BIOINFORMATICS

Vol. 19 no. 9 2003, pages 1155–1161
DOI: 10.1093/bioinformatics/btg133

RASCAL: rapid scanning and correction of multiple sequence alignments

*J. D. Thompson, J. C. Thierry and O. Poch**

Laboratoire de Biologie et Génomique Structurales, Institut de Génétique et de Biologie Moléculaire et Cellulaire (CNRS/INSERM/ULP), B.P. 10142, 67404 Illkirch Cedex, France

Received on October 16, 2002; revised on December 16, 2002; accepted on January 13, 2003

ABSTRACT

Motivation: Most multiple sequence alignment programs use heuristics that sometimes introduce errors into the alignment. The most commonly used methods to correct these errors use iterative techniques to maximize an objective function. We present here an alternative, knowledge-based approach that combines a number of recently developed methods into a two-step refinement process. The alignment is divided horizontally and vertically to form a 'lattice' in which well aligned regions can be differentiated. Alignment correction is then restricted to the less reliable regions, leading to a more reliable and efficient refinement strategy.

Results: The accuracy and reliability of RASCAL is demonstrated using: (i) alignments from the BAliBASE benchmark database, where significant improvements were often observed, with no deterioration of the existing high-quality regions, (ii) a large scale study involving 946 alignments from the ProDom protein domain database, where alignment quality was increased in 68% of the cases; and (iii) an automatic pipeline to obtain a high-quality alignment of 695 full-length nuclear receptor proteins, which took 11 min on a DEC Alpha 6100 computer.

Availability: RASCAL is available at ftp://ftp-igbmc.u-strasbg.fr/pub/RASCAL.

Contact: poch@igbmc.u-strasbg.fr

Supplementary information: http://bioinfo-igbmc.u-strasbourg.fr/BioInfo/RASCAL/paper/rascal_supp.html

INTRODUCTION

Multiple sequence alignments of protein families are essential in many areas of molecular biology, ranging from sequence database searching, 2D/3D structure prediction and the detection of key functional residues to the study of evolutionary relationships. Recently, multiple alignments have also been incorporated in high-throughput systems such as genome annotation and analysis and protein domain/motif database management systems, where automation of the alignment construction process is essential. (For a review, see Lecompte *et al.*, 2001). The reliability and accuracy of all of these methods depends critically on the quality of the underlying multiple alignment.

Unfortunately, optimal exact alignment of more than about 20 sequences remains impractical due to the intensive computer resources required (e.g. Gupta *et al.*, 1995). In general, heuristics offer more practical solutions, but may introduce errors into the alignment. The most widely used heuristic method is the progressive alignment algorithm (Feng and Doolittle, 1987), in which a multiple alignment is built up progressively by a series of pairwise alignments. More recently, iterative strategies have been used to refine and improve the alignment (e.g. Eddy, 1995; Gotoh, 1996; Katoh *et al.*, 2002). These methods make more or less random changes to the alignment in order to maximize a global objective function (OF; Carrillo and Lipman, 1988; Thompson *et al.*, 2001). However, current systems are too time-consuming to be practical in automatic, high-throughput systems. Furthermore, there is no guarantee that the 'optimal' alignment corresponding to the maximum OF score is the same as the 'biologically correct' alignment. An alternative approach has therefore been to use a more co-operative strategy integrating different, complementary algorithms (Thompson *et al.*, 2000; Notredame *et al.*, 2000) and/or incorporating other biological information (Heringa, 1999; Jennings *et al.*, 2001). Although much progress has been made, all these strategies still have some weak points, resulting in alignments that are not always correct. The best multiple alignment programs (Thompson *et al.*, 1999a; Notredame *et al.*, 2000; Katoh *et al.*, 2002) are capable of producing alignments with about 86% accuracy in tests using BAliBASE (Thompson *et al.*, 1999b).

We present here a new program, RASCAL, which can be used to refine and improve any automatic or manually constructed multiple sequence alignment. It combines a number of recently developed methods for the

*To whom correspondence should be addressed.

