ANCHOR-BASED ALIGNMENT METHOD FOR THE SEQUENCE VS. SEQUENCE AND PROFILE VS. SEQUENCE ALIGNMENT

Sunyaev Sh.R.[1,2], Bogopolsky G.A.[1], Oleynikova N.V.[3,4], Vlasov P.K.[1,3], Finkelstein A.V.[5], Roytberg M.A.[4*]

[1] Institute of Molecular Biology, RAS, Moscow, Russia
[2] European Molecular Biology Laboratory (EMBL), Germany
[3] Moscow Institute of Physics and Technology, Dolgoprudny, Moscow Region, Russia
[4] Institute of Mathematical Problems in Biology, RAS, Pushchino, Moscow Region, Russia
[5] Institute of Protein Research, RAS, Pushchino, Moscow Region, Russia
e-mail: roytberg@impb.psn.ru
[*]Corresponding author

Key words: *profile, position-dependent scoring matrix (PDSM), alignment*

Resume

Motivation: Sequence vs. sequence and sequence vs. profile alignments are important methods to attribute the protein sequence to the corresponding protein family. However, the accuracy and efficiency of existent methods do not meet the needs of the contemporary genomic and proteomic studies.

Results: The anchor-based method to align protein sequence with another protein sequence or position dependent scoring matrixes (PDSM or profile) was proposed. It has been shown that the method is approximately as accurate as Smith-Waterman method, but considerably faster.

Introduction

Alignment of two protein sequences is old and probably the most classic problem in computational biology. It is a key step in database search, in computational methods for prediction of protein function and homology-based modelling of 3D protein structure. Many sophisticated computational methods in molecular biology, like multiple alignments, profile analysis, threading etc. use pair-wise sequence alignment as a sub-procedure. Smith-Waterman method (Smith, Waterman, 1981) is currently the most sensitive one for alignment, but the slowest. Faster algorithms such as BLAST, FASTA (Altschul et al., 1990, Pearson, 1996) have a tendency to some loss of accuracy.

Using position dependent scoring matrixes (PDSM or profile) usually improves the accuracy and sensibility of alignment. Profile allows to find more distant relevant homologues, because it contains information about multiple or/and structure alignment (Altschul et al., 1990, Eddy, 1998, Sunyaev, 1999).

It is known (Vogt, 1995) that alignments of proteins of low or medium percent of identity (say, 10-30%) obtained by Smith-Waterman method usually differ from those obtained from the 3D-structures alignment. Moreover, our careful comparison between the algorithmic sequence alignments and the structural ones showed that fragments of the Smith-Waterman alignments with pour similarity usually have nothing to do with the structural (and thus more reliable) alignment.

This suggests to ignore (at least for beginning) all elements of the Neadleman-Wunsch matrix, except the “anchors”, i.e. ungapped fragments of (relatively) high similarity.

The idea to start the alignment procedure from the search for such anchors is definitely not new and was implemented in several software tools (e.g. mentioned above BLAST and FASTA). However, this idea was considered as a way to increase computational speed of alignment techniques inevitably associated with the loss of alignment accuracy. Our observations suggest the way to improve computational speed without sacrificing (and even with a slight gain of) alignment accuracy and confidence compared to the Smith-Waterman algorithm. The technique is adopted both for sequence vs. sequence and for sequence vs. profile alignments.

Methods

The proposed alignment algorithm works up given sequences in three steps:

Step 1. We generate a set of ungapped high-scoring segments ("*anchors*"), marked as shadow cells on Fig. a.

Anchor is an ungapped matching of equal-length fragments, {U[a, a+L] vs. V[b, b+L]}, of sequences U and V. These fragments meet the following conditions:

a) anchor contains at least one “seed pair” {U[x, x+1] vs. V[y, y+1]} with the score exceeding a cutoff CSeed;

b) the anchor’s score (i.e., the sum of the substitution scores M(U[x], V[y]) over the anchor) exceeds a cutoff CAnchor;

c) the score of any continuous part of the anchor exceeds a cutoff CMin;

d) the anchor is "locally maximal", i.e., (i) it is not a part of any other pair of segments {U[a', a'+L'] vs. V[b', b'+L']} meeting conditions a) – c) and having greater or equal score, and (ii) it does not include any continuous part having a greater score.

Step 1. starts with identifying seed pairs and then expands them to obtain the anchors. This step is similar to the procedures used in BLAST and FASTA and is the most time-consuming step of our algorithm.

