match of score $\approx T$ flanked by two matches of high positive scores.

Overlap matches

- Example: when comparing fragments of genomic DNA sequence to each other, one sequence may overlap the other, partially or completely.
- This is similar to global alignment, except for overhaning ends. An alignment will start from the top or left border of the matrix, and finishes on the right or bottom border.
- Boundary conditions: $F(i,j) = 0$ if $i = 0$ or $j = 0$.
- Recurrence rules like those of the global alignment.
- Repeat version:

$$
\begin{aligned}
F(i,0) &= max \begin{cases} F(i-1,0), \\ F(i-1,j) - T; j = m \text{ if } i \leq n, \text{ otherwise } j = 1..m; \end{cases} \\
F(i,j) &= max \begin{cases} F(i-1,j-1) + s(x_i, y_j), \\ F(i-1,j) - d, \\ F(i,j-1) - d. \end{cases}
\end{aligned}
$$

Hybrid match conditions

- Variants can be formulated by changing boundary conditions and recurrence rules.
- Example 1: to find mutiple matches of y that appear in tandem copies, we use recurrence rules in the previous slide with a slight modification.

$$F(i,1) = max \begin{cases} F(i-1,0) + s(x_i, y_1), \\ F(i-1,n) + s(x_i, y_1), \\ F(i-1,1) - d, \\ F(i,0) - d. \end{cases}$$

- Example 2: finding a match starting at the beginning of both sequences, ending at any point.

  - setting only $F(0,0) = 0$.

  - using global recurrence rules, find maximum over the whole matrix.
- Ideally, we should be able to use algorithm designed with specific prior knowledge. In practice, people use good implementations available for a few standard cases then postprocess the results.

Other models

- Up to this point, we use the gap penalty $\gamma(g) = -g * d$.
- Affine gap score penalises gap extension less than gap opening: $\gamma(g) = -d - (g-1) * e$, where $e < d$.
- With general gap penalty $\gamma(g)$:

$$F(i,j) = max \begin{cases} F(i-1, j-1) + s(x_i, y_j), \\ F(k,j) + \gamma(i-k), k = 0, ..., i-1, \\ F(i,k) + \gamma(j-k), k = 0, ..., j-1. \end{cases}$$

- Complexity is $O(n^3)$ with general gap function, can be reduced to $O(n^2)$ if k can be bounded.

$O(n^2)$ implementation for affine gap scores

- $M(i,j)$, or $I_x(i,j)$, or $I_y(i,j)$ is respectively the best score given that x_i is aligned to y_j, or x_i to a gap, or y_j to a gap.
- The recurrence rules:

$$\begin{aligned}
M(i,j) &= max \begin{cases} M(i-1,j-1) + s(x_i,y_j), \\ I_x(i-1,j-1) + s(x_i,y_j), \\ I_y(i-1,j-1) + s(x_i,y_j); \end{cases} \\
I_x(i,j) &= max \begin{cases} M(i-1,j) - d, \\ I_x(i-1,j) - e; \end{cases} \\
I_y(i,j) &= max \begin{cases} M(i,j-1) - d, \\ I_y(i,j-1) - e. \end{cases}
\end{aligned}$$

- Implicit assumption: a deletion won't be followed directly by an insertion. Generally true if $-d-e \leq$ lowest mismatch score.
- The recurrence rules can be simulated by a finite state automation (FSA), which can be turned into a HMM!

Linear space alignment

- Simple to achieve if only maximal score is needed.
- To find the optimal global alignment: *divide and conquer* principle.
- Let $u = \lfloor \frac{n}{2} \rfloor$ and v be the row such that the cell (u, v) is on the optimal alignment.
- Split the DP matrix into 2 smaller submatrices: one from $(0, 0)$ to (u, v), one from (u, v) to (n, m).
- Recursively split down until the submatrices are small enough for using standard $O(n^2)$ DP procedure.
- Runtime still $O(n^2)$.
- Way to find v: keep tracks of $c(i, j)$ such that $c(i, j) = v$ for $i > u$. If the cell (i', j') precedes (i, j), set $c(i, j) = j'$ if $i = u$, else $c(i, j) = c(i', j')$.

Substitution matrices

- Recap: *scores are to measure the relative likelihood that the sequences are related as opposed to being unrelated.*
- Model M for being related, R for unrelated. Prior probability $P(R) = 1 - P(M)$.
- In R, the residues in the sequences occur independently:

$$P(x, y|R) = \prod_i q_{x_i} \prod_j q_{y_j}$$

- In M, aligned residue pairs have a joint distribution p_{ab}:

$$P(x, y|M) = \prod_i p_{x_i y_i}$$

- Let $S = log\left(\frac{P(x,y|M)}{P(x,y|R)}\right)$ and $S' = S + log\left(\frac{P(M)}{P(R)}\right)$.
- $P(M|x, y)$ is probability of being relared versus being unrelated:

$$P(M|x, y) = \frac{e^{S'}}{1 + e^{S'}}$$

- $P(M|x,y)$ increases with S'. If we fixed the prior log-odds ratio $log\left(\frac{P(M)}{P(R)}\right)$, we only need to maximize S'.

$$S' = \sum_i log\left(\frac{p_{x_i y_i}}{q_{x_i} q_{y_i}}\right)$$

- The substitution matrix contains the log likelihood ratio of residue pair (a,b) being related as opposed to being unrelated: $s(a,b) = log(\frac{p_{ab}}{q_a q_b})$
- Counting frequencies in confirmed alignments won't work!

  - protein sequences come in families, so alignments tend not to be independent from each other.

  - when divergence is higher, p_{ab} tends to background frequency $q_a q_b$, so scores should correspond to expected divergence of the sequences.
- Common matrices PAM and BLOSUM take the difficulties into account.
- Prior odds ratio becomes important when searching database for a possible significant match. When the ratio is fixed, the probability of finding a match increases with the number of queries. Changing the ratio can corrigate this probability.