Higher accuracy protein Multiple Sequence Alignment by Stochastic Algorithm

[1]Jeevitesh.M.S, [1,2,*]Narayan Behera, [1]Justin Jose, [3]Krishna Kant & [3]Alpana Dey

[1]Institute of Bioinformatics and Applied Biotechnology, G-05, Discovery Building, ITPB, Whitefield Road, Bangalore - 560066, India;

[2]Poorna Prajna Institute of Scientific Research, 4, Sadashivanagar, Bangalore – 560080, India;

[3]Department of Information Technology, Electronics Niketan, 6 CGO Complex, New Delhi - 110003, India

Multiple Sequence Alignment (MSA) gives insight into the analysis of sequence conservation, and evolutionary, structural and functional relationships among the proteins. Constructing a precise MSA for proteins is an arduous task as the computational complexity rises with sequence length and number of sequences. Here, a stochastic algorithm is developed to find a more accurate protein MSA to handle fairly large number of sequences with high sequence length. This method basically uses the alignment outputs of two important individual MSA programs – MCoffee and ProbCons – and combines them in a genetic algorithm model. The evolutionary operators of the genetic algorithm, namely, mutation and selection are utilized in the optimizing process of the algorithm. Performance of this Protein Alignment by Stochastic Algorithm (PASA) tool is tested on the Balibase version 3 benchmark reference protein datasets. The efficiency of protein sequence alignments is evaluated in terms of Total Column (TC) score which is equal to the number of correctly aligned columns between the test alignment and the reference alignment divided by the total number of columns in the reference alignment. In terms of TC scores, the PASA optimizer achieves, on an average, significant better alignment over the well known

individual bioinformatics tools. This PASA bioinformatics tool is statistically the most accurate protein alignment method. It can have potential applications in the drug discovery processes in the biotechnology industry.

*corresponding author

Email: nbehera@ibab.ac.in

Generally, Multiple Sequence Alignment (MSA) for proteins is used to understand the extant of sequence conservation, and to find the evolutionary, structural and functional relationship among the protein families. MSAs are also necessary for building character profiles, establishing phylogenetic relationship, designing primer in PCR experiments and predicting protein structures. Constructing a precise MSA for proteins is still a difficult task because the computational complexity grows very fast in proportion to the sequence length and the number of sequences. Furthermore, it is tough to create an objective function to assess the alignment quality[1]. An accurate solution is possible only for a small number of related sequences[2]. Therefore, most MSA packages use heuristic progressive alignment algorithm that don't necessarily provide optimal solutions[3].

Some of the routinely used well-known sequence alignment programs are ClustalW, Mafft, Muscle, TCoffee, M-Coffee and ProbCons. The popular ClustalW uses the progressive alignment approach to align protein sequences. It builds up a multiple alignment progressively using a series of pair wise alignments by using phylogenetic tree as the reference. It first aligns the most closely related sequences and then the distant ones. But it suffers from the limitation that the alignment errors made early in the process can never be rectified[4]. T-Coffee method uses a consistency-based objective function where an information library is built up by the local and global alignments and also from heterogeneous sources, such as a mixture of alignment programs and structure

superpositions[5]. M-Coffee program is a meta-method that merges the output of various MSA programs into one single better alignment. It is an extension of the T-Coffee method which uses consistency approach to build an alignment[6]. Mafft tool uses a Fourier Transform to determine the homologous position in which an amino acid's volume and polarity are taken into account[7]. However, the Muscle method of sequence alignment is based on iterative progressive alignment algorithm. It uses the traditional sum of pairs score as the alignment quality[8]. ProbCons method is a progressive protein multiple alignment algorithm that uses probabilistic consistency information in constructing the alignment[9].

