Universität Bielefeld
Graduiertenkolleg Strukturbildungs–prozesse

Divide-and-Conquer Multiple Sequence Alignment

J. Stoye, S. W. Perrey, A. W. M. Dress

Basic Principle

We present a fast heuristic algorithm for multiple sequence alignment [1,2]: *Divide-and-Conquer Multiple Sequence Alignment* (DCA).

Figure 1 shows the method in a schematic way: The sequences are divided at appropriate positions near to their midpoint. This way, the problem of aligning one family of sequences is reduced to that of aligning two sequence families, each of sequences of approximately only half the length. After reiterating this division procedure sufficiently often, the subsequences are sufficiently short, say, shorter than a threshold L, and can be aligned optimally.

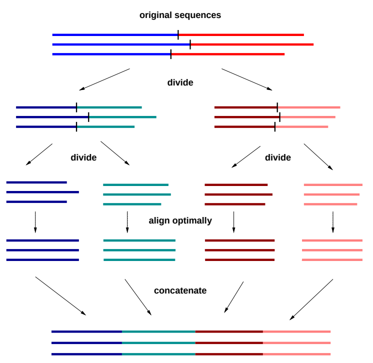

Figure 1

Computing Slicing Positions

The main question arising with this method is how to find "appropriate" slicing positions.

For simplicity, we explain our solution for the case of a family of three sequences $\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3$. The generalization to more than three sequences is straightforward.

First note that for every division point c_1 of $\mathbf{s}_1$, there exist corresponding positions c_2 and c_3 of $\mathbf{s}_2$ and $\mathbf{s}_3$, respectively, so that an optimal alignment results from concatenating optimal alignments of the three prefix and the three suffix sequences defined by the slicing positions c_1, c_2, c_3. We take advantage of this fact by first dividing $\mathbf{s}_1$ at its midpoint $\hat{c}_1 := \lceil |\mathbf{s}_1|/2 \rceil$ and then searching for compatible positions in the other sequences.

To estimate the compatibility of slicing positions c, d of sequences $\mathbf{s}$, $\mathbf{t}$, we define the *additional cost* $C_{\mathbf{s},\mathbf{t}}[c,d]$ imposed by first slicing $\mathbf{s}$ and $\mathbf{t}$ at c and d, respectively, and then aligning $\mathbf{s}$ and $\mathbf{t}$ by concatenating the optimal alignments of the resulting prefix and suffix sequences by

$$C_{\mathbf{s},\mathbf{t}}[c,d] := w_{opt}(\text{prefix}) + w_{opt}(\text{suffix}) - w_{opt}(\text{total})$$

where w_{opt} denotes an appropriate measure of fit of the three optimal alignments in question (see Figure 2 for an example).

To obtain an estimate for the additional cost imposed by a particular choice c_1, c_2, c_3 of slicing positions, we take a (weighted) sum of pairwise additional costs:

$$\begin{aligned} C(c_1, c_2, c_3) \quad := \quad & \alpha_{1,2}\, C_{\mathbf{s}_1,\mathbf{s}_2}[c_1, c_2] \\ & + \alpha_{1,3}\, C_{\mathbf{s}_1,\mathbf{s}_3}[c_1, c_3] \\ & + \alpha_{2,3}\, C_{\mathbf{s}_2,\mathbf{s}_3}[c_2, c_3] \end{aligned}$$

where $\alpha_{1,2}$, $\alpha_{1,3}$, $\alpha_{2,3}$ are appropriately chosen weight factors reflecting e.g. phylogenetic relationships.

We implemented several heuristics which allow to speed-up the search for slicing positions minimizing this value which work well for up to a dozen and even more sequences.

For the optimal alignment of the finally resulting short subsequences, we use the program MSA [3,4].

