DOI 10.3233/BME-151479
IOS Press

A comparative analysis of multiple sequence alignments for biological data

Umar Manzoor[a,*], Sarosh Shahid[b] and Bassam Zafar[a]
[a] *Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia*
[b] *Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad, Pakistan*

Abstract. Multiple sequence alignment plays a key role in the computational analysis of biological data. Different programs are developed to analyze the sequence similarity. This paper highlights the algorithmic techniques of the most popular multiple sequence alignment programs. These programs are then evaluated on the basis of execution time and scalability. The overall performance of these programs is assessed to highlight their strengths and weaknesses with reference to their algorithmic techniques. In terms of overall alignment quality, T-Coffee and Mafft attain the highest average scores, whereas K-align has the minimum computation time.

Keywords: Multiple sequence alignment, clustalW, k-align, muscle, mafft, t-coffee, progressive alignment

1. Introduction

The idea that the construction of evolutionary relationships between organisms can be interpreted using DNA based sequences, proposed by Crick, laid the foundation of modern evolution and comparative genomics [1]. Nowadays, biological databases contain a huge amount of DNA and protein sequence data collected from high throughput experiments in biotechnology. One of the challenging tasks is to analyze these sequences and extract biologically significant but hidden information [2]. Construction and analysis of multiple sequence alignment (MSA) is a prerequisite in these studies and in post-genomic biological research [3].

MSA construction is a way of aligning more than two sequences, either DNA or protein, and identifying homologous positions in columns by placing gaps. These gaps indicate insertion or deletion of residues (amino acids or nucleotides). The sequences are then aligned after identification of similarities between two sequences [4]. Next, a substitution matrix is used to assign a score to each column on the basis of matches, mismatches and gaps. The substitution matrix contains a score for each amino acid substitution [5].

[*] Address for correspondence: Umar Manzoor, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia. Tel.: +96622870026; Fax: +96622870024; E-mail: umarmanzoor@gmail.com.

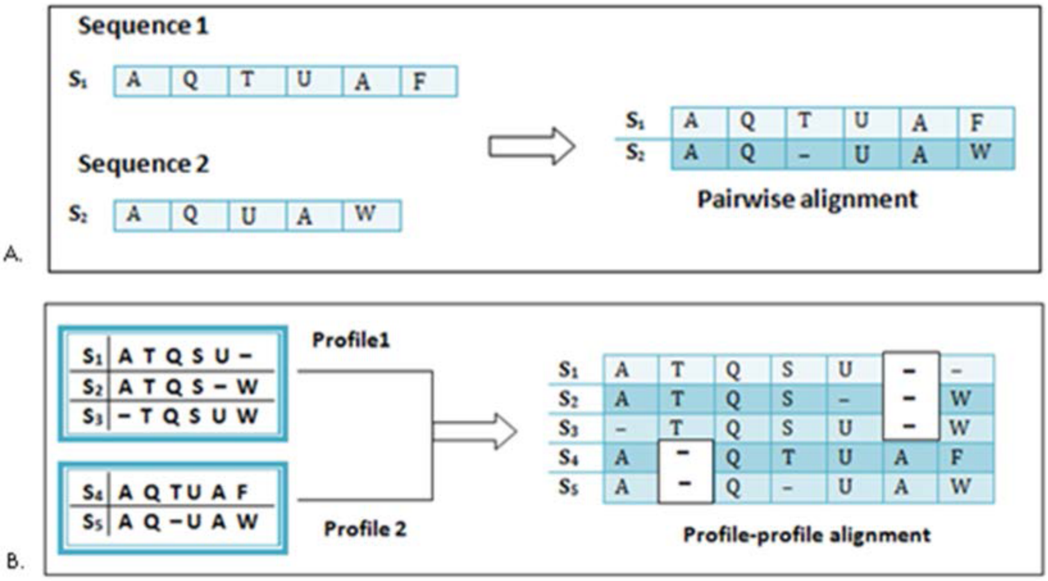

Fig. 1. Types of alignment. (A) Sequence-sequence alignment. (B) Profile-profile alignment. A profile is formed by aligning sequence with another sequence (or profile).

The primary aim of MSA is to detect similarities between sequences and evolve evolutionary relationships between these sequences. Construction of a phylogenetic tree from MSA leads to structure and functional predictions of sequences as well. Many sequences remain conserved throughout evolution. By highlighting these conserved regions, the motif, domain and catalytic sites of proteins can be obtained [3].