The manually constructed protein structural alignments are needed as benchmarks to compare the effectiveness of various MSA tools. The Balibase 3.0 benchmark alignment database is an assembly of 386 structural protein alignments which are manually verified alignments[10]. This benchmark is categorized into five different groups. The first group is made of phylogenetically equidistant members of similar length. The second group contains up to three orphan sequences with close relatives. The third group contains distantly related sequences, while the fourth and fifth groups involve long terminal and internal insertion respectively. The basic purpose of a benchmark is to provide a set of tests to compare the efficiencies of alternative computational tools. So, the idea behind this benchmarking is that the average best performing software package will be able to find the best alignment of the uncharacterized protein sequences.

In this model, we use the sequence alignment outputs of two programs, namely, ProbCons and MCoffee as the initial alignments. Then a stochastic algorithm with suitable mutation and selection operators is used to get a better alignment. The stochastic algorithm is a search algorithm that imitates the processes in natural evolutionary systems. It is modeled on the principles of evolution via natural selection, employing a population of individuals (multiple alignments) that undergo selection in the presence of variation inducing mutation operator. It is variously called as genetic algorithm, evolutionary

computation and so on. Holland first introduced this type of algorithm and later it has been applied to many problems in science and engineering systems in finding optimal or near optimal solutions[11]. Genetic algorithm technique has been successfully implemented in various multiple sequence alignment problems[12-14]. Genetic algorithm as a sequence alignment optimizer has been successfully applied by taking ClustalW as the initial seeding alignment[15].

In this model, a better protein alignment solution is allowed to evolve over many generations, starting from a set of population of alignments. The idea is to obtain the most optimized solution. An objective function is used to evaluate the quality of individual alignments. In the current model, new kinds of operators, such as block shift and block removal are implemented. The initial 100 population of MSAs from the two program outputs, ProbCons and MCoffee are constructed with equal probability. The Protein Alignment by Stochastic Algorithm (PASA) combines and improves the alignments over successive generations until the most optimized alignment is obtained at the end. This PASA method is tested on the most accepted benchmark, Balibase version 3. It achieves a statistically significant enhancement of alignment over the other popular methods.

The alignment quality of each bioinformatics tool is determined by measuring: Quality (Q) and Total Column (TC) scores. Q is the number of correctly aligned residue pairs between test alignment and reference alignment divided by the total number of aligned residue pairs in the reference alignment. TC is the number of correctly aligned columns between test alignment and reference alignment divided by the total number of columns in the reference alignment. In general, TC score is lower than the Q score. However, the TC score provides a more important measure to evaluate the efficiency of a sequence alignment as far as the conserved blocks are concerned. So we are using TC score to find the best alignment for the purpose of analysis. Then the corresponding Q for that alignment is determined. TC and the Q scores are calculated using software QSCORE.

In order to compare the performance of various MSA programs with the PASA method, we conduct the Friedman rank test. This is basically a non-parametric test. It makes no assumption about the distribution of alignment scores across different pairs of MSA programs. Here, instead of using alignment score directly, the ranking of the score across the pairs of programs is used for finding the efficiency of a MSA method. The higher the alignment score of an alignment program, the better is its rank. Then the ranksum is calculated as the sum of ranks for a given MSA program. The concept of null hypothesis is used to compare the efficiencies of two MSA programs in terms of TC and Q values. Null hypothesis assumes that a pair of programs is equally likely to be good. The ranksum is further used to calculate the P-value which measures a probability factor for rejecting the null hypothesis. If the P-value is very small (say, ≤ 0.05), the above null hypothesis is rejected. Furthermore, the higher the ranksum, the better is the program. If the P-value is greater than 0.05, there is no statistically significant difference between the efficiencies of two comparable MSA programs. For a set of scores (say, Q and TC) the P-values are obtained using the Friedman rank test from the statistical analysis package R (http://www.r-project.org/).

