Multiple Sequence Alignments

BIOC 218 Final Project

Amy Zou

Multiple sequence alignments (MSAs) have become some of the most widely used tools in molecular biology. Alignments can show the biological relationship between the different sequences, and can give information pertinent to phylogenetic analysis, function and structure prediction, and the detection of crucial residues. Because of their importance, many models have been proposed in order to optimize the creation of these alignments. This paper will discuss some of the computational methods used to create MSAs, and analyze some important programs that use these methods.

Pairwise Alignment

Pairwise alignments are generally found using dynamic programming designed to generate a globally optimal solution. This method relies on scoring matrices such as PAM, which uses an evolutionary model of rates for mutation, and BLOSUM, which uses information gathered from families of related proteins. While dynamic programming can guarantee an optimal alignment based on the scoring method used, it is inefficient to use to generate an alignment for multiple sequences. Because the computational complexity of this method is $O(n^k)$ for k sequences of length n, it becomes prohibitively expensive after more than a couple sequences (Lipman et al 1989, Wang 1994). Creating an accurate, biologically significant sequence alignment in an acceptable amount of time is a complex problem, so a variety of different methods to accomplish this have been developed over the years.

Progressive Alignment

Many algorithms have been developed to minimize the computational complexity involved in MSA, and the most widely used approach is a heuristic known as progressive alignment. This strategy involves three major steps: First, sequences are aligned in pairs that fill a distance matrix. Second, a clustering algorithm such as UPGMA or neighbor-joining is then applied to the distance matrix to form a rooted binary guide tree. Finally,

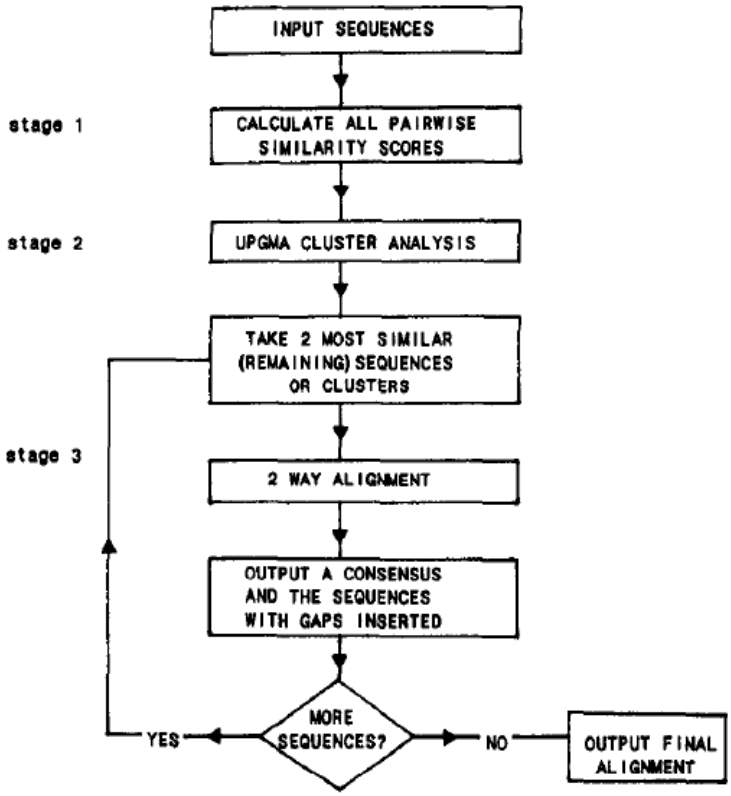

Figure 1. Clustal's strategy for forming an MSA.
(Higgins and Sharp 1988)

alignments are found using the guide tree by traveling up the tree from leaf to root, progressively adding sequences by aligning the two sequences at each node. The use of the guide tree limits the number of pairwise alignments: An alignment constructed from a guide tree of N sequences requires N – 1 pairwise merges, so the computational cost of the alignment is effectively linear for the number of sequences (Do and Katoh 2008).

The greedy nature of this heuristic is its greatest drawback. Because it looks at two sequences at a time and ignores the remaining data, it cannot guarantee an optimal solution. Mistakes made in earlier alignments are not corrected, and so they propagate throughout the process, and increasing the number of sequences increases the severity of these problems.

ClustalW: A Progressive Alignment-based MSA Program

ClustalW was first introduced in 1994, but remains a widely used alignment program. It uses the progressive alignment method (Fig. 1), but employs several methods in order to overcome the shortcomings of this approach. It uses a weighting system to correct for over - representation of extremely similar sequences, and uses position-specific gap penalties such that areas where gaps occur frequently are given lower penalties (Thompson et al 1994). In an attempt to overcome the greediness of the progressive alignment method, it delays the incorporation of the most divergent sequences until the end. For relatively similar sequences having identity above 30%, ClustalW is able to

produce reasonably accurate results quickly, but its utility is limited for more divergent sequences (Pei 2008).

