Protein Multiple Sequence Alignment: Benchmarks and Comparison

Roham Gharegozlou
Biochemistry 218 Final Project
March 16 2009

Introduction

The aim of protein multiple sequence alignment (MSA) is to assemble amino acid sequences in alignments that reflect their biological relationship, whether evolutionary, structural, functional, or a combination of the three (Notredame 2007). MSA is utilized in the analysis of functionally similar proteins to identify regions of homology or common motifs, or to identify putative motifs in newly characterized sequences, providing a hint towards biological function (Aiyar 2000).

Pairwise alignments – alignments of two amino acid sequences – usually use dynamic programming to arrive at a mathematically optimal alignment based on estimates of evolutionary rates of mutation and conservation given in the PAM and BLOSUM matrices, including penalties for insertions and deletions of amino acid residues that cause divergence between the sequences. Extrapolating these methods of dynamic programming to more than a few sequences at a time is computationally expensive and infeasible (Wang and Jiang 1994, Elias 2006, Pei 2008). Given the immense computational demands, a series of Multiple Sequence Alignment (MSA) programs have been created to speed the process as well as derive more accurate estimates employing various techniques of heuristics; programmed algorithms designed to arrive at reasonable but inexact estimates of homology and alignment between sets of sequences.

This report aims to give an overview of the important considerations when choosing a program for MSA in order to help the biologist navigate the broad differences between algorithms. I will begin by outlining some recent developments in the field, and subsequently focus on illuminating the different benchmarks used to evaluate the performance of MSA programs. I will end the report with individual analysis of each MSA program’s strengths and drawbacks.

Table of Contents

Different Approaches

Extrapolating the methods of dynamic programming in pairwise alignment to more than a few sequences at a time is computationally expensive and infeasible (Wang and Jiang 1994, Elias 2006, Pei 2008). Mathematically, for *k* protein sequences of *n* amino acids each, the computational capacity and space requirements would be n^k, which quickly approaches impossibility for increasing numbers of ever-longer sequences (Brutlag 2007). As a result, deriving evolutionary trees from sequence relationships is an approximate process. Given that rarely are sequences from evolutionary ancestors available, all phylogenetic relationships are inferred from homologies in present-day sequences, also known as observed taxonomic units (OTUs). Because alignments can be evolutionary, structural, functional, or a combination of the three, the role of the biologist is critical and leads to much more refined and biologically-relevant results than fully automated algorithms. Therefore a major goal in developing new tools for MSA has been to increase the biological significance of results through the incorporation of any and all additional information available, especially in order to improve the alignment of sequences whose similarity is below the “Twilight Zone” of <20% identity (Pei 2008).

Most MSA packages employ the progressive algorithm first popularized by Clustal: this algorithm involves first estimating a guide tree from the given sequences based on sequence similarity, then progressively aligning sequences along each level of the guide tree using pairwise alignment. Considerable effort has been put into developing accurate scoring schemes for the pairwise alignment algorithm, and current MSA programs utilize one of two types of scoring schemes: matrix- or consistency-based. The former method takes into account a more limited set of data, only considering residues within the immediate locale of the position being aligned, while consistency-based methods incorporate a wider dataset, including the alignment with third-sequences not in the immediate pair being compared (see Figures 4-5). Matrix-based alignment is used by programs such as ClustalW (Thomson 1994) and MUSCLE (Edgar 2004), while consistency-based alignment was developed in T-Coffee (Notredame 2000) and expanded in more recent programs such as PCMA (Pei 2003), ProbCons (Do 2005), MUMMALS (Pei 2006), and MAFFT (Katoh 2005) (reviewed in Notredame 2007). As will be discussed in more detail below, consistency-based methods are generally shown to be more accurate, but take *N* times longer, where N is the number of amino acid sequences to be analyzed (Blackshields 2006).

Furthering the goal of biological accuracy, a new suite of programs has emerged that incorporates structural as well as sequence information in MSA. These include PRALINE and SPEM as well as Expresso, an update to the popular consensus-based method TCoffee. Nevertheless, the venerable choices for protein MSA, such as ClustalW, TCoffee, and ProbCons, still prove popular especially when structural information cannot be found.

Choosing a Program

Perhaps most important from the point of view of most users of MSA programs are the considerations in choosing among the multitude of algorithms currently available. The three main considerations are biological accuracy of alignments, computational time, and computational memory usage. Given the ever-increasing capacity of modern computers, usually biological accuracy is the primary concern. This is not straightforward to measure; benchmark suites have been developed that test accuracy of predictions against a library of families of known homology based on 3D structure. Top-scoring MSA programs are led by ProbCons, followed by MAFFT, MUSCLE, and TCoffee, but the exact distribution differs based on similarity of test sequences as well as the benchmark dataset used. Newer versions of MAFFT include consistency-based scoring and rival the accuracy of ProbCons (Edgar and Batzoglou 2006).