Downloaded from https://academic.oup.com/bioinformatics/article/19/9/1155/285019 by guest on 24 April 2024

analysis and correction of alignments into an integrated, knowledge-based system. The alignment is initially divided horizontally by defining potential functional subfamilies and vertically to identify well-aligned, reliable regions. Potential alignment errors are detected by comparing statistical models (Gribskov *et al.*, 1987) of the reliable regions. RASCAL then performs a single realignment of each badly aligned region using an algorithm similar to that implemented in ClustalW (Thompson *et al.*, 1994).

The accuracy, reliability and efficiency of the RASCAL alignments are demonstrated using the 3D structural alignments in the BAliBASE benchmark database, a large-scale project, using a subset of 946 alignments from the ProDom protein domain database (Corpet *et al.*, 2000) and a large alignment of 695 nuclear receptor proteins constructed using the MAFFT progressive alignment program. Thus, RASCAL provides the essential, final refinement step in an automatic process to construct fast and accurate multiple alignments of complete protein sequences.

SYSTEM AND METHODS

Testing and training sets

The BAliBASE benchmark database (version 2) (http://www-igbmc.u-strasbg.fr/BioInfo/BAliBASE2/) is designed specifically for the analysis of multiple alignment problems. It consists of 142 high-quality multiple alignments based on 3D structural superpositions and organized into a number of test sets: Reference 1 contains a small number of equi-distant sequences, Reference 2 contains families aligned with a highly divergent sequence, Reference 3 contains subgroups with <25% residue identity between groups, References 4 and 5 contain sequences with large N/C-terminal extensions or internal insertions, respectively. Four different programs were used to construct alignments of each BAliBASE test case; two progressive alignment programs, ClustalW and FFT-NS2 from the MAFFT program suite (referred to here as MAFFT-2); one iterative method, FFT-NSI (referred to here as MAFFT-I), which iteratively refines an initial MAFFT-2 alignment and one co-operative method, T-COFFEE. We calculate two different scores to estimate the quality of the alignments produced. The sum-of-pairs score is the percentage of correctly aligned pairs of residues and the column score is the percentage of correctly aligned columns in the test alignment.

The ProDom database (version 2001.3) contains 305 465 protein domain alignments, constructed using MULTALIN (Corpet, 1988). Because no reference alignments are available, we use the NorMD OF (Thompson *et al.*, 2001) to evaluate the alignments. We selected a subset of 946 ProDom alignments having NorMD scores between 0 and 0.5, suggesting that the alignments may contain errors and thus represent potential test cases for refinement by RASCAL.

The final test set consists of 695 full-length nuclear receptor sequences, detected by a series of BlastP (Altschul *et al.*, 1997) searches. The sequences contained both the highly conserved DNA binding domain and the more divergent ligand binding domain. The test set thus presents many of the problems encountered when aligning large sets of complete sequences, including multi-domain proteins, large N/C terminal extensions, internal insertions and fragments. The 3D structures of nine proteins allowed an objective evaluation of the quality of the multiple alignments obtained in this study.

ALGORITHM

RASCAL uses a combination of different techniques for the analysis and refinement of multiple alignments. The procedure consists of two stages: (i) an initial alignment scanning and validation stage; and (ii) an error correction stage.

Initial alignment analysis

The initial step in RASCAL involves the localization of well aligned regions in the alignment. The alignment is divided horizontally into sequence subfamilies using Secator (Wicker *et al.*, 2001) and vertically into 'core block' regions that are reliably aligned in the majority of the sequences. These 'global' core blocks are determined using the mean distance (MD) column scores implemented in the NorMD OF. A sliding window analysis of the MD scores is performed using a window length of 8 (Fig. 1). As MD column scores are normalized in the range of 0 to 100, we can define a threshold above which columns are considered to have significant scores. Global core blocks are thus defined as regions of at least four columns with mean MD score above a threshold of eight. Local core block regions are also determined for each subfamily individually, using the same method as for the complete family. These regions represent the blocks that are well conserved within a particular subfamily, but exclude the global core blocks defined for the complete alignment.