Because ClustalW is so widely used, it remains a useful tool to start with when attempting to create MSAs. However, ClustalW it is less accurate and scalable than modern programs, so it is typically better to supplement with the alignments produced by other programs. Its main advantage is its comparatively low memory usage, so it is an optimal choice only in limited cases when memory size is an issue (Edgar and Batzoglou 2006).

Iterative strategy

The shortcomings caused by the greedy nature of the progressive alignment can be addressed through the iterative strategy. In this method, subsets of the aligned sequences are realigned after an initial progressive alignment. These iterations almost always improve the accuracy of the MSA (Wallace, et al 2005), and they can be repeated a set number of times or until convergence.

MUSCLE and MAFFT

MUSCLE and MAFFT are two programs that rely on iterative refinement to create a MSA with accuracy comparable or better than ClustalW. Their most significant advantage over ClustalW and other alignment process is their speed.

MUSCLE speeds up the progressive alignment method by constructing the guide tree more quickly. In contrast to ClustalW, pairs of sequences are compared and clustered based on the number of *k*mers, sub-sequences of length k, that they share. This information is then used to create the guide tree. The Kimura distance correction is applied to the generated MSA and a new tree is constructed. This is repeated at least once to further improve the tree (Fig. 2). Once the tree is fixed, MSA generation is further optimized by realignment based on different sub-trees.

MAFFT also avoids the time-costly initial pairwise alignments by detecting homologous segments using the Fast Fourier Transformation (FFT). In this method, the sequences are represented by volume and polarity values, and areas of homology have high FFT peaks. MAFFT offers several options for generating the final MSA, including FFT-NS-i, which uses tree-dependent restricted partitioning to iteratively refine the MSA. When FFT-