TC and Q scores and the statistical significance of the alignments are summarized in tables 1-2. As the algorithm is stochastic, the result will be dependent on initial conditions and the way random numbers are called during the run of the program. That is why twelve different simulations are considered to find the best result. More number of simulations will increase computational time while less number of simulations will provide less efficient result. So there are twelve pairs of TC and Q scores for an alignment. But while deciding the best alignment, the best TC is considered. The best Q is considered for that corresponding alignment. As there is no straightforward relationship between Q and TC, a higher Q score might have been lost in some cases. When the alignment of unknown sequences are considered, the alignment having the highest alignment score is picked up

from the twelve simulations and then the corresponding TC and Q values are found. On all the test sets and quality measures, PASA model achieves the highest ranking and a statistical significant enhancement over the well known alignment methods. In the ranksum of Friedman rank test, the program with the highest ranksum most often constructs the most accurate alignment.

The results on Balibase benchmark alignment database are shown in Table 1-2. PASA achieves improvements of 0.7% over the MCoffee, 1.2% over the ProbCons, 14% over the ClustalW, and 9.28% over the Mafft in terms of Q scores on Balibase 3 benchmark, as shown in Table 1. PASA has enhancements of 3.6% over MCoffee, 7% over ProbCons, 28% over ClustalW, 24% over Mafft, 14% over Muscle and 24% over TCoffee in terms of TC scores measured on Balibase 3 as shown in Table 1. The statistically significant differences in the overall TC and Q scores are shown in Table 2.
It is shown that PASA tool is able to improve sequence alignment by 3%-26% in terms of TC scores measured on the Balibase benchmark 3 protein dataset.

It requires delicate analysis of the stochastic algorithm to obtain the best alignment. A number of operators, such as block insertion, block shifting, block searching in terms of the gaps and different types of block crossover and point crossover have been tried. Most of those operators have improved the sum of pair scores but in terms Q and TC scores, they have failed. In our model we are not using point and other type of crossovers as they are found to be disrupting for the alignment. Finding the highest alignment score of a multiple protein alignment is an open field of research that is evolving rapidly. We have used a simple idea of evolutionary optimization and a genetic algorithm model to start the initial population of alignments as the MSA program outputs of two most efficient tools, ProbCons and MCoffee. These two sequence alignment programs are different than the others in the sense that the average length of characters in the aligned sequences are greater than the corresponding aligned sequences found in Balibase version 3 reference alignment.

So the mutational procedure of gap elimination operator plays a significant role in enhancing the final alignment. Eventually we have obtained significant enhancement of alignment in terms of Q and TC scores in comparison to the individual MSA methods. It is very interesting to note that this PASA alignment program structure is such that the program running time reduces by a factor of about 10 when the codes are written in C language instead of using Perl.

It has been reported that structural alignment programs produce outputs where 11% – 19% of the core residues are misaligned[16]. Majority of the benchmark alignments are obtained by using the structural alignment programs. So there is a concern over the effectiveness of the benchmark alignments. In that case, we suspect that our PASA alignment program will provide better alignment accuracy.

This type of analysis can be extended to RNA alignments although the work would be very messy.

It is known in the scientific literature that although a genetic algorithm model can give better result, it takes generally more computational time due to the stochastic nature of the algorithm. Here we have tracked the program running time for the Balibase subset RV12 protein reference alignment benchmark (consisting of 88 alignments). It takes 17 minutes and 79 minutes in the case of Probcons and MCoffee respectively while for the PASA program it is 96 minutes. The absolute computational time can be drastically reduced when the program is allowed to run on a multi-cluster system having hundreds of nodes. So the PASA tool gives statistically better alignment over the two competing programs while maintaining the scale of computational time. The traditional genetic algorithm approach towards sequence alignment like SAGA tends to build alignment from the initial random alignments of sequences[17]. But in our current approach, initial alignment solutions are near to global optimum as they are the outputs of other important programs. So it takes very less time compared to the conventional genetic algorithm programs (data not shown). This

proves that the stochastic algorithm can be used as an excellent optimizer for the sequence alignment problem. In finding a better sequence alignment, generally it is a tough task to choose the right MSA program over several programs available in the literature. So the present PASA is a better alternative to combine some of the efficient individual methods and further improve them to find still better alignment. Furthermore, the PASA is quite robust with respect to the evolution of novel individual methods. The PASA has incorporated biological knowledge such as structure based derived matrix and some novel genetic algorithm operators. It generates good results even below the twilight zone of sequence similarity. PASA tool achieves a statistically significant result over the popular and efficient protein alignment programs like MCoffee, ProbCons and others.