Step 2. We find the optimal *block alignment* path through the set of anchors (marked by thick line). Block is a continuous part of anchor. Block alignment is a chain of the blocks $\{B_1, \ldots, B_N\}$, where B_i precedes B_{i+1} both along U and V sequences. The block alignment $\{B_1, \ldots, B_N\}$ is optimal if it has maximal possible *block score,* which is defined as follows:

$$\text{Score}(B_1, \ldots, B_N) = \text{Score}(B_1) - \text{Link}(B_1,B_2) + \text{Score}(B_2) - \ldots - \text{Link}(B_{N-1},B_N) + \text{Score}(B_N). \quad (1)$$

$\text{Score}(B_i)$ is the total score of matches along block B_i according to the given substitution matrix M. $\text{Link}(B_i, B_{i+1}) = \alpha + \beta \bullet |(y-x) - (y'-x')|$ is the linkage penalty for the blocks B_i and B_{i+1}, where α (*linkage open penalty)* and β (*linkage elongation penalty)* are analogs of the traditional gap opening and gap elongation penalties, while x, y are the last residues of block B_i, and x', y' are the first residues of block B_{i+1} in sequences U and V, respectively. Note that we penalize links between the blocks even if the blocks are placed on the same diagonal.

To find the optimal block alignment from the created set of anchors we use either the Wilbur-Lipman algorithm (Wilbur, Lipman, 1983) (if the number of the initial anchors K is small), or (if K is large) the sparse dynamic programming (SDP) method (Eppstein et al., 1992). These procedures produce the same alignments (given the same parameters and set of anchors), but differ in the run-time: the Wilbur-Lippman algorithm run-time is proportional to K^2, while the SPD run-time is of order K*log(L), where L is the length of the shorter sequence. The first procedure performs faster (and therefore is used to find the optimal block alignment) if $K < 20$; otherwise the second procedure is evoked.

Step 3. We specify the alignment path in regions between the blocks. To this end we use a global version of the Smith-Waterman algorithm. Our experiments show that usually this step comprises only a small part of the total run-time of our algorithm.

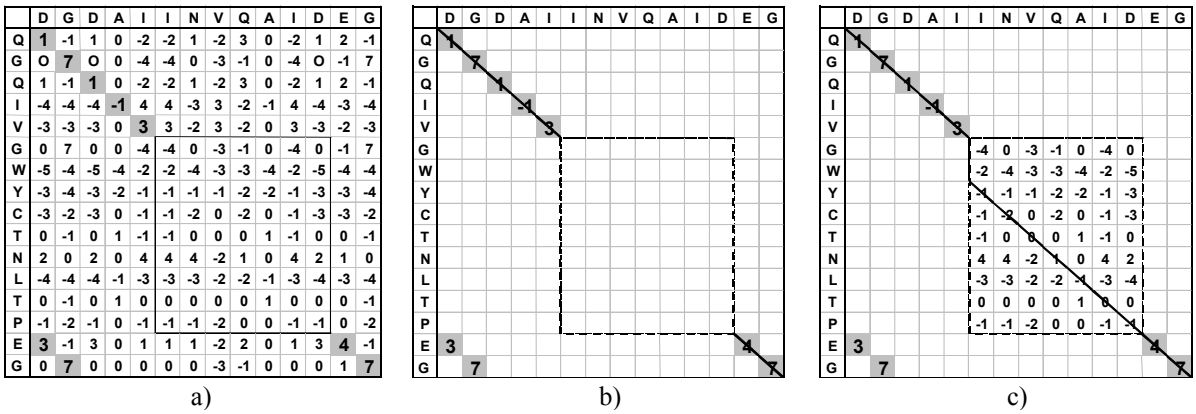

Fig. Three steps of new alignment algorithm: (a) finding of anchors set; (b) obtaining of the optimal block alignment path through the set of anchors; (c) improving the alignment between anchors by a global version of the Smith-Waterman algorithm.

The main difference between implementations of the method for the sequence vs. sequence and sequence vs. profile alignment is in creation of anchors (step 1). In both cases we start with the generation of all possible seed pairs with the similarity score exceeding a cutoff Cseed. To do this we scan the first sequence or the profile U. For each position *i* of U we create the list of all k-tuples (usually, k = 2) having significant (> CSeed) similarity with the k consequent positions of U, starting in *i*-th position. The run-time for this step is proportional to the length of U, but if U is a profile, the multiplicative constant is larger. This is the most time consuming step of the algorithm. Fortunately, in case of the database search this step is performed only once per the whole of the database and therefore its run-time is not crucial for the effectiveness of the database search. When the lists of the potential homologues of the k-tuples are created, the standard FASTA-like technique to generate the anchors can be implemented.

Results and Discussion

Accuracy and confidence of the method have been tested through comparison with 583 standard alignments extracted from BaliBase databases (Thompson, Plewniak, Poch. Bioinformatics, 1999, 15, 87-88). For each pair of protein sequences the golden standard alignments have been compared to the alignments constructed by Smith-Waterman algorithm with standard settings. Percentage of amino acid pairs correctly aligned by an algorithm with respect to the golden standard alignment was

used as a measure of the algorithmic alignment ***accuracy*** and percentage of amino acid pairs correctly aligned by an algorithm with respect to the algorithm alignment was used as a measure of the algorithmic alignment ***confidence***.

For testing profile method we use the following technique: from the database of multiple alignment we delete one sequence, produce profile [PSIC] and align the deleted sequence with obtained profile.