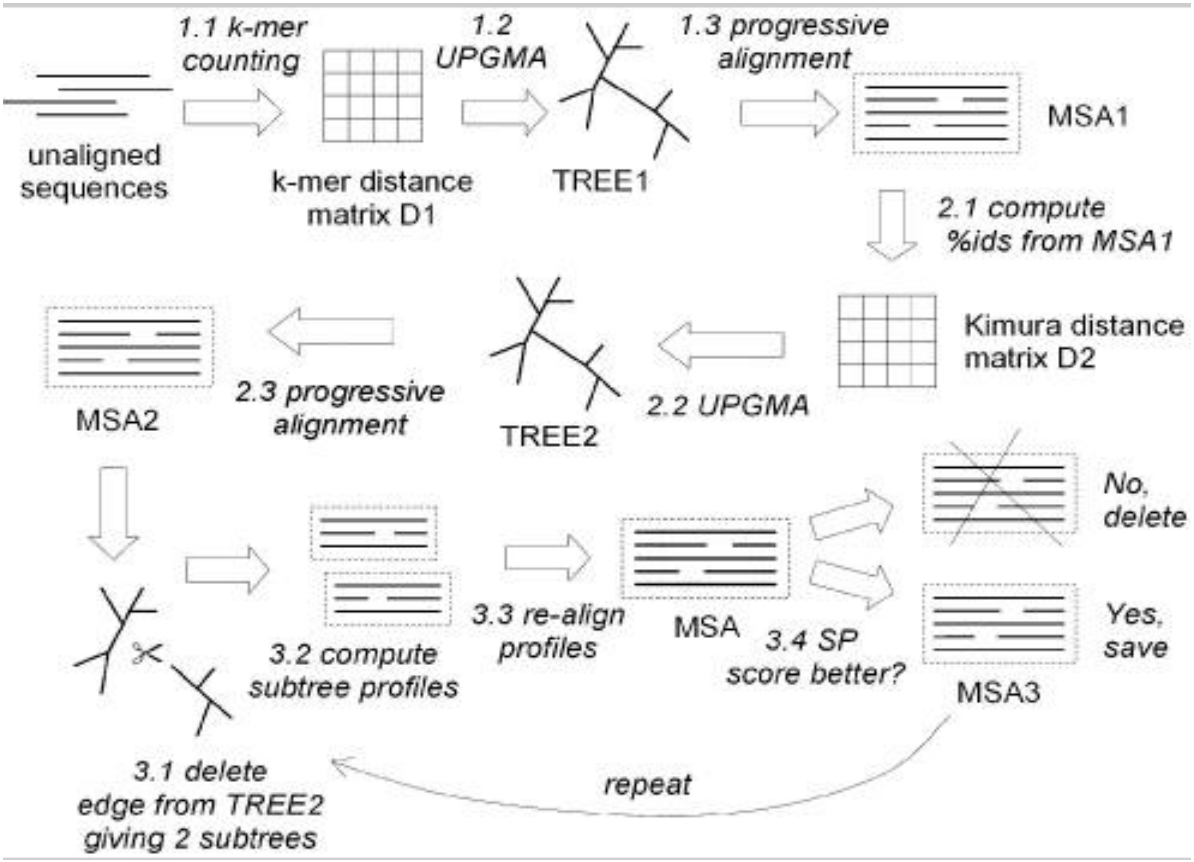

Figure 2. Three main stages behind MUSCLE's algorithm. 1. A draft progressive alignment is performed; 2. An improved progressive is performed; and 3. The MSA is refined. An MSA is available at each stage. (Edgar 2004)

NS-i is used on larger groups of sequences, it can be 100 times faster than rival programs without sacrificing accuracy (Katoh, et al 2002).

Because these two programs both avoid the time-costly dynamic programming used to generate the initial tree, both are much faster than ClustalW, yet maintain reasonable accuracy and computational cost (Edgar and Batzoglou 2006). As such, MAFFT and MUSCLE are great choices when aligning large numbers of sequences (>100). These programs also have the added advantage of being extremely flexible, and allow the user to modify the programs to run even faster, if some accuracy can be sacrificed.

Consistency - Based

Consistency-based methods attempt to overcome the limitations of progressive alignment by incorporating more information when constructing MSAs. The main problem of progressive alignment is due to its greedy nature: It takes very limited information into

account at once. Most consistency-based methods are also greedy heuristics, but still are able to incorporate more information while forming the alignment and are thus able to reach higher levels of accuracy. These methods commonly evaluate the pair-wise alignment through comparison to a third sequence: Given three sequences, A, B and C, the alignments of A-B and B-C may imply that A and C should also be aligned (Edgar and Batzoglou 2006). Since consistency-based aligners maintain this transitive nature, they are typically more accurate than regular iterative progressive aligners like ClustalW, Muscle, or Mafft. Besides increased accuracy, consistency-based methods offer the additional advantage of being able to incorporate different sources of data such as local, global, and structure-based alignments (Pei 2008). This increased accuracy comes at a significant cost to speed: consistency-based algorithms take N (number of sequences) times more CPU processing time on average (Kemena and Notredame 2009).

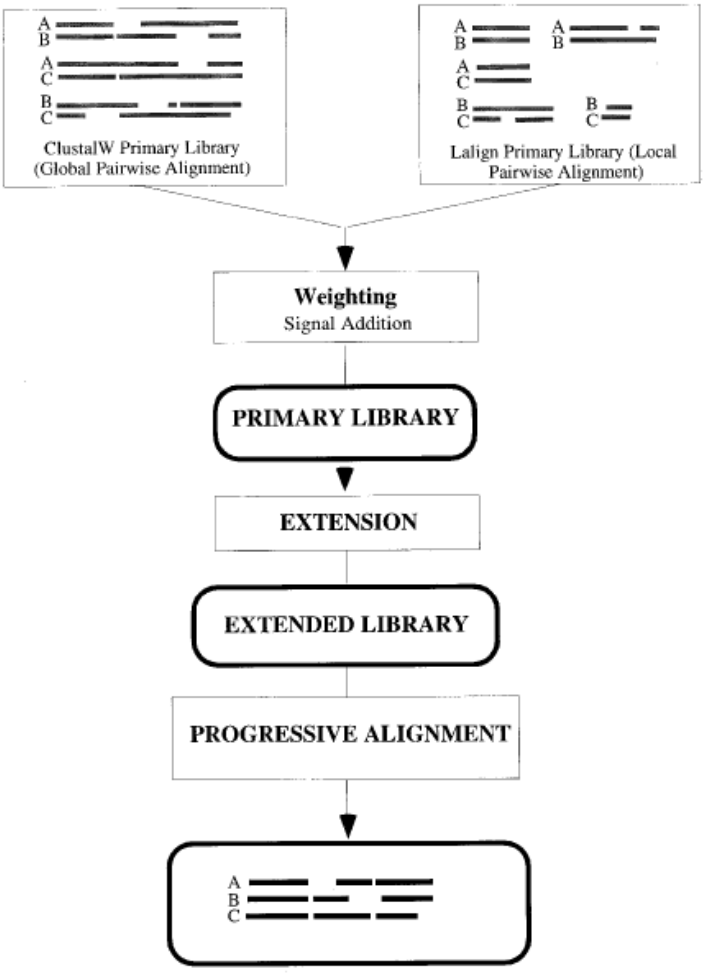

Figure 3. T-Coffee's strategy. (Notredame et al 2000)

T-Coffee

T-Coffee was the first program that combined consistency-based scoring with progressive alignment, and was a significant improvement at the time of its creation. It features two main innovations: First, it is able to use heterogeneous data sources to generate the MSA; second, it finds the MSA in a manner that considers the alignments between all pairs at a time (Notredame, et al 2000).