Method

Assessment of a multiple sequence alignment is made by using an Objective Function (OF). The fitness value reflects the quality of multiple sequence alignment. It also provides an insight into the implicit structural and evolutionary relationships that subsist among the aligned sequences. We apply the sum of pairs method as a measure for alignment quality. The objective is to maximize the score of alignment. This sum of pair scores (S) is defined as,

$$S = \sum_{i} \sum_{j} \mathbf{S}(i, j) \qquad \textbf{(1)}$$

where i = 1, 2, .., n-1 (where n = number of sequences in the alignment), j = i +1, i+2, .., n and S(i,j) is the value obtained using structure based matrix. The overall alignment score of a MSA is the sum of each pair of rows. The alignment score of a pair of rows is the sum of the alignments of the individual pair of residues.

Mutation is a significant part in the genetic algorithm for finding the optimal solution. It helps to prevent the population from stagnating at any “local optima”. Mutation

alters one or more positions in the sequence from its initial state. Mutation is implemented by inserting a gap randomly in a sequence. This can result in an entirely new alignment. With these new sequence alignments, the genetic algorithm may be able to arrive at a better alignment.

For each alignment in the population of alignments, gaps are inserted randomly with a fixed probability (p) given by the following formulae.

$$p = \ln(xy)/(I \times 10) \qquad (2)$$

where x is the maximum length of a sequence in the multiple sequences, y is the number of sequences and I is the number of columns with identical residues (ignoring gaps). The equation (2) has been empirically obtained after analyzing a few set of alignment data.

Only a portion of the population of alignments is to be replaced during each generation. The simulation terminates when the difference of the best fitness for ten consecutive generations is less than 1%. At the n^{th} ($n>10$) generation, the percentage differences between the best fitness of $(n-i)^{th}$ generation and $(n-10)^{th}$ generation are found, where i varies from 0 to 9. If all these ten differences are less than 1%, the program is terminated, else it proceeds to the next generation.

Full methods will be available on the online version of the paper.

1. 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.

2. Lipman,D.J., Altschul,S.F. and Kececioglu,J.D. (1989) A tool for multiple sequence alignment. *Proc. Natl Acad. Sci.*, **86**, 4412-4415.

3. Hogeweg,P. and Hesper,B. (1984) The alignment of sets of sequences and the construction of phylogenetic trees. An integrated method. *J. Mol. Evol.,* **20**, 175-186**.**

4. Thompson,J.D., Higgins,D.G. and Gibson,T.J. (1994) CLUSTALW: 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.

5. 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.

6. Wallace, I.M., O'Sullivan, O., Higgins, D.G. and Notredame, C. (2006) M-Coffee: combining multiple sequence alignment methods with T-Coffee. *Nucleic Acids Res*., **34,** 1692-1699.

7. Katoh, K., Misasa, 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.

8. Edgar,R.C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res*., **32**, 1792-1797.

9. Do,C.B., Mahabhashyam,M.S., Brudno,M., Batzoglou,S. (2005) ProbCons: Probabilistic consistency-based multiple sequence alignment. *Genome Res*., **15**, 330-340.

10. Thompson, J.D., Koel, P., Ripp, R. and Poch, O. (2005) Balibase 3.0: Latest Developments of the Multiple Sequence alignment Benchmark. *Proteins*., **61,** 127-136.

