SGN-6156, Lecture 2
Biological sequence analysis

Harri Lähdesmäki, harri.lahdesmaki@tut.fi

(part of the material by Juha Kesseli)

Department of Signal Processing,

Tampere University of Technology

02.04.2008

Optimal solution with dynamic programming

- Given a scoring model, an optimal alignment needs to be founds.
- There are exceedingly many possible alignments, thus it is prohibitive to go through all exhaustively.
- Optimal solution(s) can be found efficiently by dynamic programming.
- Dynamic programming is guaranteed to find the best alignment(s).
- In dynamic programming, the optimal solution to the sequence alignment problem is found by combining partial optimum solutions.
- Heuristic methods can improve computational efficiency even further, but they are not guaranteed to find the best solution(s).
- Given the log-odds scoring scheme, we want to maximize the score.
- Assume linear gap model (for simplicity).

- Given the scoring model, alignments have a probabilistic interpretation as well. The dynamic programming presented below can be viewed as a Hidden Markov model solution.

Global alignment: Needleman-Wunsch algorithm

- Global alignment allowing gaps.
- Construct a matrix $F(i,j)$ (recursively) which contains the score of the best alignment between subsequences $x_1,\ldots,x_i$ and and $y_1,\ldots,y_j$
- Initialize $F(0,0)=0$ and proceed from top left to bottom right.
- If $F(i-1,j-1)$, $F(i,j-1)$ and $F(i-1,j)$ are known then we can compute $F(i,j)$
- Recursion:

$$F(i,j)=\max\begin{cases}F(i-1,j-1)+s(x_i,y_j) & (x_i \text{ is aligned to } y_j)\\ F(i-1,j)-d & (x_i \text{ is aligned to gap})\\ F(i,j-1)-d & (y_j \text{ is aligned to gap}).\end{cases}$$

- See Figure 2.4 in (Durbin et al 1998).
- $F(i,0)$ represents alignments of $x_1, \ldots, x_i$ to all gaps in the beginning of y, thus initialize $F(i,0) = -id$ (similarly $F(0,j) = -jd$).
- $F(n,m)$ contains the best score for aligning x and y.
- During the recursion, keep pointers from $F(i,j)$ to the cell from which it was computed.
- Following the reverse pointers gives the optimal alignment itself. This is called traceback.
- See Figure 2.5 in (Durbin et al 1998).
- Dynamic programming relies on the additivity of the score.

Algorithmic complexity, big-O notation, recap

- A function g is a tight upper bound for a function f, denoted as $f(n) \in O(g(n))$, if and only if there exist positive constants n_0 and c such that

$$n > n_0 \Rightarrow 0 \leq f(n) \leq cg(n).$$

- Needleman-Wunsch algorithm needs to store $(n+1) \times (m+1)$ values and each value is computed by three sums and three $\max$-operation.
- Thus, both computational complexity and memory requirements are $O(nm)$.
- For $n \approx m$ this means $O(n^2)$
- For large-scale problems, $O(n^2)$ is typically considered relatively efficient but $O(n^3)$ is already a bit slow.

Local alignment: Smith-Waterman algorithm

- Local alignment between subsequences of x and y allowing gaps.
- For highly diverged species, local alignment can be more sensitive than global alignment.
- Smith-Waterman is similar with Needleman-Wunsch, recursion:

$$F(i,j) = \max \begin{cases} 0 & \text{(Start a new alignment)} \\ F(i-1,j-1) + s(x_i, y_j) & (x_i \text{ is aligned to } y_j) \\ F(i-1,j) - d & (x_i \text{ is aligned to gap}) \\ F(i,j-1) - d & (y_j \text{ is aligned to gap}). \end{cases}$$

- Option 0 corresponds to starting a new alignment. Consequently, initialization needs to be changed to $F(i,0) = 0$ for all i (similarly $F(0,j) = 0$).

- Alignment can end anywhere in the recursion matrix.
- The optimal local alignment can now be found by starting from the position with the highest score and following (tracebacking) the path until reaching score zero.
- See Figure 2.6 in (Durbin et al 1998).
- For local alignments to be found, the expectation of the scores needs to be negative for alignment of random sequences.
- Otherwise, long matches between totally unrelated sequences will have high scores and the optimal alignment would be nearly global.