Biological modeling methods are extensively dependent on MSA. A number of different algorithmic techniques have been proposed in the past, but none of these programs are capable of delivering 100% accurate results. The computational complexity to calculate an exact optimal solution of MSA for N different sequences, each having length L, is O (L^N). However, with this complexity, even the computation of small sequences takes more time than desired. To achieve maximal accuracy, heuristic methods are commonly used [6].

2. Literature review

Different methods adopt different algorithmic techniques to compute MSA. In this paper, the algorithmic techniques of five different MSA programs that are the most popular are discussed.

2.1. ClustalW

The most common method to construct MSA is progressive alignment. In this method, two sequences are aligned first, and then the remaining sequences are added one by one. ClustalW is the most common program that uses the progressive alignment technique [7]. In ClustalW, the first step is the computation of the distance matrix, which represents the similarity between all the input sequences in the form of floating point values. These values are computed by performing pair-wise alignment of all sequences and counting the number of identical residues between two sequences. In the second step, a guide tree is constructed from the distance matrix using the neighbor-joining algorithm. In the last step, sequences are added one by one in the alignment on the basis of branching order in the tree to construct MSA. In the beginning, a sequence is aligned with another sequence, but later, at each subsequent step, one of the following conditions arises: profile-profile alignment or sequence-profile alignment, as shown in Figure 1 [8].

2.2. MUSCLE

In MUSCLE, progressive alignment is used, and MSA results are refined by continuous iteration. In the first step, the similarity of each sequence is calculated using k-mer. On the basis of pair-wise similarities, the distance matrix is computed and then a tree is constructed, using either UPGMA or the tree joining algorithm. The tree is traverse in postfix order, and MSA is produced at the root. In the second step, the similarity of each pair of sequences is calculated, using fractional identity from the already-constructed MSA, and another tree is constructed using the clustering method. Both of these trees are compared, and the alignment of only the different nodes is reconstructed. When the root is approached, the algorithm either loops to the second step or jumps to the third step. In the third step, alignment is refined iteratively. Firstly, an edge is deleted from the tree, dividing it into two unique subsets. The columns containing no residues are then deleted from the two profiles extracted in the previous step. The two profiles are then realigned using profile alignment, and the score of this new alignment is evaluated. The score is calculated using the sum-of-pairs (SP) score. In SP, scores are calculated by adding the substitution matrix score for each aligned pair, and a penalty is assigned to the gap. If the score is greater than the previous alignment, it is retained; otherwise, this alignment is discarded. Although the iterative approach yields more accurate results, the iterations increase the execution time.

In order to find similarity, MUSCLE used two different similarity measure methods: k-mer and fractional identity. K-mer is a sequence of substrings, and it is acknowledged that related or homologous sequences contain more frequent k-mer as compared to divergent sequences. The basic motivation of k-mer is to avoid pair-wise alignment of all the sequences and reduce the algorithm's complexity. K-mer enhances speed in computing the distance matrix by three times, compared to dynamic programming. In fractional identity, the sum of the columns that contain similar residues are calculated and then divided by total number of columns (it neglects the columns that contain only gaps) [9].

2.3. K-align

K-align computes the distance between sequences using the Wu-Manber approximate string-matching algorithm. In terms of computational speed, this method is as fast as the k-mer similarity measure with improved accuracy. Take two sequences—'XXYXXY' and 'XXXXXX'—that are 66% identical and aligning these sequences using k-mer with window size 3 does not yield any matching within the string. In K-align, a dynamic programming matrix is constructed, and a score is assigned to each column. The sequence similarity is then calculated on the basis of the highly scored diagonals. After finding the similarity scores and constructing a guide tree, sequences are aligned in similar fashion as the above two programs. K-align provides the users with another option for incorporating the dynamic matrix score of the pattern matching from previous steps to improve the quality of the alignment. For this purpose, two extra steps are included in the algorithm: a consistency check and updating of the pattern match position. In the consistency check, the number of matching patterns is identified in both sequences A and B involved in the current alignment. Updating patterns matches the position deals with the profile; when a sequence is aligned with a profile, the similarity score is disturbed because of already-inserting gaps in the profile. In this step, the position of residues in a profile is adjusted, although it matches with a sequence [10].