11. Holland, J.H. (1975) Adaptation in natural and artificial systems. Univ of Michigan press, Ann Arbor, MI.

12. Zhang, C and Wong A.K. (1997)A Genetic algorithm for multiple molecular sequence alignment. CABIOS, 13(6): 565-581.

13. Cai, L, Juedes, D and Liakhovitch, E. (2000) Evolutionary computation techniques for multiple sequence alignment. Proceedings of the second congress on evolutionary computation. 829-835.

14. Anbarasu, L. A., Narayanasamy and Sundararajan, V. (2000) Current Science Vol. 78 858-863.

15. Thomsen, R., Fogel, G.B and Krink, T. A (2002) ClustalW alignment improver using evolutionary algorithms. Proceedings of the fourth congress on evolutionary computation 1 121-126.

16. Kim, C., Lee, B. (2007) Accuracy of structure-based sequence alignments of automatic methods. *BMC Bioinformatics*., **20**, 8355

17. Notredame, C., Higgins, D.G. (1996) SAGA: sequence alignment by genetic algorithm. *Nucleic Acids Res*., **24**, 1515-1524.

Acknowledgements We thank Shinsuke Yamada of the Yamana Laboratory, Waseda University, JAPAN and Kishlay Kumar Singh of IBAB, Bangalore, India for certain statistical analysis. N.B. acknowledges a research grant from the Department of Information Technology, Government of India (grant no.- DIT R&D)/BIO/15(5) /2006 to NB)

Author contributions N.B. supervised the research work, J.M.S. developed the major part of the algorithm and did the statistical analysis, J.J. helped in developing the algorithm, K.K. and A.D. had given some suggestions about the model.

Author information Correspondence and request for materials should be addressed to N.B. (HYPERLINK "mailto:nbehera@ibab.ac.in" nbehera@ibab.ac.in).

Table legends

Table 1: Average of TC scores on the Balibase benchmark

The column represents the average of TC score for all the alignments. The Ranksum

values are obtained from the Friedman test for all the alignments. The highest score in each benchmark set is highlighted in bold.

Table 2: Statistical analysis on the Balibase benchmark 3

Each value in the table contains the P-value assigned by the Friedman rank test, indicating the significance of the difference of alignments between the programs. The upper triangle matrix values are derived from the Q scores on the Balibase 3. The signs + and - represent that a program in a row performs significantly better and worse respectively than that of a program in a column. If the P-value is greater than 0.05, the difference is not significant and is shown in parentheses. For example, the PASA ranks higher than the ClustalW with a P-value of 2.2×10^{-16}. The lower triangle matrix values are obtained from the TC scores on the Balibase 3.

Table 1

Methods	Ref 1.1 (76)	Ref 1.2 (88)	Ref 2 (82)	Ref 3 (60)	Ref 4 (49)	Ref 5 (31)	Overall (386)	Ranksum
PASA	**35.36**	**67.41**	**36.12**	**39.34**	**21.47**	**33.33**	**38.83**	**2312.5**
MCoffee	32.74	66.47	33.94	38.15	21.12	32.43	37.47	2060.5
ProbCons	34.02	64.79	33.50	34.53	19.64	31.17	36.27	1783
ClustalW	23.11	58.70	26.03	29.48	18.58	25.07	30.16	1030.5
Mafft	23.46	57.25	28.53	32.48	18.68	27.44	31.30	1050
Muscle	30.07	62.36	28.87	32.87	20.37	28.64	33.86	1364.5
TCoffee	25.89	62	28.23	27.88	19.02	23.83	31.14	1207