Correction of alignment errors

The error correction stage involves a hierarchical, progressive realignment of the local regions defined in the initial analysis stage. The first step is to detect and realign alignment errors within each subfamily. Secondly, the conserved regions for each subfamily are compared to all other subfamilies and if necessary, realigned. Thirdly, the orphan sequences, i.e. the sequences not clustered by Secator, are treated separately, after the correction of the subfamilies. Finally, the global core blocks are

Downloaded from https://academic.oup.com/bioinformatics/article/19/9/1155/285019 by guest on 24 April 2024

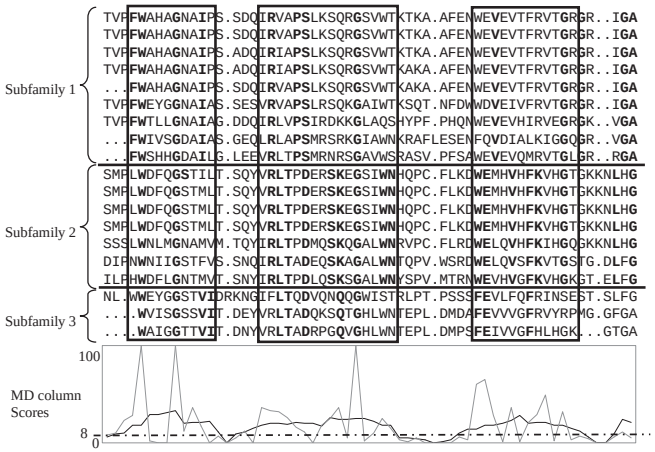

Fig. 1. Part of an alignment of 19 lectin domain sequences clustered into three subfamilies. Positions conserved within each subfamily are shown in bold and core blocks are boxed. Scores: MD column scores are grey and averaged MD scores (window length = 8) are black.

recalculated and the regions between the core blocks, as well as the N/C terminal regions, are realigned.

(i) Badly aligned sequences within subfamilies In order to detect sequences that may be locally misaligned, RASCAL considers each sequence within the context of its subfamily (Fig. 2A). For each local core block region in each subfamily, a profile (Gribskov *et al.*, 1987) is built from the alignment. For a group of N aligned sequences of length L, let a_{ij} be the amino acid in sequence i at position j. The rows of the profile correspond to positions in the alignment and the columns represent all possible amino acids. The profile score at row r and column c is given by:

$$\text{Profile}(r, c) = \sum_{i=1}^{N} W_i M(a_{ir}, a_c)$$

where M is the value in the comparison matrix for two residues, a_c is the residue represented by column c in the profile and W_i is the weight of the ith sequence (Thompson *et al.*, 1994). Each sequence within the subfamily is assigned a score S_i where:

$$S_i = \sum_{j=1}^{L} \text{Profile}(j, a_{ij})$$

A threshold score for each subfamily is then defined based on the distribution of the sequence scores:

$$\text{Threshold} = \text{Q1} - 2 * (\text{Q3} - \text{Q1})$$

where Q1 and Q3 are the lower and upper quartiles of the sequence scores, respectively. The threshold score is based on the inter-quartile range because this provides a general rule-of-thumb that does not depend on the underlying distribution of the sequence scores. Sequence 'outliers' that score below the threshold are considered as potential alignment errors.

When outliers are detected, the sequence segments for realignment are extended to the limits of the nearest high-scoring core blocks before and after the low-scoring block or set of blocks (Fig. 2B). Each sequence segment is then realigned against the remaining sequences in the subfamily using a dynamic programming algorithm similar to that used in ClustalW (Thompson *et al.*, 1994).

(ii) Badly aligned subfamily core blocks Potential alignment errors between subfamilies are detected by performing pairwise profile–profile comparisons of all profiles in each subfamily with all other subfamilies (Fig. 3). The score for aligning two profiles is defined as the sum of the scores for each pair of columns, and the calculation of the score for two columns is the same as that used in ClustalW. Let F1 and F2 be two subfamilies in the alignment. A score S_{ij} is assigned to the jth profile in the ith subfamily that represents the score for the initial alignment of the profile with the other group. For example, in Figure 3a, the initial scores, S_{12} and S_{22} for aligning profile P_{12} with profile P_{22} are both equal to the sum of the scores for columns 47 to 54 = 128: all other initial scores are zero. Then, for each pair of profiles in subfamilies F1 and F2, the optimal score O_{nm} for the alignment of the nth profile in F1 and the mth profile in F2 is determined using a dynamic programming algorithm.