First, two libraries of all pairwise alignments are generated for two different sources: Lalign,

which creates sets of non-overlapping local alignments, and ClustalW. This means that the final alignment will be produced from both global and local alignment information. T-Coffee then uses a weighting system to score each pair of residues in both libraries based on the sequence identity. These two libraries are then combined, and identical pairs are merged into one extended library, with the weight of the two original weights summed together. These weights are used to generate a Neighbor Joining tree, and an alignment is produced via the progressive method (Fig 3).

T-Coffee remains as one of the best programs in terms of accuracy. The increase in accuracy is especially clear with difficult test cases, and is always evident regardless of the spread of the sequences in the tests (Notredame 2000). Because it does have a higher computation time and memory usage, however, it is not ideal for larger (>100) sets of sequences (Edgar and Batzoglou 2006). Besides its accuracy, it has the additional advantage of being able to incorporate more than one type of information, which can significantly increase accuracy if structural information is included.

ProbCons and MUMMALS

Both of these programs are similar to T-Coffee, but use a probabilistic framework instead to determine consistency. They determine the posterior probability that a pair would be aligned in each particular position. These programs then use these probabilities to generate the guide tree for progressive alignment.

ProbCons uses pair - hidden Markov models (HMM) to compute the posterior-probability matrices for every pair of sequences. Dynamic programming is then used to find the alignment that maximizes expected accuracy for each pairwise alignment. The match accuracy scores are re-estimated using a probabilistic consistency transformation. The guide tree is then created and an MSA is progressively created. The MSA is refined iteratively as many times as desired by realigning random bi-partitions of the alignment.

MUMMALS expands on the probabilistic consistency approach by using more complex HMMs. It also uses a pre-aligning step that results in groups of more divergent sequences because highly similar sequences are aligned without consistency scoring. This step allows it to better balance speed and accuracy.

Method	Structural similarity						Sequence similarity	
	DALI Z-score	GDT-TS	TM-score	3D-score	LBcona	LBconb	Sequence identity	Blosum62 score
HMM_1_1_0	0.1178	0.2510	0.3005	0.2499[a]	0.2181	0.2828	0.0953	0.1687
HMM_1_1_1	0.1200[a]	0.2519[a]	0.3010[a]	0.2514[a]	0.2190[a]	0.2838	**0.0955**	**0.1688**
HMM_3_1_1	0.1217[a]	0.2540[a]	0.3034[a]	0.2532[a]	0.2215[a]	0.2872[a]	0.0938	0.1665
HMM_1_3_1	0.1226[a]	0.2564[a]	0.3061[a]	0.2557[a]	0.2230[a]	0.2892[a]	0.0944	0.1662
HMM_3_3_1	**0.1231**[a]	**0.2570**[a]	**0.3070**[a]	**0.2563**[a]	**0.2240**[a]	**0.2909**[a]	0.0932	0.1651
ProbCons	0.1003	0.2324	0.2767	0.2307	0.2060	0.2670	**0.0983**	0.1719
MAFFT-fftnsi	0.0982	0.2333	0.2814	0.2297	0.2004	0.2632	0.0917	0.1621
MAFFT-einsi	**0.1136**	0.2425	0.2886	0.2410	0.2105	0.2763	0.0940	0.1666
MAFFT-linsi	0.1135	**0.2485**	0.2982	**0.2467**	0.2143	**0.2820**	0.0923	0.1632
MAFFT-ginsi	0.1126	0.2454	**0.2960**	0.2429	**0.2152**	0.2803	0.0972	**0.1725**
MUSCLE	0.0980	0.2297	0.2777	0.2266	0.1941	0.2535	0.0939	0.1686
ClustalW	0.0723	0.1916	0.2318	0.1876	0.1551	0.2030	0.0733	0.1344

Table 1. Assessment of multiple sequence alignment programs. The first five methods are MUMMALS, using different HMMs. MUMMALS' best score and the best score of the other programs are bolded. With this test, MUMMALS did statistically better with $p < 0.01$. (Pei and Grishin 2006)

Both of these programs achieve statistically significant improvements in accuracy over other leading methods (Do et al 2005), with MUMMALS achieving a slightly higher scores (Table 1). They are also flexible: ProbCons offers several options to increase accuracy by repeating consistency replication and iterative refinement steps; MUMMALS has the option to use a more complex HMMs to increase accuracy in exchange for a slower alignment. As these are consistency - based methods, however, they are computationally expensive and are not recommended for large alignment problems (>100 sequences) (Edgar and Batzoglou 2006).