Table 2

Methods	PASA	MCoffee	ProbCons	ClustalW	Mafft	Muscle	T Coffee
PASA		+1.0 x 10^{-08}	+4.8 x 10^{-12}	+<2.2 x 10^{-12}	+<2.2 x 10^{-12}	+< 2.2 x 10^{-16}	+< 2.2 x 10^{-16}
MCoffee	-< 2.2 x 10^{-16}		+3.6 x 10^{-06}	+ < 2.2 x 10^{-16}	+< 2.2 x 10^{-16}	+< 2.2 x 10^{-16}	+< 2.2 x 10^{-16}
ProbCons	-< 2.2 x 10^{-16}	-1.9 x 10^{-09}		+< 2.2 x 10^{-16}	+< 2.2 x 10^{-16}	+< 2.2 x 10^{-16}	+< 2.2 x 10^{-16}
ClustalW	-< 2.2 x 10^{-16}	- < 2.2 x 10^{-16}	- < 2.2 x 10^{-16}		-1.6 x 10^{-10}	-< 2.2 x 10^{-16}	-< 2.2 x 10^{-16}
Mafft	-< 2.2 x 10^{-16}	- < 2.2 x 10^{-16}	- < 2.2 x 10^{-16}	(0.60)		-< 2.2 x 10^{-16}	- 4.5 x 10^{-16}
Muscle	-< 2.2 x 10^{-16}	- < 2.2 x 10^{-16}	-8.0 x 10^{-16}	+2.1 x 10^{-16}	+9.0 x 10^{-09}		(0.33)
TCoffee	-< 2.2 x 10^{-16}	- < 2.2 x 10^{-16}	- < 2.2 x 10^{-16}	2.4 x 10^{-16}	+0.3 x 10^{-04}	(0.34)	

Full methods

We have implemented affine gap penalty. In this scheme two types of penalties are used for the score calculation: one for gap opening and the second for gap extension. The gap opening penalty is applied only once when a gap is introduced into the sequence and the gap extension penalty is added to the standard gap penalty for each additional gap. Optimum gap opening penalties are tested in the range from 5 and 20 and the extension penalties are tested between 0 and 2. It is observed that a gap opening penalty of 15, and gap extension penalty of 0.9 yielded higher accuracy. The terminal gaps are not scored. Sequence weights are incorporated in a multiple sequence alignment in order to correct the unequal representation. It has been observed that the inclusion of weighting scheme gives only a small improvement in alignment accuracy (about 1%) on the Balibase benchmark (16). So PASA is not implementing any kind of weighting scheme.

For a gap insertion in a MSA, a random number r is generated in the range of 0 to 1 and for $r < p$, a gap is inserted at a random position else no gap is inserted in that alignment. After insertion of a gap, the remaining sequences of the MSA are padded with gaps so that all sequences are of the same length.

In the Hill Climbing mutation, a new solution is obtained by mutation if the new solution is fitter. Otherwise the current solution is retained. The Hill Climbing algorithm works as follows:

For each alignment $\boldsymbol{a}$ of the population of alignments, its current fitness $f(\boldsymbol{a})$ is calculated. Mutate $\boldsymbol{a}$ to produce a mutant $\boldsymbol{m}$ by inserting a gap randomly in one of the sequences and

then gaps are padded at the end of the other sequences.

If f (***m***) is fitter than f (***a***), then replace ***a*** with ***m*** else ***a*** is retained.

The fundamental idea behind this local search is that the good solutions tend to cluster together.

Block shift

Inspired from the natural phenomenon of jumping genes, where a genetic material moves around to different positions within the genome, we have introduced an operator that searches a block of gaps and shift it to the neighboring positions.

Steps are as follows:

1. Generate two random numbers, r1 and r2 to find out the sequence number and the character position in that sequence respectively.

2. The position corresponds to r2 can be a character or a gap.

2.1 If it is a character then look forward for a gap and that position is taken as the gap starting position (Gs). If we don't find any gaps, then move on to the next individual of alignment.

2.2 Count the no of gaps from the gap starting position (Gs) and it is considered as the Gap counts (Gc).

2.3 Now find out the first gap starting position and count gaps (up to next character) for all other sequences from Gs to (Gs + Gc).