For any two pair of profiles, P_{1i} and P_{2j}, if the optimal score O_{ij} is significantly higher than the initial scores S_{1i} and S_{2j}, the two blocks are realigned (Fig. 3b). Figure 3c shows the final alignment with P_{12} realigned on P_{21} and P_{13} realigned on P_{22}.

(iii) Badly aligned orphan sequences Orphan sequences are only aligned once all the subfamilies are aligned together properly. The refinement process is the same as that used for correcting sequence errors within the subfamilies, except that only the global core blocks are considered and each orphan sequence is compared to the alignment of all clustered sequences.

(iv) Unreliable regions The final refinement step consists of the realignment of the regions between the global core blocks and the N/C terminal regions of the alignment. The global core blocks are recalculated in order to take into account the realignments performed in (i), (ii) and (iii) above. Each region between two consecutive core blocks is considered separately. All sequences are realigned using dynamic programming and a progressive alignment algorithm.

Downloaded from https://academic.oup.com/bioinformatics/article/19/9/1155/285019 by guest on 24 April 2024

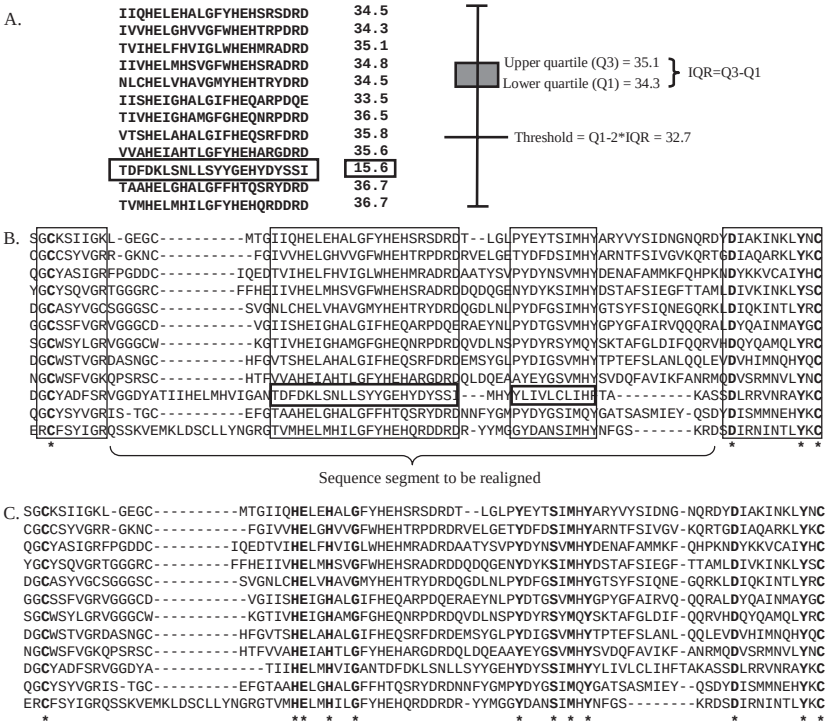

Fig. 2. Detection of badly aligned sequences. (**A**) A core block in a subfamily of 12 zinc protease sequences, with the two zinc binding histidines and the catalytic glutamic acid. Sequence versus profile scores are shown to the right of each sequence. The black box indicates a sequence 'outlier'. (**B**) The segment of the sequence to be realigned is extended to the nearest core blocks in which the sequence is correctly aligned. (**C**) After realignment of the sequence errors; conserved positions are bold and highlighted by asterisks.

IMPLEMENTATION

RASCAL consists of a suite of programs, all written in ANSI C. The programs were installed and tested on a DEC Alpha 6100 computer running OSF Unix. A UNIX shell script is provided that calls the C programs. The Secator program (http://www-bio3d-igbmc.u-strasbg.fr/~wicker/Secator/secator.html) is required for sequence clustering. RASCAL takes multiple alignments in either MSF or FASTA format as input and outputs the new refined alignment in either MSF or FASTA format, as requested by the user.