While the alignment programs explored thus far in this paper vary in performance for different tests, it is impossible to predict which program will function best for a specific dataset. All of these methods also perform poorly on sequences with similarity below the "twilight - zone" of identity less than 20% (Pei 2008). Meta - methods such as M-Coffee attempt to address these problems by combining several methods into one MSA. While M-Coffee creates alignments better than any of the individual alignment programs on most of the considered datasets, the improvement is very small - at a few percent (Wallace 2006). Furthermore, M-Coffee is unable to solve the problem of poor performance on remote homologs, suggesting that it may not be possible to reach significantly better alignment using only sequence information.

Template - Based Methods

Since alignment methods are still unable to align remote homologs, the best alternative is to use additional information to increase the accuracy of predictions. Template - based methods refers to improving a sequence by using templates such as 3D structure or any sort of profile or prediction. Typically, these templates fall into two categories: structural or homology extension.

Structural extension involves using the Protein Data Bank (PDB) to find homologs of pairs of sequences. These PDB templates can then aligned with each other or the original sequences to generate an MSA (Fig 4). Generally, structure is more conserved than sequence, so the addition of structural information should provide a more biologically significant MSA. Homology extension uses the same principles, but uses profiles rather than structures. Aligners that use this extension replace each sequence with a homolog profile, typically generated using PSI-BLAST. The use of both structural and homology templates results in increased accuracy in all cases (Kemena and Notredame 2009).

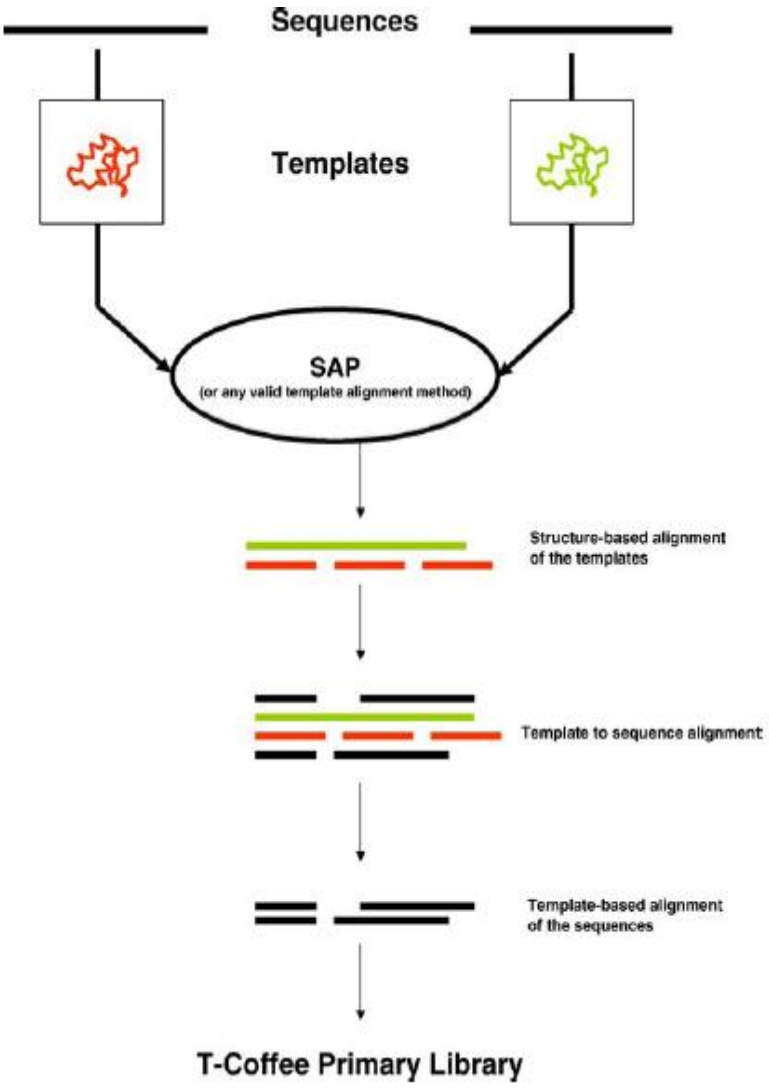

Figure 4. Framework of a Template-Based Method. (Notredame 2007)