Nevertheless, the statistically "best" program is rarely optimal in every single circumstance. Within benchmarks, programs based on different algorithms show distinct strengths in particular areas: though ProbCons outperforms Clustal W by an average of 5%, on certain tests ClustalW outperforms ProbCons by 9%. Because it is as of yet not possible to predict the method that will work best on a given set of sequences, it is generally suggested that multiple programs based on different algorithms be used, and the results compared for discrepancies (Edgar and Batzoglou 2006, Pei 2008). Similarly, in evaluating the performance of MSA programs, different types of benchmarking software should be used in order to present diverse challenges.

Though earlier dismissed, demands of computational power further highlight the need for different and diverse MSA programs: while consensus-based programs show very high accuracy, they are computationally intensive. Therefore, highly accurate programs such as TCoffee and ProbCons often are unable to handle more than 100 sequences without memory problems on typical computers (Edgar and Batzoglou 2006). In comparison, MAFFT and MUSCLE are high-throughput methods that can handle large sets of sequences with accuracies comparable to ClustalW: the progressive algorithm MUSCLE-p can align 5000 sequences of average length 350 in 7 minutes on an average desktop computer (Edgar 2004).

Benchmarking methods for Multiple Sequence Alignment

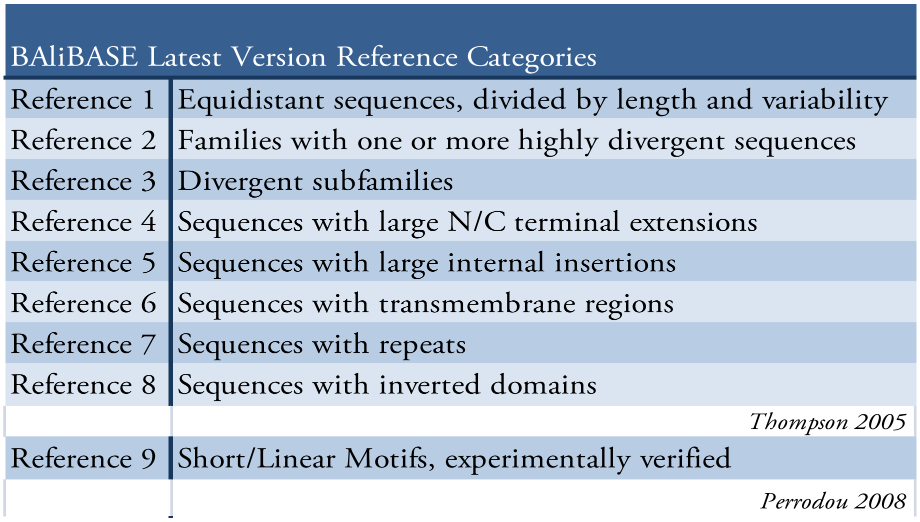
When MSA programs were first developed, limited availability of accurate information as well as processing power meant performance of algorithms was demonstrated based on a small number of example sequences, called gold standards. Given that the benchmark tests are freely available, developers of new programs could specifically tune their programs to attain artificially high scores, reducing the test's applicability. Newer programs such as BaliBASE (Thompson 1999) incorporate larger datasets, making it far more difficult for developers to target the performance of their algorithms. Some programs use available 3D structural data to test sequence alignments (HOMSTRAD – Mizuguchi 1998, OxBench – Raghava 2003, PREFAB – Edgar 2004), while others have employed probalistic models to simulate evolution of sequences and generate test MSAs on the fly (Lassmann and Sonnhammer 2002). Some databases are manually refined (BAliBASE, HOMSTRAD), while others are predominantly automated (PREFAB, OxBench).A short review of common benchmarking methods aimed primarily at global sequence alignment is included below. To evaluate local alignment in MSA programs, IRMBASE (Subramanian 2005, not discussed) is often utilized.

BAliBASE

BAliBASE was one of the first large-scale benchmarks developed specifically for testing and comparison of multiple sequence alignment programs. The alignments are based on protein 3D structure superposition, and are "manually refined" to ensure a higher alignment quality than purely automated methods (Thompson 2005). Only reliably aligned regions are annotated as core blocks; distant regions with limited structural homology are discarded as their inclusion would require arbitrary alignment that would lead or be construed as bias (Thompson 1999). The original BAliBASE was divided into four references, but this has since expanded to eight, reflecting the increased quantity and complexity of information. Each reference contains alignments organized in order represent real MSA problems faced by biologists (cf. Table 1). Version 3.0 of BaliBASE also expanded the number of sequences from 1444 to 6255 and provided a more user-friendly interface for the program (Thompson 2005). A past criticism of BaliBASE was the absence of full-length sequences and thus bias in favor of global alignment methods (Lassmann and Sonnhammer 2002), but this has since been rectified: BaliBASE now includes full-length sequences (Blackshields 2006). The software also includes a semi-automatic update protocol, allowing the program to keep pace with new data on protein sequences and families as they become available (Thompson 2005).