RESULTS

Benchmarking with BAliBASE

In order to evaluate the reliability and sensitivity of RASCAL, we used the BAliBASE benchmark alignment database (see Methods). For each BAliBASE reference alignment, automatic alignments were constructed using ClustalW, T-COFFEE, MAFFT-2 and MAFFT-I and were then refined using RASCAL. Table 1 shows the alignment scores before and after refinement. The most significant improvements were observed for the alignments constructed by ClustalW and MAFFT-2. This was to be expected as these two programs gave starting alignments of lower quality than either T-COFFEE or MAFFT-I. While RASCAL was relatively successful in correcting the alignment errors in reference sets 2 and 3 (see Methods), the other reference sets reveal one of the limiting factors of the RASCAL strategy; namely the number of sequences in the alignment. The other reference sets contain a relatively small number of sequences that cannot be sensibly divided into subfamilies and which are generally insufficient for the construction of meaningful core blocks. Nevertheless, for the majority of the 568 alignments tested, little deterioration of the initial alignment was observed (see Supplementary information).

A Friedman test (Friedman, 1937) ($S = 8$, $N = 142$, Test Statistic $= 59.22$) was used to statistically compare the accuracy of the alignment programs. For each test alignment, the programs were assigned a rank between 1 and 8 and the ranks were then summed over all alignments. Significant improvements were observed for the ClustalW, MAFFT-2 and MAFFT-I methods ($\alpha = 0.05$). Furthermore, the combination of the MAFFT-2 program with RASCAL performed as well as the programs MAFFT-I and T-COFFEE ($\alpha = 0.05$) and required significantly less CPU time.

Downloaded from https://academic.oup.com/bioinformatics/article/19/9/1155/285019 by guest on 24 April 2024

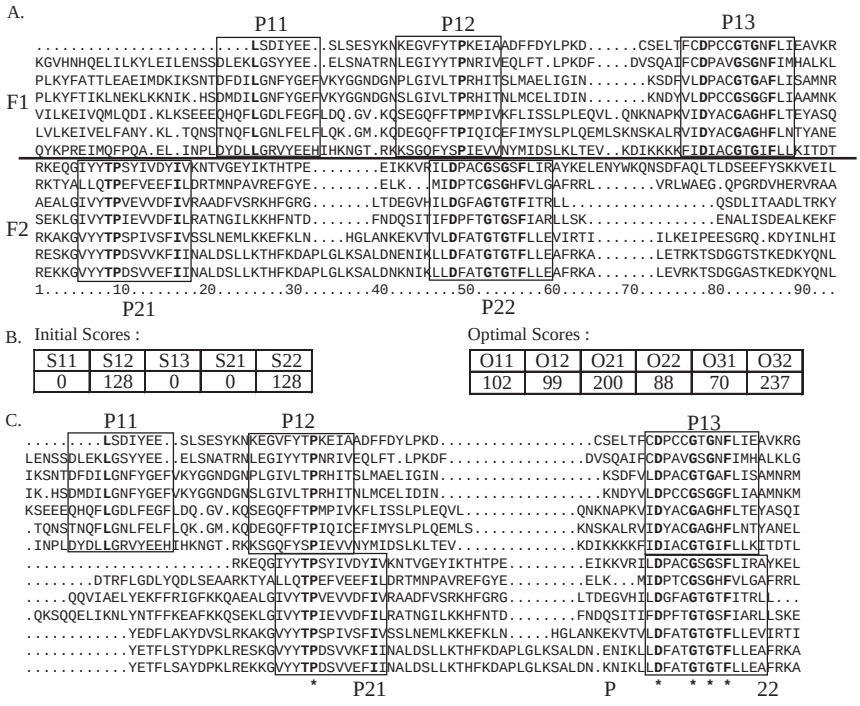

Fig. 3. Detection of badly aligned core blocks. (**A**) An alignment clustered into two subfamilies, F1 and F2. Conserved residues in each subfamily are in bold. (**B**) The initial scores (S_{ij}) and the optimal scores (O_{ij}) for the core blocks. (**C**) After realignment of the badly aligned blocks; conserved residues are marked by asterisks.