Expresso/3DCoffee

Expresso/3DCoffee is a server that uses structural information produce the MSA. It is essentially an expansion on the consistency-based heuristic that T-Coffee uses. The pairwise alignments in Expresso, however, are generated

by using a BLAST search to identify templates from the PDB. The criteria for template selection limits selection to close homologs by requiring a minimum of 60% identity with the source sequence and at least 70% of the source residues matched (Armougom 2006). Once every input sequence has an assigned template, it applies several alignments to the data. For each pair, the global and local alignments are generated using the Needleman-Wunsch method and Lalign respectively. Structure - structure alignment is performed by LSQman, which uses rigid body superposition iteratively to find an optimal superposition, or SAP, which computes the alignments. In pairs where only one structure is known, sequence - to - structure alignment is performed by FUGUE, which turns the structure into a position-specific substitution matrix. When the library is created, the MSA is created using T-Coffee's method.

With the addition of just one structure, the alignment produced by Expresso shows increased accuracy, and accuracy increases proportionally to the amount of structural information provided (Poirot et al 2004). In distantly related sequences, Expresso shows a linear correlation between accuracy and the ratio of structures to sequences. The alignment is also relatively fast, and share's T-Coffee's flexible nature to potentially include any structural analysis method (O'Sullivan 2004). The high accuracy, however, must be taken with a grain of salt as the references themselves are often created in a similar way.

PRALINE and SPEM

PRALINE can optimize alignments through a variety of methods. First, it can use global or local alignment, or the PSI-PRALINE methods to create a homology - extended alignment. The PSI-PRALINE strategy uses PSI-BLAST on each sequence to create the profiles that are later used to progressively generate the MSA. Second, it can integrate secondary structure information by using the Protein Data Bank (PDB). If this information is unavailable, then PRALINE can use one of seven (PHDpsi, PROFsec, SSPRO 2.01, YASPIN, PSIPRED, JNET and PREDATOR) methods to predict the secondary structure and use these predictions in the profiles instead. Finally, PRALINE can also include iterative refinement to further improve the alignments.

SPEM focuses on the creation of pairwise alignments. It also uses PSIBLAST to search for homologous sequences to create profiles. Then, secondary sequence information

Methoda	ClustalW	T-Coffee	MUSCLE 6.0	ProbCons	SP^{2}b	SPEMc
Superfamily (462)	49.9	54.8	54.8	56.2	65.7	67.0
Twilight (236)	21.9	27.4	26.3	29.2	43.5	43.9
All (698)	40.4	45.5	45.2	47.1	58.2	59.2
P-valued	5.6×10^{-112}	1.5×10^{-96}	2.3×10^{-85}	3.8×10^{-83}	8.3×10^{-6}	—

Table 2. Alignment accuracies given from several methods on SABmark 1.63 benchmark. Spem shows considerable improvement on sequences in the "twilight" set. (Zhou and Zhou 2005)

or predictions are used to apply secondary structure dependent gap penalty values. A consistency - based scoring method is used in order to further refine the pairwise alignments. From there, a tree is created, generating the final alignment.

Both of these programs use homology and the structure extension to improve the alignments generated. The PSI-PRALINE method increases accuracy compared to other alignment programs such as T-Coffee and MUSCLE on average, and has an even greater increase for sequences of low homology (Simossis and Heringa 2005). The addition of secondary structure information to both basic PRALINE and PSI-PRALINE also shows improvement. SPEM also shows improvement with remote homologs on a variety of benchmarks, while remaining statistically indistinguishable from ProbCons and MUSCLE when aligning sequences with >30% similarity (Table 2).

While the quality of the alignments produced by the PRALINE and SPEM are extremely accurate, they are both computationally expensive: PSI-BLAST must be ran for each sequence, and SPEM utilizes a consistency - based scoring method. These two programs are best used with smaller numbers of diverse sequences.

PROMALS

PROMALS is a method that uses the probabilistic consistency-based scoring that was developed with ProbCons, but improves it by including predicted secondary structure and homolog information. PROMALS follows a similar method as MUMMALS, and has a first step in which highly similar sequences are aligned quickly (Fig 5). In the second alignment stage, PSI-BLAST is used to search for homologs, with hits of <20% identity removed and up to 300 hits selected (Pei and Grishin 2007). For each pair, profiles are developed from the PSI-BLAST alignment and predictions of secondary structure created by PSIPRED. It