Most recently, the developers of BAliBASE have added a ninth reference set, to aid in the performance evaluation of alignments of short unstructured motifs, termed linear motifs (Perrodou 2008). The majority of protein multiple sequence alignment programs are designed for the identification and alignment of globular protein domains; longer lengths of sequences with predicted functions based on structure and amino acid residues. Their performance is benchmarked using 3D structure superpositioning. These programs are not ideal for the analysis of non-globular proteins without a defined structure, or unstructured parts of larger proteins; while unstructured regions can be simple linkers between peptide domains, in which case the amino acid sequence doesn't matter, often unstructured regions play an important part in the biology of the protein. They can contain functional domains such as protein-interaction sites, cell compartment targeting signals, and sites for post-translational modification or cleavage; large parts of insulin receptor substrates are unstructured for example, as is the entirety of Tau, a brain protein implicated in the pathology of Alzheimer's Disease. Linear Motifs (LM) are typically between 3 and 10 amino acids in length, and have degrees of variability in their sequences, requiring quite a high level of sensitivity to distinguish LM patterns from background noise. The new reference set includes only experimentally-verified verified functional motifs, extracted from the Eukaryotic Linear Motif (ELM – Puntervoll 2003) database, refined and confirmed manually (Perrodou 2008).

Among BAliBASE's most touted advantages is its "expertly refined" and highly accurate database (Thompson 2005). However the reliance on expert validation introduces elements of subjectivity and therefore bias (Blackshields 2006), as well as increasing the time and labor required for maintenance and expansion of the database. The search for fully automated processes led to OXBench, PREFAB, HOMSTRAD, and SABmark, discussed below.



BAliBASE Latest Version Reference Categories	
Reference 1	Equidistant sequences, divided by length and variability
Reference 2	Families with one or more highly divergent sequences
Reference 3	Divergent subfamilies
Reference 4	Sequences with large N/C terminal extensions
Reference 5	Sequences with large internal insertions
Reference 6	Sequences with transmembrane regions
Reference 7	Sequences with repeats
Reference 8	Sequences with inverted domains
	Thompson 2005
Reference 9	Short/Linear Motifs, experimentally verified
	Perrodou 2008

Table 1 – BAliBASE Latest Version – Reference Categories

OxBench

OxBench's data set incorporates the 3Dee database of protein structural domains, which contains definitions for proteins of experimentally-determined three dimensional structures from the Protein Data Bank (PDB) up to July 1998, about 729 domain families and 9,015 domains. These are then filtered: low-resolution structures, domains with less than 40 amino acid residues, single-member families, domains with uncertain secondary structure, and structures with low stereochemical quality as assessed by PROCHECK are all removed from the data set, as are highly similar domains, which provide limited information. Taking out multiple segment domains leaves 1,1168 domains in 218 families. Structural similarity is assayed as automatically as possible through the STAMP multiple structure comparison algorithm (Russell 1992), cases on the high or low extremes of similarity were inspected by human experts and the algorithm adjusted. Domains are then organized into three datasets: Master, Full, and

Extended. The Master set comprises the core sequences with known structure, divided into subcategories based on sequence identity and similarity. The Full set adds available full-length sequence data, while Extended adds similar sequences extracted from the SWALL sequence database (Raghava 2003). See Figure 1 below for a schematic representation. It has been reported that some tests in the Extended set are too large for the more computationally-intensive programs: in a recent study TCoffee could only align 235/276 test cases from the Extended set, while Align-m could only align 107 (Blackshields 2006).