Table 1. Scores for 142 alignments from the BAliBASE database, before and after refinement by RASCAL

Reference Set	Initial alignment method	ClustalW Before	ClustalW After	MAFFT_2 Before	MAFFT_2 After	MAFFT_I Before	MAFFT_I After	T_COFFEE Before	T_COFFEE After
Reference 1	SP	0.86	**0.87**	0.84	**0.85**	0.86	**0.86**	0.86	**0.86**
	Column	0.78	**0.80**	0.76	**0.78**	0.79	**0.79**	0.78	**0.79**
Reference 2	SP	0.93	**0.94**	0.91	**0.92**	0.92	**0.93**	0.93	**0.94**
	Column	0.59	**0.63**	0.49	**0.53**	0.52	**0.57**	0.58	**0.59**
Reference 3	SP	0.72	**0.75**	0.75	**0.79**	0.76	**0.78**	0.75	**0.75**
	Column	0.48	**0.50**	0.49	**0.52**	0.52	**0.53**	0.51	**0.50**
Reference 4	SP	0.82	**0.83**	0.92	**0.93**	0.93	**0.93**	0.94	**0.94**
	Column	0.58	**0.62**	0.79	**0.80**	0.79	**0.80**	0.80	**0.81**
Reference 5	SP	0.86	**0.86**	0.96	**0.95**	0.96	**0.95**	0.96	**0.95**
	Column	0.63	**0.65**	0.85	**0.85**	0.86	**0.85**	0.90	**0.89**
CPU time		307	**84**	104	**86**	571	**81**	3902	**81**

The CPU time before refinement corresponds to the time required to construct the 142 initial multiple alignments. The scores and the time required for the RASCAL refinements are bold. SP = sum-of-pairs score, Column = column score

Large scale tests with ProDom domain alignments

Multiple alignments from the ProDom domain database were used to test the accuracy and reliability of RASCAL in a high-throughput, automatic system. A subset of alignments was selected (see Methods) that presented a number of different problems, including very large sets of sequences, highly divergent sequences, sequences of very different lengths, transmembrane sequences and sequences containing repeats. The alignments selected contained from 3 to 5000 sequences and differed in

Downloaded from https://academic.oup.com/bioinformatics/article/19/9/1155/285019 by guest on 24 April 2024

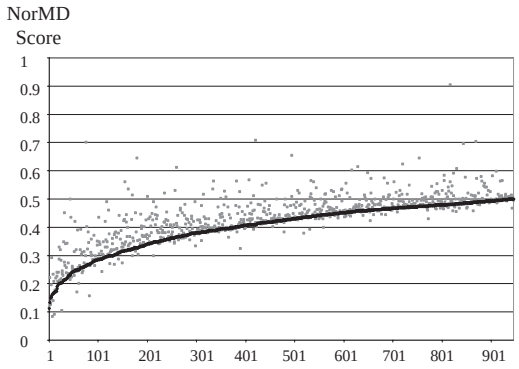

Fig. 4. The NorMD scores for 946 alignments from the ProDom database. The alignments are represented on the *x*-axis, sorted into ascending order of NorMD score for the initial ProDom alignments (black). The corresponding scores for the same alignments after refinement by RASCAL are grey.

length from 37 to 1622 alignment columns. As no reference alignments are available, we used the NorMD OF to compare the quality of the alignments before and after refinement by RASCAL (Fig. 4). After refinement, the NorMD scores of 645 (68%) of the alignments were increased. A Wilcoxon signed rank test (Wilcoxon, 1945) showed that the difference in scores is statistically significant ($p = 1.0$). An example alignment before and after refinement can be seen in the Supplementary information.