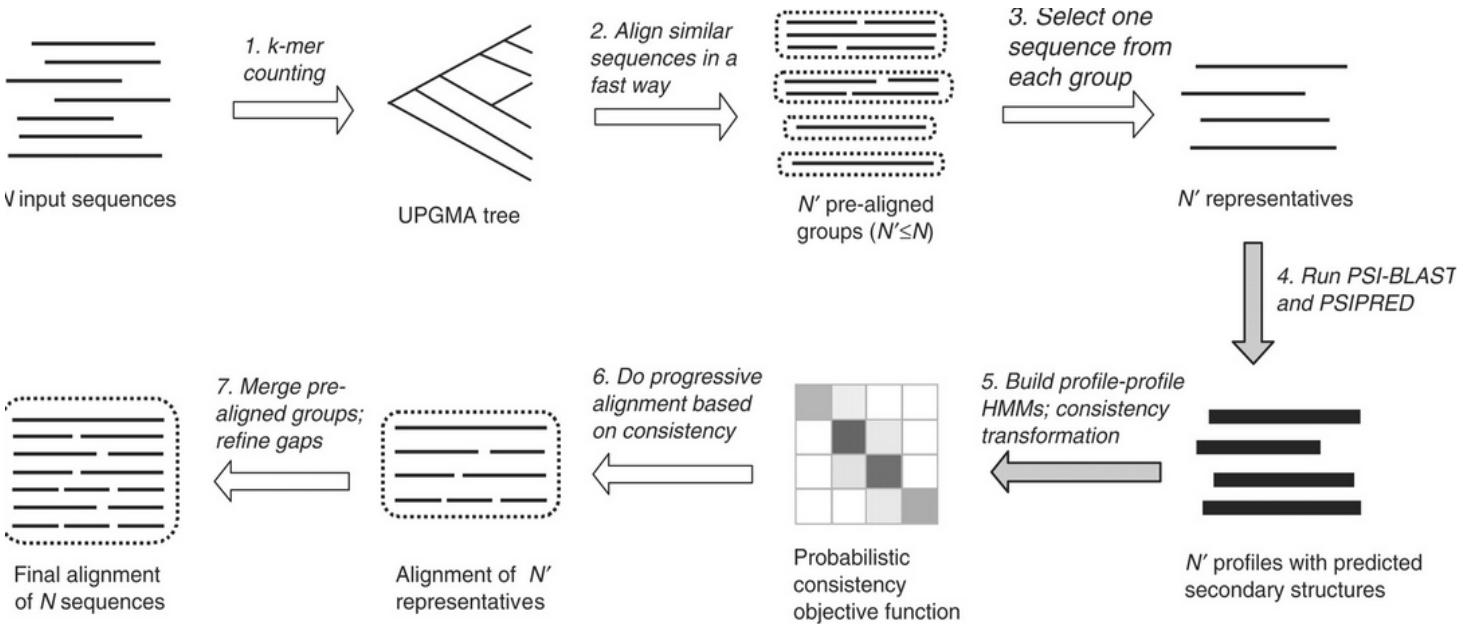

Figure 5. PROMALS procedure. The gray arrows show the most time-consuming steps: Running PSI-BLAST and PSIPRED and running the consistency transformation. (Pei and Grishin 2007)

uses a profile-profile HMM that creates the matrix of posterior-probabilities that is needed for probabilistic consistency-based scoring to create the MSA.

Compared to the best alignment methods that rely solely on sequence information, PROMALS is up to 30% more accurate, with the most improvement in highly divergent homologs (Table 3). While PROMALS also shows some improvement over SPEM and other template-based programs, it is still is unable to provide the best alignment each time, suggesting that alignments can vary greatly when performed on divergent sequences. Since PROMALS spends a great amount of time collecting homolog and structural information, it runs very slowly, taking around half an hour while stand-alone programs finish their alignments in less than a minute (Pei and Grishin 2007).

Method	SCOP[a] 0–10% (355)	SCOP[a] 10–15% (432)	SCOP[a] 15–20% (420)	SABmark-twi (209)	SABmark-sup (425)	PREFAB[c] (1682)
PROMALS	**0.435/0.457**	**0.612/0.619**	**0.761/0.772**	**0.391**	**0.665**	**0.790**
SPEM	0.377/0.411	0.558/0.578	0.727/0.751	0.326	0.628	0.774
HHalign[b]	0.406/–	0.567/–	0.730/–	–	–	0.787
MUMMALS	0.151/0.329	0.335/0.520	0.586/0.732	0.196	0.522	0.731
ProbCons	0.116/0.290	0.294/0.486	0.536/0.701	0.166	0.485	0.716
MAFFT-linsi	0.116/0.301	0.262/0.500	0.495/0.707	0.184	0.510	0.722
MAFFT-ginsi	0.116/0.308	0.265/0.497	0.496/0.714	0.176	0.495	0.715
MUSCLE	0.139/0.262	0.293/0.452	0.507/0.661	0.136	0.433	0.680
ClustalW	0.136/0.210	0.270/0.357	0.482/0.565	0.127	0.390	0.617

Table 3. Evaluation of alignment methods. For each data set, PROMALS yields statistically higher accuracy (bold numbers) than any other method (P-value <0.000001). (Pei and Grishin 2007).