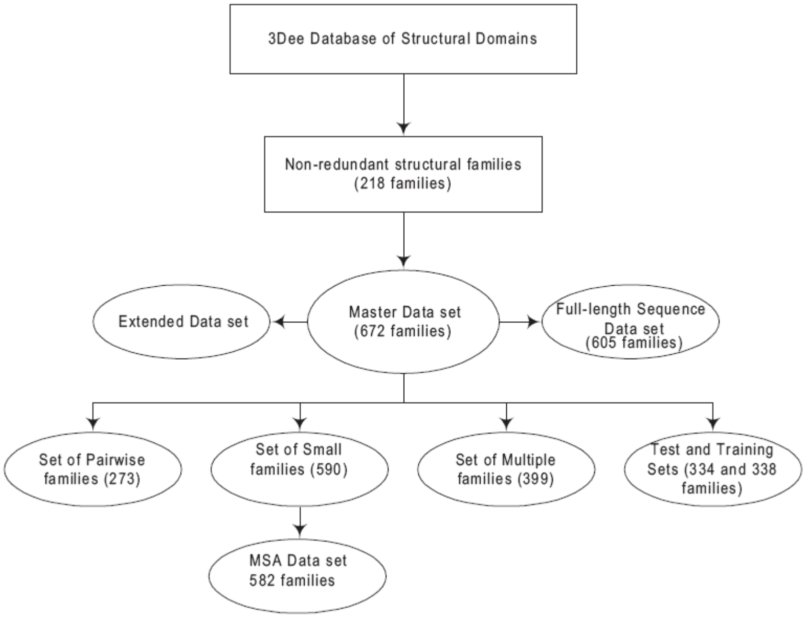

3Dee Database of Structural Domains
Non-redundant structural families (218 families)
Extended Data set
Master Data set (672 families)
Full-length Sequence Data set (605 families)
Set of Pairwise families (273)
Set of Small families (590)
Set of Multiple families (399)
Test and Training Sets (334 and 338 families)
MSA Data set 582 families

Figure 1 – Flowchart outlining relationships between OXBench datasets and subsets. Source: Raghava 2003

HOMSTRAD

HOMSTRAD incorporates structural information from the Protein Data Bank (PDB) and SCOP as well as sequence information from Pfam, a manually compiled database of homologous protein sequence familieis. SCOP is based on structures from the PDB, classified by experts into hierarchical categories (Stebbings and Mizuguchi 2004). The original HOMSTRAD incarnation (Mizuguchi 1998) only comprised of the structural information from the PDB; integration with the Pfam sequence database was added in 2001, expanding the relevance of the benchmark. All HOMSTAD sequences were scored against Pfam profile hidden Markov models; highly-scoring homologous sequences were aligned with the HOMSTRAD structures using FUGUE (de Bakker 2001). The newest version of HOMSTRAD stores core family information in a MySQL database, allowing for increased flexibility in updating the structural database as more data becomes available as well as faster access times and lower memory requirements (Stebbings and Mizuguchi 2004). While not explicitly intended to benchmark MSA programs, it is often used for this purpose (Wallace 2006, Blackshields 2006, Pei 2008).

PREFAB

PREFAB was developed in order to reduce the need for expert intervention in the database creation process of such databases as BAliBASE and provide a fully automated protocol for benchmarking the accuracy of multiple sequence alignment programs. First, pairs of proteins are aligned structurally without incorporating sequence information. A query is then sent to a database for each sequence to obtain close homologues, and the entire set of resultant proteins is aligned using an MSA method. The structural alignment accuracy was assayed using test sets from the FSSP database (Holm and Sander 1998) as well as realigned structures using the Combinatorial Extension (CE) aligner (Shindyalov and Bourne 1998); only pairs that had considerable agreement between the two methods were kept, in order to reduce "questionable and ambiguous structural alignments" (Edgar 2004). Subsequently, the full-chain sequence of each structure was used to launch a PSI-BLAST search of the NCBI non-redundant protein sequence database, hits were filtered down to 80% maximum identity between pairs, and 24 homologous proteins selected at random. Thus the original pair and their 24 homologous sequences are combined into sets of <50 sequences; this number is arbitrary and is only limited by the computational power available, which needs to be considerably high for the more demanding MSA algorithms such as TCoffee. The accuracy of the alignment is tested only between the original pair of sequences (Edgar 2004).

SABmark

The Sequence Alignment Benchmark (Van Walle 2005) focuses only on dissimilar sequences, with very low to intermediate similarity (0-50% identity). In practice, the performance of most programs converges with increasing sequence similarity and offers little opportunity for improvement (Sauder 2000, Blackshields 2006), and so SABmark illuminates the region of highest difference between programs. Sequences are grouped into two alignment sets: Twilight Zone, with sequences of very low similarity, and Superfamilies, with sequence of low similarity. Sequences are matched with high-quality structures from the SCOP database (Murzin 1995), continually updated to reflect new releases. (Von Walle 2005)