Results of these tests show that the novel method slightly outperforms the SW algorithm both in accuracy and confidence. As has been stated above, focusing on high-scoring regions can significantly improve the computational speed of the algorithm. The current software has been compared with the standard (search at http://biobase.dk/programs/Ordered_by_Funtionality/Multifunctional_Programme_Pack/PEARSON/Pearson_Programme_List/pearson_programme_list.html) and in house implementations of the SW algorithm. Table shows that the suggested method requires about 1.5 times shorter computational time than the classic SW technique. Origins of these issues discussed in the Methods section.

Table. Presents the data on the characteristics of the algorithm for align protein sequences.

Family	Num	Len	%ID	KOP_time	SW_Time	SW_Acc%	SW_Conf%	KOP_Acc%	KOP_Conf%
1idy	10	54	12.7	9	16	15.66	28.96	13.01	28.88
1tvxA	21	58	21.1	12	19	24.92	40.61	25.12	41.26
1aboA	120	59	24.9	8	20	65.02	72.07	64.44	71.21
1tgxA	105	63	37.0	16	23	61.86	68.37	63.21	67.90
1r69	10	70	15.5	15	28	23.84	33.98	23.84	30.35
1ubi	6	85	24.2	22	42	28.15	51.36	32.56	52.56
1csy	55	85	30.8	19	41	67.23	69.93	68.79	70.35
2trx	6	92	18.8	27	47	27.30	39.23	27.30	39.23
1wit	10	97	17.4	25	55	43.85	66.44	43.14	56.66
1uky	6	203	15.0	152	235	29.63	35.97	33.55	35.44
1havA	171	211	29.9	130	249	58.41	69.99	60.31	66.41
2hsdA	6	245	18.8	209	344	39.20	43.00	36.44	39.42
1sbp	15	245	16.1	210	341	13.71	14.47	16.62	17.28
2pia	6	257	13.2	202	379	47.47	60.49	47.87	58.51
kinase	6	270	24.0	276	412	71.00	75.35	73.48	78.01
1ped	3	350	24.7	684	704	53.89	65.25	52.68	62.61
4enl	3	364	20.3	738	768	28.74	30.03	31.3	32.26
1ajsA	6	371	13.7	724	813	15.41	27.14	25.32	28.78
1cpt	6	398	20.8	853	946	61.90	70.47	62.39	67.15
2myr	6	407	15.8	1082	997	20.60	21.66	22.20	23.00
1pamA	6	470	20.5	1499	1377	45.03	57.89	50.55	53.08
All	***583***	***138***	***27.6***	***111***	***162***	***55.07***	***63.79***	***56.20***	***62.30***

Data for logarithmic data set of parameters for pare-wise alignments: 15 (opening) and 1 (extension) deletion penalties for Smith-Waterman algorithm; CAnchor = linkage open penalty = 17 and linkage elongation penalty = 1; substitution scoring matrix = Gon250. For profile vs. sequence alignments data is the same.

Notation: family – protein's family name in Bali-base; Num – number of performed alignments; Len – average length of sequences in the family; %ID – average percent of identity; KOP_time and SW_time – time need to make new and SW alignment respectively; _Acc% and _Conf% - average percent of accuracy and confidence.

Acknowledgements

This work was supported by the INTAS grant 99-01476, by the Netherlands Organization for Scientific Research (NWO) grant, by 00-04-48246, 01-04-48400, and 01-01-00287 RBRF grants, 13/hg grant from Russian State programme Human Genome, partly by French-Russian Lyapunov Centre and by an International Research Scholar's Award to A.V.F. from the Howard Hughes Medical Institute.

References

1. Altschul S.F., Madden T.L., Schäffer A.A., Zhang J., Zhang Z., Miller W., Lipman D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucl. Acids Res. 25:3389-3402
2. Eddy S.R. (1998) Profile hidden Markov models. Bioinformatics. 14, 755-763.

3. Eppstein D., Galil Z., Giancarlo R., Italiano G.F. (1992) Sparse dynamic programming. 1. Linear cost-functions. J. of the ACM. 39, 519-545.
4. Pearson W.R. (1996) Effective protein sequence comparison. In Meth. Enz., R.F.Doolittle, ed. (San Diego: Academic Press). 266:227-258
5. Smith T.F., Waterman M.S. (1981) Identification of common molecular subsequences. J. Mol. Biol. 147, 195-197.
6. Sunyaev S.R., Eisenhaber F., Rodchenkov I.V., Eisenhaber B., Tumanyan V.G., Kuznetsov E.N. (1999) PSIC: profile extraction from sequence alignments with position-specific counts of independent observations. Protein Eng. 12(5):387-394.
7. Vogt G., Etzold T., Argos P. (1995) An assessment of amino acid exchange matrices in aligning protein sequences: the twilight zone revisited. J. Mol. Biol. 249, 816-831.
8. Wilbur W.J., Lipman D.J. (1983) Proc. Natl Acad. Sci. USA. Rapid similarity searches of nucleic acid and protein data banks. 80, 726-730.