Conclusions

MSAs will continue to be important in the future, and tools to create them will continue to improve. Some of the recent trends include the ability to process many sequences rapidly, combine methods, and most importantly, incorporate more information. Methods such as MAFFT and MUSCLE allow for the processing of many sequences rapidly. Meta-methods that can combine different methods in one framework and reduce the amount of work needed on the user's part. As additional structural information becomes available, template-based methods will continue to be important, and will become better able to combine more data and continue to raise accuracy, especially with distantly-related sequences. As computational abilities increase, consistency-based scoring methods can be expanded to examine even more of the data at a time, and increase accuracy further. As more research is done on MSAs, programs will be able to better and better approximate true biological relationships.

References

Armougom, et al. Expresso: automatic incorporation of structural information in multiple sequence alignments using 3D-Coffee. Nucleic Acids Res. 2006, 34(Web): W604–W608.

Do, Chuong B, and Katoh, Kazutaka: Methods in Molecular Biology 2008, 484(4): 379-413

Do, et al. ProbCons: Probabilistic consistency-based multiple sequence alignment. Genome Res. 2005, 15: 330-340

Edgar, Robert C: MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res. 2004, 32(5):1792-7.

Edgar and Batzoglou: Multiple sequence alignment. Current Opinion in Structural Biology 2006, 16(3): 368-73.

Gong, et al: Performance assessment of protein multiple sequence alignment algorithms based on permutation similarity measurement. Biochemical and Biophysical Research Communications 2010, 399: 470–474.

Katoh, et al: MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. Nucleic Acids Research 2002, 30(14): 3059-66.

Kemena and Notredame: Upcoming challenges for multiple sequence alignment methods in the high-throughput era. Bioinformatics 2009, 25(19): 2455–2465

Higgins and Sharp: CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. Gene. 1988 , 73(1):237-44.

Lipman DJ, Altschul SF, Kececioglu JD: A tool for multiple sequence alignment. Proc Natl Acad Sci USA 1989, 86:4412-4415.

Notredame, Cedric: Recent Evolutions of Multiple Sequence Alignment Algorithms. Public Library of Science 2007, 3(8): 1405 - 8.

Notredame, et al: T-Coffee: A Novel Method for Fast and Accurate Multiple Sequence Alignment. Journal of Molecular Biology 2000, 302 (1): 205-217.

Nuin et al: The accuracy of several multiple sequence alignment programs for proteins. BMC Bioinformatics 2006, 7:471.

O'Sullivan, et al: 3DCoffee: Combining Protein Sequences and Structures within Multiple Sequence Alignments Journal of Molecular Biology Volume 340, Issue 2, 2 July 2004, Pages 385-395.

Pirovano, Walter and Heringa, Jaap: Multiple Sequence Alignment. Methods in Molecular

Biology 2008, 452(2): 143-161.

Pei, Jimin: Multiple protein sequence alignment. Current Opinion in Structural Biology 2008, 18:382–386.

Pei and Grishin: MUMMALS: multiple sequence alignment improved by using hidden Markov models with local structural information. Nucleic Acids Res. 2006 September; 34(16): 4364–4374.

Pei and Grishin: PROMALS: towards accurate multiple sequence alignments of distantly related proteins. Bioinformatics 2007, 23(7): 802-808.

Poirot, et al. 3DCoffee@igs: a web server for combining sequences and structures into a multiple sequence alignment. Nucl. Acids Res. 2004 32 (suppl 2): W37-W40.

Thompson, et al: CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Research 1994, 22(22):4673-80

Simossis and Heringa: PRALINE: a multiple sequence alignment toolbox that integrates homology-extended and secondary structure information. Nucleic Acids Research 2005 33 (suppl 2): W289-W294.

Wallace,I.M. et al: Evaluation of iterative alignment algorithms for multiple alignment. Bioinformatics 2005, 21:1408–1414.

Wallace, et al: M-Coffee: combining multiple sequence alignment methods with T-Coffee. Nucleic Acids Research 2006, 34(6): 1692-1699.

Wang L, Jiang T: On the complexity of multiple sequence alignment. J Comput Biol 1994, 1:337-348.

Zhou, Hongyi and Zhou, Yaoqi: SPEM: improving multiple sequence alignment with sequence profiles and predicted secondary structures. Bioinformatics 2005 21(18): 3615-21.