Overview of Global MSA Programs

ClustalW

Still the most popular method for multiple sequence alignment, ClustalW was first introduced in 1994. Clustal W was released as an update to the venerable Clustal program; the 'W' stands for 'weights' and is homage to the program's distinguishing characteristic. While previous MSA algorithms incorporated single-weight matrices applying equal phylogenetic divergence to all sequences, ClustalW incorporated the Neighbor Joining weighting method to assign heavier weights to divergent sequences than more redundant ones, correcting for reduced sampling at more distant evolutionary distances, correcting for what is known as the parameter choice problem (Thomson 1994). This increases the sensitivity of the test greatly (Thomson 1994), but such weighted algorithms will tend to overweight erroneous sequences, suggesting biological relationships when in fact there are none (Brutlag 2007). Clustal W uses gaps to optimize alignment, and introduced position-specific gap penalties in order to reflect the biological significance of gaps in areas important structurally or functionally: gaps in regions of limited functional or structural consequence incur a lower penalty than gaps in important folding regions. Clustal W, similar to more recent progressive algorithm techniques, first generates a guide tree based on sequence similarity as calculated through pairwise alignment using the BLOSUM distance matrix, then assigns the weights as described above, weighting more distant relationships. The program then progressively aligns pairs of sequences at each node of the tree incorporating the position-specific weights and gap penalties. This "greedy" property of the algorithm means that if an error is made in the early stages of the alignment, its effect is compounded at every level of the tree (Thomson 1994, Blackshields 2006, Brutlag 2007).

Since its introduction over 14 years ago, limited improvements and modifications have been made to the ClustalW algorithm, and it has been surpassed by modern algorithms in terms of either speed, accuracy, or both (Edgar and Batzoglou 2006). ClustalX is an updated version of the original program with limited algorithmic improvements but with a new graphical user interface and various usability improvements (Thompson 1997). Lack of program updates has not stopped development of new versions of the software to allow operation on various platforms: while Clustal was originally written for MS-DOS, Clustal-W and -X have recently been rewritten in C++. ClustalX version 2.0 boasts increased compatibility with newer OS's, a streamlined process for future improvements to the algorithm and code, and cessation of reliance on the NCBI's *Vibrant* toolbox, which was used to develop the original Clustal X graphical user interface but is no longer supported (Larkin 2007). Part of ClustalW's enduring success is certainly due to its low memory requirements, despite that when compared with modern programs, ClustalW is both less accurate and less scaleable (Edgar and Batzoglou 2006).

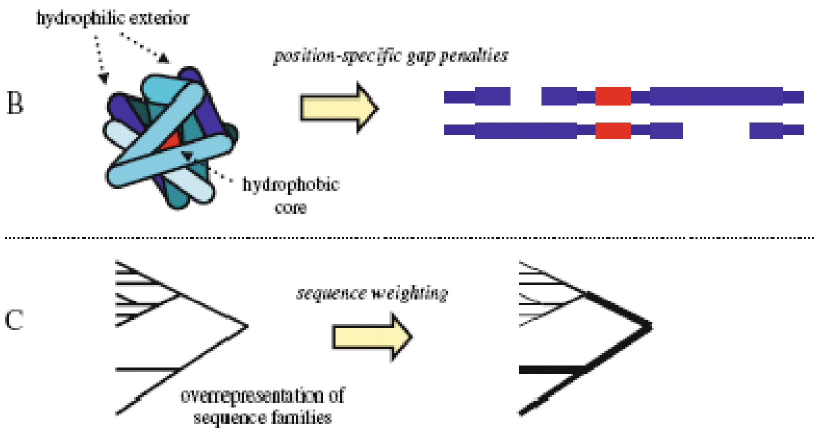

B
hydrophilic exterior
position-specific gap penalties
hydrophobic core
C
sequence weighting
overrepresentation of sequence families

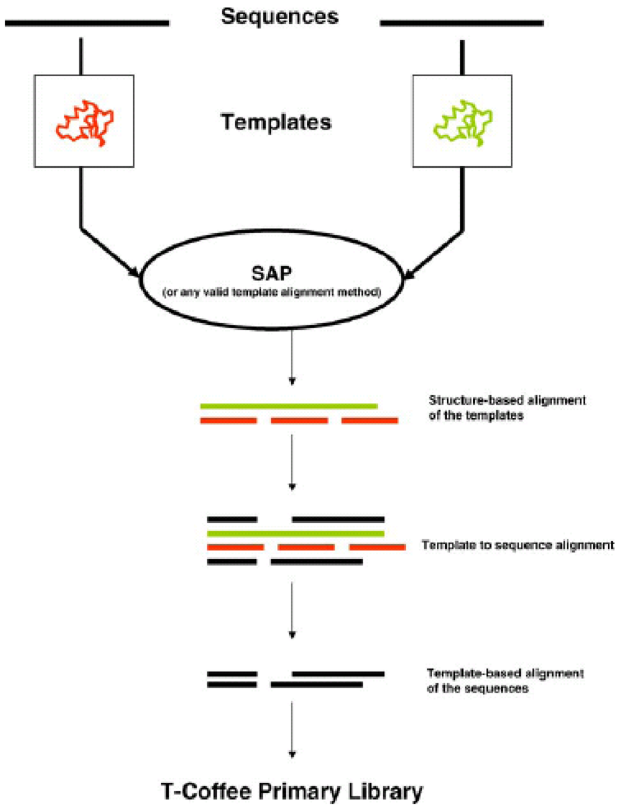
Figure 2 – B: there are relatively more alignment gaps in the hydrophilic exterior than the hydrophobic interior of globular proteins: position-specific gap penalties are higher for regions with hydrophobic residues and lower for regions with hydrophilic residues. C: Figure illustrating the increased weight applied to underrepresented families in weighted MSA such as ClustalW. (Do and Katoh 2008)