Aligning repeated matches

- Repeated matches can be important if one of the sequences is long.
- E.g. a repeated short domains or motif in a protein (asymmetric problem).
- Assume one is looking for alignment scores that are above a threshold T.
- x is the longer sequence that supposedly contains many occurrences of a motif y.
- See Figure 2.7 in (Durbin et al 1998).
- $F(i, j)$ is now the best sum of match scores to $x_1, \ldots, x_i$ (assuming x_i is in a matched region and match ends at (i, j)), and $F(i, 0)$ is now the sum of the best completed match scores.

- Yet another recursion(s):

$$F(i,0) = \max \begin{cases} F(i-1,0) & \text{(unmatched region)} \\ F(i-1,j) - T, j = 1, \ldots, m & \text{(end of match)} \end{cases}$$

and

$$F(i,j) = \max \begin{cases} F(i,0) & \text{(Start of a new match)} \\ F(i-1,j-1) + s(x_i, y_j) & (x_i \text{ is aligned to } y_j) \\ F(i-1,j) - d & (x_i \text{ is aligned to gap}) \\ F(i,j-1) - d & (y_j \text{ is aligned to gap}). \end{cases}$$

- The optimum score in $F(n+1,0)$.
- The match alignments can be tracebacked from $(n+1,0)$ to $(0,0)$.
- **Recall:** the above alignment methods have a probabilistic interpretation!

More complex aligning problems

- For example, a different search strategy is needed if
  - we expect that one sequence contains the other or if they overlap (overlap matches).
  - a repetitive sequence is found in tandem copies (i.e., not separated by gaps)
  - etc.
- Linear gap model is not realistic.
- Dynamic programming works with an arbitrary gap model $\gamma(g)$, but in general requires $O(n^3)$ time.

- Recursion:

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

- Fortunately, the affine gap model has a dynamic programming solutions that works in time $O(n^2)$.

Heuristic alignment methods

- The above aligning methods are exact, in that they are guaranteed to find the optimal solution(s).
- For long sequences/large sequence databases, running time of $O(nm)$ can become an issue.
- Heuristic approaches to sequence alignment are often used in practice, although they are not guaranteed to find the optimal solutions. Heuristics can also be complemented with a dynamic programming solution of an optimal alignment when the search can be restricted to make it feasible.
- The goal of heuristic methods is to search as a small fraction as possible of $F(i, j)$ but still to find high scoring matches.

BLAST

- Basic Local Alignment Search Tool (BLAST) is perhaps the most common alignment tool in use for database searches.
- The search is made faster by utilizing the fact that any interesting local alignment is very likely to contain a shorter sequence stretch of identities, or a subalignment with a very high score.
- For each subsequence of the query, BLAST scans the database for matches (typically e.g. of length 13 for DNA, 4 for proteins) that have a score that exceeds a chosen threshold.
- In the standard version, each of these local hits is then extended as an (ungapped) alignment in both directions, stopping at the maiximum scoring extension.
- The extended hits are called high-scoring segment pairs (HSPs).

- Since the extension stage is rather slow and a large number of extensions will need to be done in order to obtain sufficient sensitivity, some later versions require that there are two hits within a window of specified length in order to proceed with extension.
- Many versions of BLAST also do gapped alignment by starting a dynamic programming run for particularly promising matches.
- BLAST can be used online at the NCBI (National Center for Biotechnology Information) website or downloaded for use locally. Several implementations are available.

- There are numerous versions of BLAST for different purposes and many implementations, e.g.:
  - BLASTn, DNA query from a DNA database.
  - BLASTp, Protein query from a protein database.
  - BLASTx, DNA query from a protein database, the conceptual translation products of all six possible frames are searched.
  - PSI-BLAST, Position-specific iterative version, which builds a sequence motif out of a set of related sequences found and uses the motif to improve the search results iteratively (multiple alignment).

