Local Multiple Sequence Alignment

Stephen F. Altschul

National Center for Biotechnology Information
National Library of Medicine
National Institutes of Health
Multiple Alignment Substitution Scores

a) Sum-of-the-pairs or SP-scores

4 matches; 6 mismatches


Multiple Alignment Substitution Scores

b) Tree scores

Multiple Alignment Substitution Scores

c) Star or consensus scores

3 matches; 2 mismatches
Multiple Alignment Substitution Scores

d) Entropy-based scores

\[
\begin{array}{c}
A \\
A \\
A \\
C \\
C \\
\end{array}
\]

\[
\log(4) - 0.6 \log(0.6) - 0.4 \log(0.4) = 1.03 \text{ bits}
\]

Multiple Alignment Substitution Scores

e) Log-odds scores

\[ S(\vec{x}) = \log \frac{Q(\vec{x})}{P(\vec{x})} \]

"Bayesian Integral Log-odds" or "BILD" scores

The construction of column scores from Dirichlet mixture priors

\[
Q(\vec{x}) = \sum_{i=1}^{M} m_i \frac{\Gamma(\alpha_i)}{\Gamma(\alpha_i + c)} \prod_{j=1}^{20} \frac{\Gamma(\alpha_{i,j} + c_j)}{\Gamma(\alpha_{i,j})} \\
P(\vec{x}) = \prod_{i} p_{x_i}
\]

where \( \vec{c} \) is the amino acid count vector implied by \( \vec{x} \)

Assuming uniform Dirichlet priors,

\[
S("AAACC") = \log(1.83) = 0.87 \text{ bits} \\
S("AAACT") = \log(0.91) = -0.13 \text{ bits}
\]

Multiple Alignment Gap Scores

Gap scores should, in general, be defined consistently with substitution scores.

For example, if “SP” substitution scores are used, gap scores should also be defined as the sum of gap scores for the implied pairwise alignments.

Following this prescription completely rigorously for affine gap scores entails unacceptable algorithmic complications, which can be avoided by a slight modification of one’s definition of gap score.

Local Multiple Alignment: The Problem

Neither pattern nor locations known
Local Multiple Alignment

**Desiderata for an ideal local multiple alignment algorithm:**

- Employs an appropriate measure of alignment quality
- Measure can reflect known amino acid relationships (proteins)
- Width of pattern not unduly constrained
- Width need not be specified *a priori*
- Pattern may be missing or present in multiple copies
- Alignment of segments may contain gaps
- Output is independent of order of input sequences
- Algorithm can find multiple distinct patterns
- Algorithm is deterministic
- Algorithm is rigorous optimization procedure
- Time complexity is linear in number of sequences
Approaches to Local Multiple Alignment

Consensus Word Methods
Template Methods
Progressive Alignment Methods
Pairwise Consistency Methods
Statistical Methods

Notation

- Number of sequences: $N$
- Average length of sequences: $L$
- Size of alphabet: $A$
- Specified width of pattern: $W$

We will consider only approaches that do not allow gaps
Consensus Word Methods

Find a **consensus word** that is “close” to a word in each, or a large number of the input sequences. This may be thought of as finding an optimal star alignment.

**Algorithmic outline**

For each word of fixed width $W$, define a *neighborhood of* $B$ adjacent words, each with an associated score.  

*Example:* For DNA, words could be 8-tuples, and the neighborhood of $X$ all those words with no more than 2 mismatches with $X$.

Given each sequence $S$, for each word $X$ in $S$, update the “best match to $S$” for all neighbors of $X$.

**Time complexity:** $N(LB + A^W)$.  **Space complexity:** $A^W$.

**Advantages**

For fixed word and neighborhood size, time complexity is linear in input data. Given definition of problem, algorithm is rigorous.

**Disadvantages**

Pattern length predefined. Fairly severe restrictions on pattern length and neighborhood size. No scores for words outside neighborhood.

**Verdict:** May be OK for some DNA applications; of very little use for proteins.

Template Methods

Search for a set of templates within each input sequence.

Algorithmic outline

Define a set of templates of total size $B$.

Example: For protein sequence comparison, a template could be “V*C**D”, where ‘*’ is a wild card.

Compare each template to all input sequences, updating a score for the template whenever a match is found.


Comment

This is basically an inversion of the consensus word methods, but with processing done one template at a time, rather than one sequence at a time.

Advantages and Disadvantages

Essentially the same as those for the consensus word methods.

Verdict: Probably somewhat faster and more flexible than consensus word methods, but with similar major limitations for protein comparison.

Progressive Alignment Methods

Build up local multiple alignments of fixed width in a progressive manner.

**Algorithmic outline**

Select a fixed pattern width $W$. Compare all segments of this width in the first sequence to all such segments in the second, using an arbitrary scoring system. Retain the $B$ best pairs. Compare these to all segments in the third sequence, etc.

**Variations**

Retain the best multiple alignment for each segment from the first sequence.

**Time complexity:** $NLBW$

**Advantages**

No significant restriction on pattern width. For fixed $B$ and $W$, linear time in length of input data. Can use arbitrary score function.

**Disadvantages**

Heuristic: optimal solution not guaranteed. Dependent on sequence order. Parameter $B$ may need to be very large to yield good results.


Pairwise Consistency Methods

Compare all pairs of sequences. Seek consistency among aligned letters, or diagonals.

**Algorithmic outline (Schuler et al.)**

Execute ungapped Smith-Waterman algorithm on all pairs of sequences. Mark all *diagonals* containing segment pairs that exceed a threshold score \( H \). Build up “high-dimensional” diagonals, all (or almost) all of whose pairwise projections have been marked. Search any such high-dimensional diagonals for high-scoring ungapped local multiple alignments.

**Time complexity:** \( N^2 L^2 + f(H) \)

**Advantages**

- No predefined pattern width required. Can find multiple distinct patterns.
- Can use arbitrary scoring system. Need not include all sequences. Rigorous optimization procedure, given constraint on pairwise projection scores.

**Disadvantages**

- Quadratic time in input length. Space and time complexity balloon for small \( H \).

**Comment**

- Good for a moderate number of sequences. Implemented in interactive “MACAW” program.


Local multiple sequence alignment can be viewed as an optimization problem in a rough, high-dimensional space.

One may approach this classic problem with the deterministic \textit{expectation-maximization (EM)} method (Dempster, et al., 1977).

Alternatively, one may apply one of the related stochastic methods of \textit{simulated annealing} (Metropolis, et al., 1953) or \textit{Gibbs sampling} (Geman & Geman, 1984).

Applied to local multiple sequence alignment, these approaches alternate between refining a provisional pattern, based upon its assumed locations within the sequences, and updating these locations, given the pattern.

\begin{flushleft}
\end{flushleft}