T-Coffee

Similar to ClustalW, T-Coffee first uses weighted pairwise comparisons of sequences to compile an initial primary library. TCoffee combines local and global alignment techniques: local alignments are created using Lalign and a global alignment using ClustalW. TCoffee is a consistency-based algorithm: alignments are based both on the sequences to be aligned, pair by pair, as well as on how all the sequences align with each other. TCoffee includes in a pairwise alignment between sequences A&B the information from a third sequence C: for each matched residue *x* in A and *y* in B, there is an alignment with a residue *z* in C. The alignments A-C and B-C together constitute an alignment A-B; these scores are taken into account when building the initial guide tree. (Pirovano and Heringa 2008). Figure 3, Figure 4 and Figure 5 show the development of the initial library as well as its extension using alignments with a third sequence as described above.

The library is re-weighted using its own consistency and used as a position-specific substitution matrix to carry out progressive MSA, through a similar mechanism as ClustalW and other progressive methods. In particular, a neighbor-joining distance matrix tree is computed in order to guide progressive pair-wise alignments leading to an overall MSA (Notredame 2000).

TCoffee has among the highest accuracy of modern MSA methods, but is computationally intensive and becomes impractical at alignments involving more than 100 sequences (Edgar and Batzoglou 2006). Compared to ProbCons, with typically higher accuracy scores, TCoffee has the important advantage of incorporating extensions, in the form of 3DCoffee/Expresso and M-Coffee, both discussed separately below, under Structure-Based Methods and Consensus Methods, respectively.


Sequences
Templates
SAP
(or any valid template alignment method)
Structure-based alignment of the templates
Template to sequence alignment
Template-based alignment of the sequences
T-Coffee Primary Library

doi:10.1371/journal.pcbi.0030123.g002

Figure 3 –Process of formation of TCoffee's primary library: structural templates are first identified, mapped onto sequences, and aligned using a template alignment method such as SAP. This template is then used to guide the alignment of the original sequences leading to the final MSA. Source: Notredame 2008

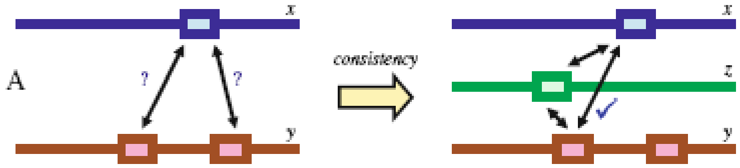


Figure 4 – Figure illustrating the use of consistency-based scoring, as in TCoffee. Source: Do & Katoh 2005.