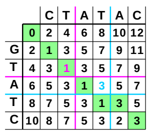

Figure 2a

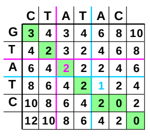

Figure 2b

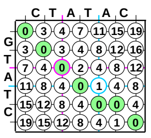

Figure 2c

Figures 2a and 2b show the standard dynamic programming distance matrices of the sequences **s** = GTATC and **t** = CTATAC (using unit cost and penalty +2 for each single insertion/deletion), computed from the upper left to the lower right (Fig. 2a) and from the lower right to the upper left (Fig. 2b). Figure 2c displays the *additional−cost* matrix, containing the values $C_{\mathbf{s},\mathbf{t}}[c,d]$ for each pair of slicing positions (c,d), e.g.

$C_{\mathbf{s},\mathbf{t}}[2,2] = w_{opt}[\text{CT,GT}] + w_{opt}[\text{ATAC,ATC}] - w_{opt}[\text{CTATAC,GTATC}] = 1 + 2 - 3 = 0$

$C_{\mathbf{s},\mathbf{t}}[4,3] = w_{opt}[\text{CTAT,GTA}] + w_{opt}[\text{AC,TC}] - w_{opt}[\text{CTATAC,GTATC}] = 3 + 1 - 3 = 1$

In each matrix, the optimal alignment path is colored green. Its additional−cost matrix entries are, of course, zero.

Algorithm

$$\begin{aligned} & DCA(\langle \mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3 \rangle, L) \\ & \quad = MSA(\langle \mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3 \rangle), \text{ if } \min\{|\mathbf{s}_1|, |\mathbf{s}_2|, |\mathbf{s}_3|\} \le L \\ & \quad = DCA(\langle \mathbf{s}_1(\le \hat{c}_1), \mathbf{s}_2(\le c_2), \mathbf{s}_3(\le c_3) \rangle, L) \\ & \qquad + DCA(\langle \mathbf{s}_1(> \hat{c}_1), \mathbf{s}_2(> c_2), \mathbf{s}_3(> c_3) \rangle, L) \\ & \qquad \text{where } \hat{c}_1 := \lceil |\mathbf{s}_1|/2 \rceil \\ & \qquad\qquad (c_2, c_3) \in \{0, \ldots, |\mathbf{s}_2|\} \times \{0, \ldots, |\mathbf{s}_3|\} \\ & \qquad\qquad\quad \text{such that } C(\hat{c}_1, c_2, c_3) \text{ is minimal.} \end{aligned}$$

Practicability of DCA

Our alignment algorithm has been subjected to a considerable number of test cases. The following measurements are made for simulated data (randomly generated amino acid sequences with 20 – 30 percent sequence identity).

Figures 4a and 4b show the expected trade-off between alignment quality (the relative deviation from the optimal alignment score) and computation time depending on the threshold L. A value of $L = 40$ seems to be a good compromise.

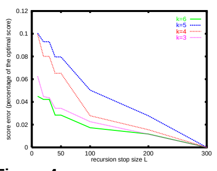

Figure 4a

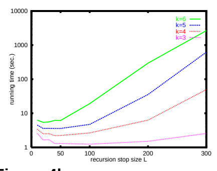

Figure 4b

We evaluated running times (Figures 5a,b) and memory usage (Figures 6a,b) for different families of k sequences depending on the average sequence length.

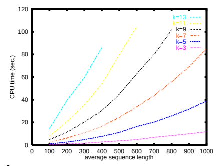

Figure 5a

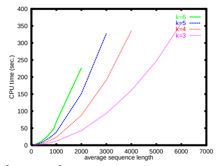

Figure 5b

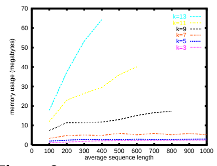

Figure 6a

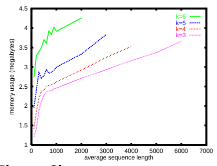

Figure 6b

Aligning Biological Sequences

We also applied our procedure to several biological sequence families.

Table 1 shows our results for the globin, kinase, aspartic acid protease, and ribonuclease H protein families from McClure *et al.* [5] compared to the results of the best and second-best alignment programs considered there. The score values denote numbers of correctly aligned motifs.

