Chapter 3
AntiClustAl: Multiple Sequence Alignment by Antipole Clustering

Cinzia Di Pietro, Alfredo Ferro, Giuseppe Pigola, Alfredo Pulvirenti, Michele Purrello, Marco Ragusa, and Dennis Shasha

Summary

In this chapter, we present a new multiple sequence alignment algorithm called AntiClustAl. The method makes use of the commonly used idea of aligning homologous sequences belonging to classes generated by some clustering algorithm and then continuing the alignment process in a bottom-up way along a suitable tree structure. The final result is then read at the root of the tree. Multiple sequence alignment in each cluster makes use of progressive alignment with the 1-median (center) of the cluster. The 1-median of set S of sequences is the element of S that minimizes the average distance from any other sequence in S. Its exact computation requires quadratic time. The basic idea of our proposed algorithm is to make use of a simple and natural algorithmic technique based on randomized tournaments, an idea that has been successfully applied to large-size search problems in general metric spaces. In particular, a clustering data structure called antipole tree and an approximate linear 1-median computation are used. Our algorithm enjoys a better running time with equivalent alignment quality compared with ClustalW, a widely used tool for multiple sequence alignment. A successful biological application showing high amino acid conservation during evolution of *Xenopus laevis* SOD2 is illustrated.

3.1 Introduction

Multiple sequence alignment is the process of taking three or more input sequences and forcing them to have the same length by inserting a universal

gap symbol $-$ in order to maximize their similarity as measured by a scoring function. In the case of biological sequences (DNA, RNA, protein), the resulting aligned sequences can be used for two purposes: first, to find regions of similarity defining a conserved consensus pattern of characters (nucleotides or amino acids) in all the sequences; second, if the alignment is particularly strong, to use the aligned positions to infer some possible evolutionary relationships among the sequences.

Formally, the problem is the follows: let Σ be an alphabet and $\mathcal{S} = \{S_1, \ldots, S_k\}$ be a set of string defined over Σ. A *multiple sequence alignment* of $\mathcal{S}$ is a set $\mathcal{S}' = \{S'_1, \ldots, S'_k\}$ such that

- $S'_i \in (\Sigma \cup \{-\})^*$ for each $i = 1, \ldots, k$
- S_i is obtained from S'_i by dropping all gap symbols $\{-\}$
- $|S'_1| = |S'_2| = \ldots = |S'_k|$

A *scoring function* defined on the alphabet Σ is a map $\sigma : (\Sigma \cup \{-\})^k \mapsto R$. It has the following properties:

1. Reflexivity (maximum score if all the same) $\sigma(a, ..., a) \geq \sigma(a_1, ..., a_k)$, provided $a \neq -$.
2. Symmetry (it doesn't matter where differences are found, so the score is based on the evaluation of the multiset of characters in the argument):
   $\sigma(x_1, ..., x_i, a, x_{i+2}, ..., x_j, b, x_{j+2}, ..., x_k)$
   $= \sigma(x_1, ..., x_i, b, x_{i+2}, ..., x_j, a, x_{j+2}, ..., x_k)$
3. Triangle inequality (recall that similarity is the opposite of distance):
   $\sigma(x_1, ..., x_i, a, x_{i+2}, ..., x_j, b, x_{j+2}, ..., x_k)$
   $+\sigma(x_1, ..., x_i, b, x_{i+2}, ..., x_j, c, x_{j+2}, ..., x_k)$
   $\geq \sigma(x_1, ..., x_i, a, x_{i+2}, ..., x_j, c, x_{j+2}, ..., x_k)$

The best score $D(|S_1|, |S_2|, \ldots, |S_k|)$ for aligning k sequences $S_1, S_2, \ldots, S_k$ with respect to σ is the one that maximizes the sum of the σs across all positions: $\sum_{i \in 1..n} \sigma(S'_1[i], S'_2[i] \ldots S'_k[i])$. If $|S_1| = |S_2| = |S_k| = n$, then the space and the time complexity of the best currently known algorithm is $\mathcal{O}(n^k)$ and $\mathcal{O}(2^k n^k) \times \mathcal{O}$(computation of the σ function), respectively. Finding the optimal solution of the multiple sequence alignment therefore requires exponential space and time complexity. If only pairwise alignment is considered, then an $\mathrm{O}(n^2/\log n))$ algorithm can be obtained [88].

The most successful solution to the problem has been provided by the program ClustalW [177]. In this chapter, we propose a new solution based on a top-down "bisector tree" [70] clustering algorithm called *antipole tree* and a linear approximate 1-median computation. Since exact 1-median computation requires a quadratic number of distance calculations and given that each of such distance computations may require quadratic time in the length of the biosequences, the use of a linear approximate 1-median computation may give a much better running time.

Both clustering and approximate 1-median computation algorithms make use of a very simple and natural technique based on randomized tournaments.