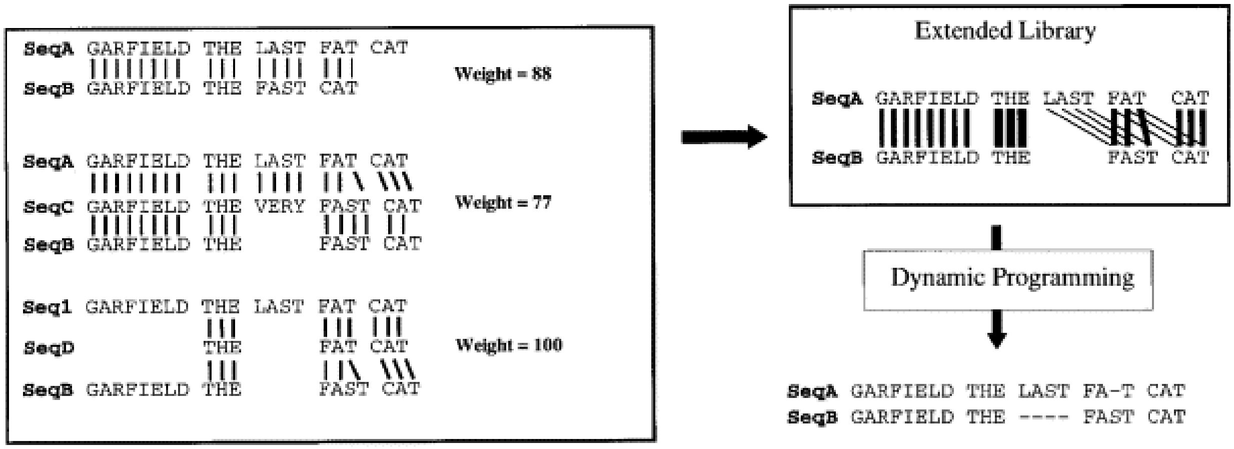

SeqA GARFIELD THE LAST FAT CAT
SeqB GARFIELD THE FAST CAT
Weight = 88
SeqA GARFIELD THE LAST FAT CAT
SeqC GARFIELD THE VERY FAST CAT
SeqB GARFIELD THE FAST CAT
Weight = 77
Seq1 GARFIELD THE LAST FAT CAT
SeqD THE FAT CAT
SeqB GARFIELD THE FAST CAT
Weight = 100
Extended Library
SeqA GARFIELD THE LAST FAT CAT
SeqB GARFIELD THE FAST CAT
Dynamic Programming
SeqA GARFIELD THE LAST FA-T CAT
SeqB GARFIELD THE ---- FAST CAT

Figure 5 – Diagram showing TCoffee's use of third sequences to refine the library used as a guide tree for dynamic programming as compared to the regular progressive alignment strategy. Source: Notredame 2003

ProbCons

This program often obtains the highest accuracy scores on benchmarks (Blackshields 2006), but suffers from the same drawbacks as TCoffee: high computational requirements, because both are based on a consistency-based approach (Edgar and Batzoglou 2006). ProbCons can both be downloaded or utilized as web server application. ProbCons starts by calculating a posterior probability for each residue match within a pairwise comparison, using a pairwise Hidden Markov Model (HMM) and expectation maximization (EM). From this the program obtains the alignment that maximizes the "expected accuracy," and similar to TCoffee uses third sequences to further refine its alignment. Dynamic programming is then used to drive at the final pairwise alignment, which is incorporated into a guide tree and used in a progressive protocol to compute the final MSA.

While computationally intensive, ProbCons incorporates flexibility in its approach: several steps in the computation are iterative, and the user can increase iterations to drive at a more refined guide tree and therefore final MSA. These include the initial calculation of posterior probability, from a default 2 iterations up to a maximum of 5, iterative refinement of alignments, from a default of 1000 to a max of 1000, and pretraining expectation maximization to optimize gap penalties to closely relate to biological significance. (Pirovano and Heringa 2008)

In various tests, ProbCons has been shown to be the most accurate single algorithm (Blackshields 2006, Essoussi 2008, Pei 2008), but can be outperformed by certain structure-based algorithms such as Expresso (Armougom 2006), or consensus-based programs such as M-Coffee (Wallace 2006) that integrate various different approaches.

MAFFT, MUSCLE, and MUSCLE-P

These programs are typically faster and more accurate than ClustalW, and offer a good tradeoff between accuracy and computational demand as compared to ProbCons and TCoffee. Adding to the versatility, MUSCLE can be operated through the web server as well as a downloadable application.

MUSCLE avoids the time-consuming dynamic programming involved in the creation of the guide tree important in ClustalW or TCoffee, instead employing iterative refinement procedures that produce high quality alignments at much higher speeds. First, sequences are ordered according to continuous amino acid segments of length *k* they share with other sequences in the set, these are termed *k-mers*. UPGMA is used to calculate the guide tree, and a MSA is created following the tree's progression. This MSA is used to construct a new tree through an iterative mechanism, applying the Kimura distance correction multiple times until no further improvement is seen. This final MSA is used to refine and drive at the last alignment; the program iterates through each node of the tree, repeating the alignment problem using each updated MSA, keeping the one with the highest alignment score (Pirovano and Heringa 2008).