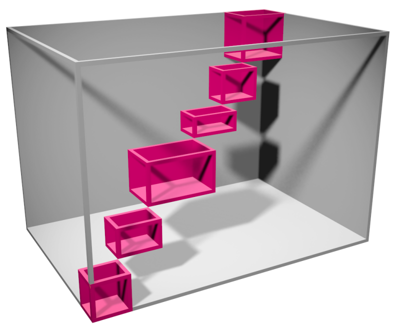

Figure 3: The reduction of search space.

sequences	max.	DFALIGN	AMULT	DCA
Globins 6	5.00	5.00	5.00	5.00
Globins 10	5.00	5.00	5.00	5.00
Globins 12	5.00	5.00	5.00	5.00
Kinases 6	8.00	7.67	7.33	8.00
Kinases 10	8.00	8.00	7.70	8.00
Kinases 12	8.00	8.00	7.75	8.00
Proteases 6	3.00	2.33	1.17	*2.50
Proteases 10	3.00	*3.00	*2.40	*2.50
Proteases 12	3.00	*3.00	2.33	*2.50
RH 6	4.00	3.67	*3.30	*3.83
RH 10	4.00	3.30	*3.20	*3.70
RH 12	4.00	3.83	*2.92	3.42

Table 1

In Table 2, running times and alignment quality are compared to the corresponding values of score-optimal alignments computed with the program MSA as published by Gupta *et al.* [4].

sequences	max.	MSA (PAM250)		DCA (PAM250)		DCA (Blosum)	
Globins A (7)	5.00	4.86	157 sec	4.86	4.3 sec	5.00	5.3 sec
Globins B (10)	5.00	5.00	130 sec	4.90	10.4 sec	5.00	10.9 sec
Kinases A (5)	8.00	8.00	10 min	8.00	10.6 sec	8.00	16.8 sec
Kinases B (6)	8.00	8.00	118 min	8.00	9.7 sec	8.00	57.5 sec
Kinases C (4)	8.00	6.75	210 sec	*7.50	4.6 sec	7.25	4.8 sec
Proteases A (5)	3.00	2.80	37 sec	2.40	2.8 sec	2.80	18.3 sec
Proteases B (4)	3.00	0.50	9 min	0.00	1.4 sec	1.00	3.3 sec
RH A (5)	4.00	2.60	68 min	*2.60	3.4 sec	3.40	29.0 sec

Table 2

Range of Applications

- Simultaneous alignment of up to a dozen sequences of length that of an average protein.
- Systematic evaluation of score functions for multiple sequence alignments.
- Rapid three- or four-way alignments, e.g. for the use in programs which simultaneously compute an alignment and reconstruct a phylogenetic tree.
- Speed-up of fragment-based multiple alignment methods.

Acknowledgement

The authors thank Peter Serocka for his help with this poster.

References

[1] U. Tönges, S. W. Perrey, J. Stoye, and A. W. M. Dress. A General Method for Fast Multiple Sequence Alignment. *Gene*, 172:GC33–GC41, 1996.

[2] J. Stoye, S. W. Perrey, and A. W. M. Dress. Improving the Divide-and-Conquer Approach to Sum-of-Pairs Multiple Sequence Alignment. *Appl. Math. Lett.*, To appear.

[3] D. J. Lipman, S. F. Altschul, and J. D. Kececioglu. A Tool for Multiple Sequence Alignment. *Proc. Natl. Acad. Sci. USA*, 86:4412–4415, 1989.

[4] S. K. Gupta, J. D. Kececioglu, and A. A. Schäffer. Improving the Practical Space and Time Efficiency of the Shortest-Paths Approach to Sum-of-Pairs Multiple Sequence Alignment. *J. Comp. Biol.*, 2(3):459–472, 1995.

[5] M. A. McClure, T. K. Vasi, and W. M. Fitch. Comparative Analysis of Multiple Protein-Sequence Alignment Methods. *Mol. Biol. Evol.*, 11(4):571–592, 1994.