Alignment of complete sequences detected by database searches

Finally, RASCAL was incorporated in an in-depth structural and functional analysis of nuclear receptor proteins. A large set of 695 full-length nuclear receptor proteins were aligned using the MAFFT-2 progressive alignment program. MAFFT-2 is capable of providing a good initial alignment relatively fast (CPU time = 129 ; NorMD score = 0.27). The MAFFT-2 alignment was then refined using two different strategies: RASCAL and MAFFT-I. After refinement by RASCAL, the NorMD score of the alignment was increased to 0.57, compared to a score of 0.44 for the iterative strategy MAFFT-I. An important advantage of RASCAL is a considerable reduction in the CPU time. RASCAL required 528 s (9 min) to detect and realign the errors in the 695 sequences, in comparison to 16 198 s (4.5 h) for MAFFT-I. In addition to the correction of a number of alignment errors, RASCAL compacted the original alignment by removing many of the long gap regions, resulting in a much shorter, more structurally reliable alignment (see Supplementary information).

DISCUSSION

Most multiple alignment programs make alignment errors, particularly in difficult cases with highly divergent sequences. Iterative refinement techniques that optimize an OF are too computationally expensive to be practical for large alignments or high-throughput systems. Here, we have presented an alternative knowledge-based approach to the correction and refinement of alignments. The rationale of RASCAL is to incorporate information from a number of different, complementary techniques in an initial analysis phase. The goal is to differentiate the well conserved regions and to localize potential alignment errors. Currently, RASCAL uses NorMD to identify conserved 'core blocks' but other column scores could be incorporated (e.g. Hertz and Stormo, 1999; Pei and Grishin, 2001). Likewise, Secator is used to cluster the sequences into potential subfamilies, but other methods are now being investigated (e.g. Wicker *et al.*, 2002). Alignment correction is then restricted to those regions that contain potential errors, resulting in a more efficient refinement strategy. The second phase of the RASCAL method involves the hierarchical, progressive realignment of the low-scoring regions. RASCAL is not an iterative process: a single realignment is performed for each error detected. If no better solution is found, either because the sequences are not superposable in this region or because a sequence contains a frame-shift error, the alignment is not modified. We have shown that, while a significant increase in alignment accuracy is often obtained, no deterioration is observed even with high-quality initial alignments. In addition, RASCAL requires considerably less CPU time than existing iterative methods and can thus be applied to large scale projects.

An important criteria in the development of the RASCAL approach was the balance between accuracy and the CPU time required to detect and correct alignment errors. Therefore, we have restricted the search to some of the most common problems currently encountered when aligning real families of proteins: subfamilies, highly divergent sequences and divergent regions. An exhaustive search of all possible alignment errors would be considerably more expensive in terms of computer time. We have been conservative in the determination of errors in order to ensure that well aligned regions are not disturbed. This means that little improvement was observed for alignments containing few sequences, where it was not possible to build meaningful core block profiles. Future developments will include an investigation of other types of errors, as well as other potential methods of error correction.

RASCAL can be applied to any multiple sequence alignment, either from automatic methods such as ClustalW, MAFFT, or PSI-Blast or manually constructed alignments. The resulting improvement in alignment

Downloaded from https://academic.oup.com/bioinformatics/article/19/9/1155/285019 by guest on 24 April 2024

quality should lead to better accuracy and better coverage in the applications which rely on multiple alignments, e.g. 2D/3D structure prediction, homology modeling and database search techniques. Finally, the combination of a rapid multiple alignment algorithm, such as MAFFT-2, with post-processing by RASCAL opens the way to fast and accurate alignment of large sets of sequences.

ACKNOWLEDGEMENTS

The authors wish to thank N. Wicker for his help with Secator; O. Lecompte, Y. Brelivet for providing alignment test cases; F. Plewniak, L. Moulinier for useful discussions and Dino Moras for his continued support. We also wish to thank D. Kahn and E. Courcelle for stimulating discussions and exchange of information concerning the ProDom database. This work was supported by institute funds from the Institut National de la Santé et de la Recherche Médicale, the Centre National de la Recherche Scientifique, the Hôpital Universitaire de Strasbourg, the Fond National de la Science (GENOPOLE) and the Fond de Recherche Hoechst Marion Roussel.

REFERENCES

Altschul,S.F., Madden,T.L., Schaffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Carrillo,H. and Lipman,D.J. (1988) The multiple sequence alignment problem in biology. *SIAM J. Appl. Math.*, **48**, 1073–1082.