2.4. T-Coffee

Almost all the programs using progressive alignment for the construction of MSA, including ClustalW and MUSCLE, are based on the greedy approach. When a tree is constructed, the program assumes that the branching order of the tree will give optimal results, but this is not always true. For example, consider four sequences S_1, S_2, S_3 and S_4, such that S_1 and S_2 are similar, whereas S_3 and S_4 are closely related. While aligning S_1 and S_2, sequences S_3 and S_4 are not considered. In this way, an important part of the sequence, such as motif, can sometimes be neglected. This is because the motif is not considered important while pair-wise alignment of two sequences is performed; however, it has significant importance in final MSA. Although the greedy approach aligns sequences progressively, it does give the best possible results while avoiding the expensive computation of comparing all the other sequences with the target sequence.

T-Coffee is a consistency-based MSA program that provides more accurate results compared to the other two methods, but with a slight compromise in computational time. In T-Coffee, at each step of progressively aligning the sequences, all the query sequences are considered, thus reducing the chance of errors in the final MSA. In the first step, a library is formed that includes the weighted global and local pair-wise alignment information of all the input sequences; in global pair-wise alignment, the whole length of the sequences is aligned; in local alignment, the top ten scoring segments that are identical within both the sequences are computed. The final step extends the previous library and utilizes the weights to get optimal results. In this step, S_1 and S_2 are first aligned through S_3 and then through S_4, and the weight for each residue pair is calculated. In this way, while aligning sequence S_1 and S_2, information from S_3 and S_4 is utilized as well. This eliminates the possibility of missing important information in the final MSA [11, 12].

2.5. MAFFT

MAFFT adopts an iterative progressive approach, like MUSCLE, and efficiently finds the homologous segments using Fast Fourier Transform (FFT). The distance matrix is computed using the 6-mer method. The major technique used by MAFFT is the FFT based group-to-group alignment algorithm. The sequences are progressively aligned using the iterative approach. However, to balance computational time and accuracy, two cycles are executed mostly for longer sequences. In the FFT group-to-group alignment algorithm, amino acids and sequences are represented in the form of vectors, and the correlation coefficient is then calculated. Homologous positions can be determined by correlation. There are three different steps in MAFFT. In the first step, namely FFT-NS-1, the distance matrix using the 6-mer method is computed. In the second step, namely FFT-NS-2, the quality of this guide tree is improved by constructing another guide tree from FFT-NS-1, along with the alignment of all the sequences. In the final step, FFT-NS-i, the iterative approach is used to improve quality. This process is repeated until no further improved results are obtained [13, 14].

3. MSA program evaluation and benchmark

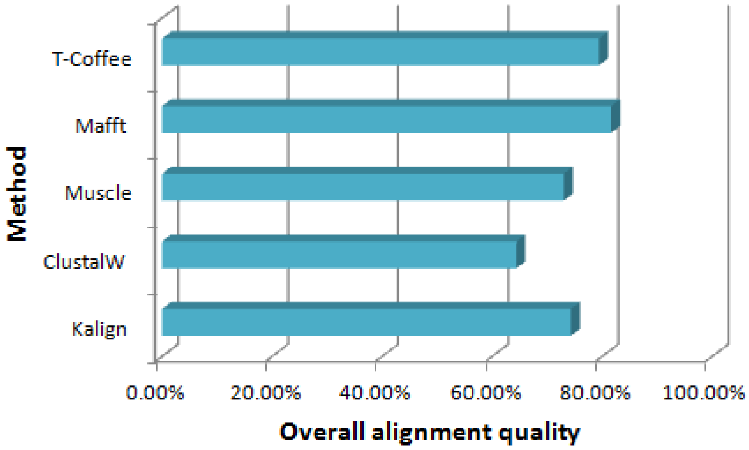

Fig. 2. Overall performance evaluations of MSA programs.

In order to evaluate the performance of the above programs, reference alignments of BAliBASE 2.01 are used in this paper. This benchmark consists of 142 reference alignments divided into five different categories. All the alignments in this benchmark are manually constructed and properly tested. In [1], equidistant sequences are included that have similar lengths. In [2], families are included that are aligned with highly distant sequences. [3] contains sequences from four different families, such that the similarity between two sequences that belong to different families is less than 25%. [4] is with N/C terminal extension, and [5] contains sequences with internal insertions [15]. The CPU execution time of the different programs is evaluated on the basis of this benchmark. In order to evaluate the overall performance of these programs, the above-mentioned test case is extended, and more reference sets are added. The new test case contains 10 reference sets with different MSA problems. This reference set of BAliBASE consists of 218 reference alignments and 17892 protein sequences [3].