Linear time algorithms

- Above methods compute matrix $F(i, j)$.
- Sometimes even memory requirements can become a limiting factor.
- There are techniques that give the optimal alignment in $O(n + m)$ space and $O(nm)$ time.

Significance of scores

- Once an optimal alignment has been found, we need to determine whether it is significant.
- That is, decide whether (or judge to what extent) the found alignment
  - represents a biologically meaningful alignment
  - is just the best alignment of totally unrelated sequences.
- Two approaches:
  - Bayesian flavored model comparison
  - the traditional hypothesis testing based approach.

Bayesian model comparison

- The above methods compute the $\log$-likelihood ratio

$$S = \log\left\{\frac{P(x, y|M)}{P(x, y|R)}\right\}.$$

- A more natural quantity would be the probability that the sequences are related or unrelated, i.e., $P(M|x, y)$ and $P(R|x, y)$.
- First specify prior probabilities $P(M)$ and $P(R) = 1 - P(M)$.
- After seeing the data (i.e. x and y), we can use the Bayes' rule

$$\begin{aligned} P(M|x, y) &= \frac{P(x, y|M)P(M)}{P(x, y)} \\ &= \frac{P(x, y|M)P(M)}{P(x, y|M)P(M) + P(x, y|R)P(R)} \end{aligned}$$

and $P(R|x, y) = 1 - P(M|x, y)$.

- If one defines

$$S' = S + \log \left\{ \frac{P(M)}{P(R)} \right\}$$

and logistic function

$$\sigma(x) = \frac{e^x}{1 + e^x},$$

then

$$P(M|x, y) = \sigma(S').$$

- Prior odds $S' - S$ can be particularly useful when a large number of different alignments need to be processed.
- If a fixed prior odds is used, then the average number of false positive increases linearly. Thus, prior odds should should be inversely proportional to the number of samples.
- Note that the above model comparison is not entirely Bayesian because the models/parameter are fixed.

Hypothesis testing, recap

- A general approach to compute a significance value for a hypothesis:

  1. Choose a null hypothesis H_0 and its complement H_1
  2. Choose a proper test statistic T
  3. Derive the null distribution of T for a random sample D from H_0

     $$f(X) = P(T(D) = X|H_0)$$

  4. Compute the value of the test statistic using the observed data D_{obs}, $T_{obs} = T(D_{obs})$
  5. Choose a significance level, e.g. $\alpha = 0.05$
  6. Compute the significance value (one sided test assumed here, large values of T less probable under H_0) and check if $p < \alpha$

     $$p = P(T(D) \geq T_{obs}|H_0) = \int_{T_{obs}}^{\infty} f(X)dX.$$

- If no parametric assumption can be made, approximate the null distribution from a properly randomized/permuted data set.

Statistical testing of alignments

- For ungapped alignments, the score of a match to a random sequence is the sum of many similar random variables.
- The central limit theorem suggests that such a quantity can be well approximated by a normal distribution.
- The asymptotic distribution of the maximum M_N of N independent normal random variables is

  $$P(M_N \leq x) = P(M \leq x)^N \simeq e^{-KNe^{\lambda(x-\mu)}},$$

  where $P(M \leq x)$ is the distribution of a single random variable and K and λ are constants.
- This is called the extreme value distribution (EVD).
- The above holds true even if the individual scores are not normally distributed.

- EVD can be theoretically motivated and derived for the ungapped alignment scores (details skipped).
- Empirical evidence suggest that EVD can be used for other (gapped) models as well.
- From less pragmatic point of view, the parameters of the EVD, K and λ, can be estimated from a large collection of aligned random (non-related) sequences and the obtained empirical distribution $\hat{P}(X \geq S)$ can be used for hypothesis testing.
- If a database consists of sequences that have different lengths, then the best local alignment scores for longer sequences are, on average, higher than the best scores for than shorter sequences. Adjust scores for varying length.

References

- R. Durbin, S. R. Eddy, A. Krogh and G. Mitchison (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press.

- H. Ji and W. H. Wong (2006). Computational biology: toward deciphering gene regulatory information in mammalian genomes, *Biometrics*, vol. 62, pp. 645–663.