2.4 From 2.2 and 2.3 we can find out the block of gaps (Bg) for the shift.

3. If the position corresponding to r2 is a gap then,

3.1 Look backward for a gap and find out the starting position of the gap. This is taken as the gap starting position (Gs).

3.2 Count the no of gaps from the gap starting position (Gs). It is considered as the Gap counts (Gc).

3.3 Now find out the first gap starting positions and gap counts (up to next character) for all

other sequences from Gs to (Gs+Gc).

3.4 From 3.2 and 3.3 we can find out the block of gaps (Bg) for the shift.

4. Now we have the block of gaps (Bg) to be shifted.

5. First we shift towards right for one place and check whether the alignment score is increased or not.

6. If the score increases, the individual is replaced by the shifted individual. Then we go for the next individual.

7. If the score does not increase, we do right shift once more and follow step 6.

8. If the score does not increase, we perform left shift by one position on the individual and follow step 6.

9. If the score does not increase we do left shift by one more position and follows step 6.

10. We perform two times right shift and two times left according to the score. If the score is not increased after all the four shifts, the individual is retained in the population and the next individual is considered.

Note here that we are not performing top and bottom block shift because as the blocks of gaps are irregular, it will disrupt the entire alignment.

Block elimination

Motivated from the biological phenomenon of genetic elimination, where genetic material is eliminated from the genome, we have designed an operator block removal. Keeping in mind the basic rule of multiple sequence alignment i.e., minimum number of gaps should be there in order to construct a multiple sequence alignment.

Steps are as follows:

Generate two random numbers, Rs for the sequence number and Rp for the position in that sequence respectively.

Find out a gap from Rp (gap starting position Gsp) and number of gaps from that position

to the very next residue for the sequence Rs. If there is no gap, then choose the next individual in the population.
Find out the gap starting positions and number of gaps for all other sequences.
Find out the first common block of gaps which includes maximum number of sequences.
Eliminate this block of gaps from the alignment.
Now delete the same number of gaps from all other sequences which are not included in the formation of block gaps.
Compare the scores before and after the block removal.
If the score increases retain that individual in the population, else discard the individual.

Half of the high scoring alignments will survive unchanged while the other half is replaced by the alignments generated by block shifting and removal operators.

To assess the efficiency of the PASA protein benchmark suites: the Balibase 3.0 is used. The program is implemented on a 3 GHz Intel Xeon Dual core processor with 8 GB RAM. Fedora core 6 is used as the operating system. The PASA program is compared with the Probcons version 1.11, Mcoffee, ClustalW version 1.83, the T-Coffee version 4.96, the Mafft version 5.861 and the Muscle version 3.6. All the above programs are exceuted on default modes.

Supplementary information is linked to the online version of the paper.

Table legend

Table 1A: Average of Q scores on the Balibase benchmark

The column represents the average of Q score for all the alignments. The Ranksum values are obtained from the Friedman test for all the alignments. The highest score in each benchmark set is highlighted in bold.

Table 1A

Methods	Ref 1.1 (76)	Ref 1.2 (88)	Ref 2 (82)	Ref 3 (60)	Ref 4 (49)	Ref 5 (31)	Overall (386)	Ranksum
PASA	**59.29**	**86.72**	**85.54**	**76.88**	**70.83**	**74.9**	**75.69**	**2322**
MCoffee	58.18	86.32	85.14	76.14	70.35	74.87	75.16	2170
ProbCons	59.21	85.80	84.68	74.80	70.28	73.98	74.79	2003.5
ClustalW	46.82	79.64	79.70	65.86	61.73	63.17	66.15	732.5
Mafft	47.08	80.58	81.77	72.04	65.22	68.91	69.26	948.5
Muscle	53.23	83.31	82.99	72.16	66.80	69.61	71.35	1326
TCoffee	50.08	83.94	83.72	70.24	66.68	70.46	70.85	1305

PAGE

PAGE 1