The performance of MSA programs is evaluated on the basis of three aspects: CPU time, performance percentage and scalability. CPU time calculates the total amount of time required to align all the sequences in the benchmark, as shown in Figure 2. In the overall performance test, the sequences are ranked on the basis of correctness of the results of all the alignments, as shown in Figure 3. With advancements in the field of biotechnology, a huge amount of data is produced. So to

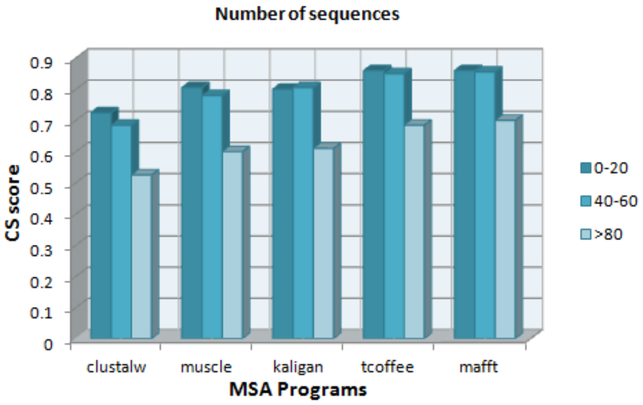

Fig. 3. Impact of number of sequences on the quality of MSA.

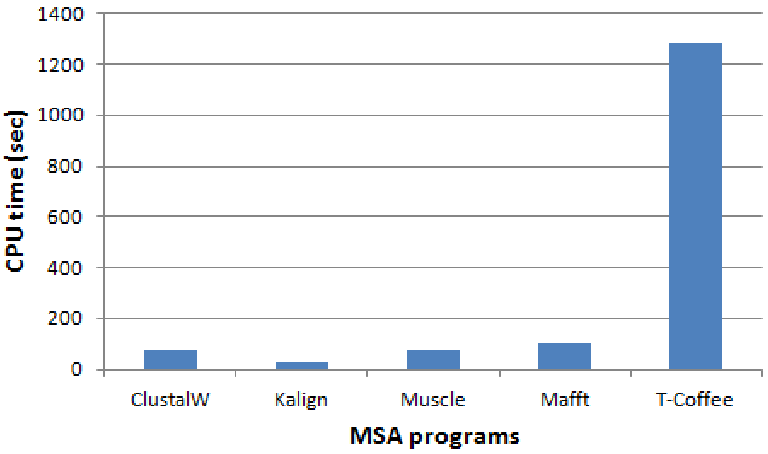

Fig. 4. Execution time of all the alignments in the benchmark.

handle and analyze this data properly and meet the current challenges, scalability is the major issue. To analyze whether the programs are scalable or not, quality of alignment is accessed with the number of sequences. An increase in the number of sequences decreases the quality of alignment, as shown in Figure 4. For the evaluation of MSA programs, data is extracted from [3, 10].

In order to find out the scalability of different programs, the MSAs of sequences with different lengths are computed. The overall performance of the alignment obtained is analyzed on the basis of the column score. The column score of an alignment is calculated by the summation of all the residue scores within a column, divided by total number of sequences for each column. For each position, if the sequence is properly aligned, a score of 1 is assigned; if otherwise, a score of 0 is assigned. The maximum a column score can be is 1, which represents perfect alignment [3].

In the last few years, different papers have been written to evaluate the performance of different MSA programs on the basis of different benchmarks. The main focus of this paper is to gather those results and evaluate their performances with reference to the algorithmic techniques they are using. This paper provides in-depth information of the five most popular programs amongst users. With different MSA programs available, the user has to choose the best possible solution (option). Their algorithmic techniques, along with the performance evaluation of these programs, will help a user choose a specific program on the basis of his objective behind performing MSA.