Corpet,F. (1988) Multiple sequence alignment with hierarchical clustering. *Nucleic Acids Res.*, **16**, 10 881–10 890.

Corpet,F., Servant,F., Gouzy,J. and Kahn,D. (2000) ProDom and ProDom-CG: tools for protein domain analysis and whole genome comparisons. *Nucleic Acids Res.*, **28**, 267–269.

Eddy,S.R. (1995) Multiple alignment using hidden Markov models. *Ismb*, **3**, 114–120.

Feng,D.F. and Doolittle,R.F. (1987) Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol.*, **25**, 351–360.

Friedman,M. (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Am. Stat. Assoc.*, **32**, 675–701.

Gotoh,O. (1996) Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments. *J. Mol. Biol.*, **264**, 823–838.

Gribskov,M., McLachlan,A.D. and Eisenberg,D. (1987) Profile analysis: detection of distantly related proteins. *Proc. Natl Acad. Sci. USA*, **84**, 4355–4358.

Gupta,S.K., Kececioglu,J.D. and Schaffer,A.A. (1995) Improving the practical space and time efficiency of the shortest-paths approach to sum-of-pairs multiple sequence alignment. *J. Comput. Biol.*, **3**, 459–472.

Heringa,J. (1999) Two strategies for sequence comparison: profile-preprocessed and secondary structure-induced multiple alignment. *Comput. Chem.*, **23**, 341–264.

Hertz,G.Z. and Stormo,G.D. (1999) Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, **15**, 563–577.

Katoh,K., Misawa,K., Kuma,K. and Miyata,T. (2002) MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Res.*, **30**, 3059–3066.

Jennings,A.J., Edge,C.M. and Sternberg,M.J. (2001) An approach to improving multiple alignments of protein sequences using predicted secondary structure. *Protein Eng.*, **4**, 227–231.

Lecompte,O., Thompson,J.D., Plewniak,F., Thierry,J.C. and Poch,O. (2001) Multiple alignment of complete sequences (MACS) in the post-genomic era. *Gene*, **270**, 17–30.

Notredame,C., Higgins,D.G. and Heringa,J. (2000) T-Coffee: a novel method for fast and accurate multiple sequence alignment. *J. Mol. Biol.*, **302**, 205–217.

Pei,J. and Grishin,N.V. (2001) AL2CO: calculation of positional conservation in a protein sequence alignment. *Bioinformatics*, **17**, 700–712.

Thompson,J.D., Higgins,D.G. and Gibson,T.J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, **22**, 4673–4680.

Thompson,J.D., Gibson,T.J., Plewniak,F., Jeanmougin,F. and Higgins,D.G. (1997) The CLUSTAL_X windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. *Nucleic Acids Res.*, **25**, 4876–4882.

Thompson,J.D., Plewniak,F. and Poch,O. (1999a) A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Res.*, **27**, 2683–2690.

Thompson,J.D., Plewniak,F. and Poch,O. (1999b) BaliBASE: a benchmark alignment database for the evaluation of multiple sequence alignment programs. *Bioinformatics*, **1**, 87–88.

Thompson,J.D., Plewniak,F., Thierry,J.C. and Poch,O (2000) DbClustal: rapid and reliable global multiple alignments of protein sequences detected by database searches. *Nucleic Acids Res.*, **28**, 2919–2926.

Thompson,J.D., Plewniak,F., Ripp,R., Thierry,J.C. and Poch,O. (2001) Towards a reliable objective function for multiple sequence alignments. *J. Mol. Biol.*, **314**, 937–951.

Wicker,N., Perrin,G.R., Thierry,J.C. and Poch,O. (2001) Secator: a program for inferring protein subfamilies from phylogenetic trees. *Mol. Biol. Evol.*, **18**, 1435–1441.

Wicker,N., Dembele,D., Raffelsberger,W. and Poch,O. (2002) Density of points clustering, application to transcriptomic data analysis. *Nucleic Acids Res.*, **18**, 3992–4000.

Wilcoxon,F. (1945) Individual comparisons by ranking methods. *Biometrics*, **1**, 80–83.

Downloaded from https://academic.oup.com/bioinformatics/article/19/9/1155/285019 by guest on 24 April 2024