4. Discussion

Performance evaluation of the programs highlights many of the limitations and strengths of the different programs. The performance of most of the programs is dependent on various factors, including length of the sequences and the similarity percentage between the sequences. Even after the development of different programs, ClustalW is still one of the most-used programs. The basic reason is that it is one of the most trusted algorithms amongst users because of its consistent accuracy. After ClustalW, many programs were proposed with better speed and accuracy, however, ClustalW is still the standard program to construct MSA. The major issue in ClustalW is that errors evolved during initial alignment cannot be modified afterward. To overcome this major issue, different algorithms with iterated approaches were proposed, e.g. MUSCLE. In MUSCLE, the results are refined when continuously iterating the MSA. In most cases, when sequences increase in length or number, the number of iterations is reduced by the user to lessen the computational complexity of the program. In MUSCLE, the value of the k-mer is of real importance. If the value of the k-mer is large in the divergent sequences, it will lead to inaccurate results; and if the value of the k-mer is very small in identical sequences, then the similarity results can be ambiguous.

The second most important factor is the number of sequences. As the number of sequence increases, the performance of the programs decreases. To obtain an optimal MSA for N sequences, each of length L has a complexity of $O(L^N)$. The optimal solution is obtained using the dynamic programming algorithm, but exponential time and space scaling issues arise. In the case of progressive alignment, complexity is reduced to $O(N^2)$, and a few 1000 sequences can be aligned because of this reduced complexity. Today, many of the genome sequencing projects demand MSA of protein families containing sequence lengths greater than 50,000. Therefore, it is really challenging to obtain results in reasonable time without compromising accuracy. Therefore, scalability is a major issue in these cases. Some of the faster algorithms, including K-align, have to compromise their complexity in order to align large sequences. Although it can be extremely fast while aligning small datasets, it is not scalable. In MUSCLE, after performing MSA on 3000 sequences, the program becomes extremely slow. Most of the time, the iteration cycle is reduced to a value of 2 in order to reduce time complexity. K-align is considered one of the fastest algorithms, as it reduces the computational time required to evaluate large sequences; however, accuracy is compromised [16, 17].

The performance of all five programs is badly affected when sequences start exceeding 80, which indicates that the performance of MSA is inversely proportional to the number of sequences. This paper concludes that quality of alignment is compromised as the number of sequences increase.

The overall performance evaluation of different algorithms yields the expected results. T-Coffee takes the most time to complete the alignments, but the results obtained have a high level of accuracy. The major objective of this program is to achieve accuracy, even at the cost of computational time. Mafft offers the optimal solution of all the given sequences, which gives the most accurate solution without taking much computational time. If the main objective of a user is to find the best possible solution in minimum time, then K-align is the best option out of the five programs. When compared with other programs, Mafft works efficiently, as it generates results from huge data in a shorter period of time. On the other hand, in order to construct MSA for sensitive data, the best solution is T-Coffee or Mafft, as both of them give the most accurate results. So the reason a user is performing MSA decides the best available program based on their objectives. Table 1 summarizes the key characteristics of all the MSA programs.

Table 1
Characteristic comparison of MSA programs

MSA program	**Key algorithmic technique**	**Alignment quality**	**Computation time**
ClustalW	Progressive method	Least accurate when compared with other 5 programs	Less as compared to T-Coffee
K-align	Wu-Manber string matching for distance estimation	Some loss of accuracy as compared to Mafft and T-Coffee	Lowest
Mafft	Fast Fourier transform	Highest alignment quality	Higher than K-align but produce more accurate results
MUSCLE	Iterative method	More accurate than ClustalW because of iterative approach	Can be reduced by reducing the number of iterations.
T-Coffee	Progressive method with extended library	High alignment quality	Highest

The quality of MSA programs is continuously improving with time. The main goal of all the programs is to achieve optimum similarity in the best time and to positively deal with scalability issues. To deal with huge amounts of data in minimum time, parallel MSA with the concept of parallel computer architecture is encouraged to decrease computational time. Another challenge in constructing MSA is the identification of noise in the data. All the programs construct MSA on the basis of sequence similarities, but with the possibility that the sequence is erroneous. So the future of MSA must deal with all these issues, and improvements in programs are essential in order to acquire the best computational analysis results.

5. Conclusion

The MSA construction is the cornerstone for all subsequent computational biological analysis. An erroneous result evolved from MSA ultimately leads to false results in all subsequent analysis methods. To improve the quality of computational results, the quality of MSA programs has to be improved. Each of the different algorithmic techniques adopted to construct MSA all have strengths and limitations. None of the programs are capable of providing the best results for all the test cases. ClustalW is a well-known and credible tool, but is less accurate and scalable compared to other programs. To improve accuracy, MUSCLE uses an iterative approach, but in the case of a large number of sequences, iterations are reduced to attain results in reasonable time. T-Coffee is used when high accuracy is required, with a compromise in computation time. Mafft achieves the highest alignment quality scores, whereas K-align alignment quality is reduced to attain results in the best computational time. A user can choose the program on the basis of his objective for performing MSA. At the present time, the strengths of different programs can be integrated to find a better optimal solution.

References

[1] E.V. Koonin, Darwinian evolution in the light of genomics, Nucleic Acids Research **37** (2009), 1011–1034.

[2] K.D. Nguyen, Y. Pan and G. Nong, Parallel progressive multiple sequence alignment on reconfigurable meshes, BMC Genomics **12** (2011), doi: 10.1186/1471-2164-12-S5-S4.

[3] J.D. Thompson, B. Linard, D. Lecompte and O. Poch, A comprehensive benchmark study of multiple sequence alignment methods: Current challenges and future perspectives, PLoS ONE **6** (2011), 1-14.

[4] V.K. Sohpal, A. Singh and A. Dey, Optimization of substitution matrix for sequence alignment of major capsid proteins of human herpes simplex virus, International Journal of BioAutomation **15** (2012), 277-284.

[5] P.W. Collingridge and S. Kelly, MergeAlign: Improving multiple sequence alignment performance by dynamic reconstruction of consensus multiple sequence alignments, BMC Bioinformatics **13** (2012), 1-10.

[6] C. Notredame, Recent evolutions of multiple sequence alignment algorithms, PLoS Computer Biology **3** (2007), 1405-1408.

[7] A. Layeb, M. Selmane and M.B. Elhoucine, A new greedy randomized adaptive search procedure for multi-objective RNA structural alignment, International Journal in Foundations of Computer Science & Technology (IJFCST) **3** (2013), 9-24.

[8] G.S. Lloyd, Parallel multiple sequence alignment: An overview [Online], 2010. Available at: http://dna.cs.byu.edu/msa/overview.pdf.

[9] R.C. Edgar, MUSCLE: A multiple sequence alignment method with reduced time and space complexity, BMC Bioinformatics **5** (2004), 1-19.

[10] T. Lassmann and E.L. Sonnhammer, Kalign – an accurate and fast multiple sequence alignment algorithm, BMC Bioinformatics **6** (2005), 1-9.

[11] P.D. Tommaso, S. Moretti, I. Xenarios, M. Orobitg, A. Montanyola, J.M. Chang, J.F. Taly and C. Notredame, T-Coffee: A web server for the multiple sequence alignment of protein and RNA sequences using structural information and homology extension, Nucleic Acids Research **39** (2011), 1-5.

[12] L.M. Wallace, O. O'Sullivan, D.G. Higgins and C. Notredame, M-Coffee: Combining multiple sequence alignment methods with T-Coffee, Nucleic Acids Research **34** (2006), 1692-1699.

[13] K. Katoh, T. Miyata, K. Kuma and H. Toh, Improvement in the accuracy of multiple sequence alignment program MAFFT, Genome Informatics **16** (2005), 22-33.

[14] K. Katoh and H. Toh, Recent developments in the MAFFT multiple sequence alignment program, Briefings in Bioinfor-matics **9** (2008), 286-298.

[15] M. Zhang, W. Fang, J. Zhang and Z. Chi, MSAID: Multiple sequence alignment based on a measure of information discrepancy, Computational Biology and Chemistry **29** (2005), 175-181.

[16] F. Sievers, A. Wilm, D. Dineen, T.J. Gibson, K. Karplus, W. Li, R. Lopez, H. McWilliam, M. Remmert, J. Soding, J.D. Thompson and D.G. Higgins, Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega, Molecular Systems Biology **7** (2011), doi:10.1038/msb.2011.75.

[17] Feng Liu, Ali Goodarzi, Haifeng Wang, Joanna Stasiak, Jianbo Sun and Yu Zhou, Frontiers in biomedical engineering and biotechnology, Bio-Medical Materials and Engineering **24** (2014), 3–6.

[18] Yifei Chen, Ping Hou and Bernard Manderick, An ensemble self-training protein interaction article classifier, Bio-Medical Materials and Engineering **24** (2014) 1323–1332.