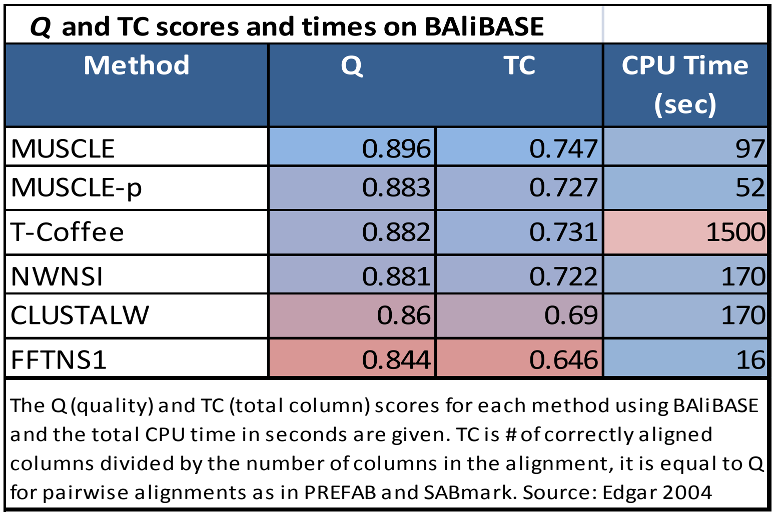
Furthermore, MAFFT and MUSCLE are both scaleable; the user can reduce accuracy in favor of speed for high-throughput applications (Edgar and Batzoglou 2006). The user can decide whether include or exclude individual stages outlined above, can give the program a given time constraint within which to make its alignment, place limits on the iteration of *k-mer*-based realignment, or use “anchor optimization” by dividing an alignment problem into vertical blocks and aligning each column separately with its own associated profile (Pirovano and Heringa 2008).

Consensus-Based Global MSA

M-Coffee

M-Coffee (Wallace 2006) draws information from a combination of MSA programs and uses T-Coffee to drive at a consensus that incorporates every one. Similar to the process used in incorporating 3D-structural information into the TCoffee library in 3D-Coffee, M-Coffee generates entries in the library from different MSA packages. Weights are assigned across the packages based on four different elements: variance/covariance, Altchul Carrillo Lipman (ACL), Thompson Higgins Gibson (THG), and accuracy (ACC) weights. Variance, covariance, and accuracy is determined based on HOMSTRAD reference alignments. ACL and THG are tree-based methods and assign higher weights to objects closer to the root of the tree; closer objects provide better estimates of the root. The result is that more accurate methods, as measured by these four criteria, count more towards the final MSA than less accurate methods.

M-Coffee combining the top eight individual programs was found to outperform all individual methods on every category of HOMSTRAD and Prefab (more than 1400 total), as well as 6/10 BAliBASE reference sets. On average, M-Coffee is twice as likely to deliver most accurate MSAs than the best individual methods (Wallace 2006). For full results see Table 5.

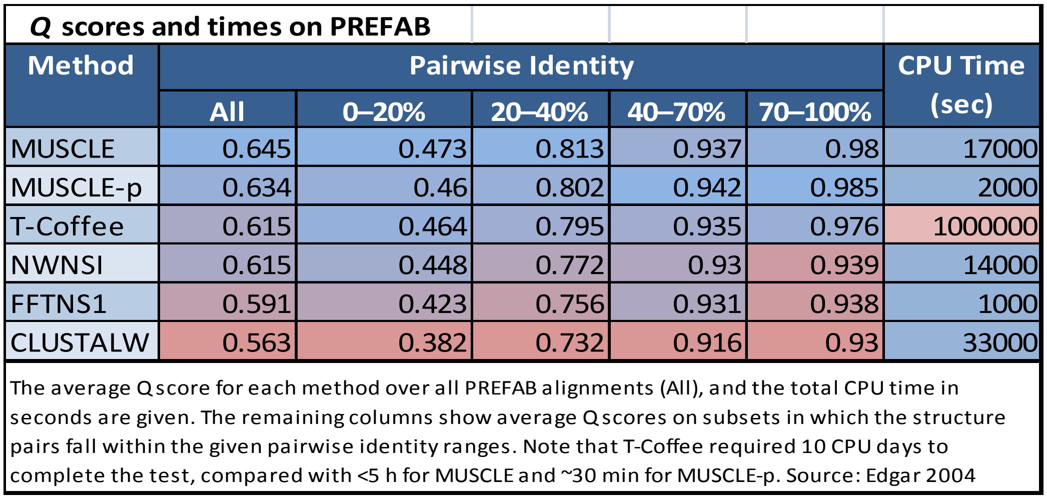


Q and TC scores and times on BAliBASE

Method	Q	TC	CPU Time (sec)
MUSCLE	0.896	0.747	97
MUSCLE-p	0.883	0.727	52
T-Coffee	0.882	0.731	1500
NWNSI	0.881	0.722	170
CLUSTALW	0.86	0.69	170
FFTNS1	0.844	0.646	16

The Q (quality) and TC (total column) scores for each method using BAliBASE and the total CPU time in seconds are given. TC is # of correctly aligned columns divided by the number of columns in the alignment, it is equal to Q for pairwise alignments as in PREFAB and SABmark. Source: Edgar 2004

Table 2

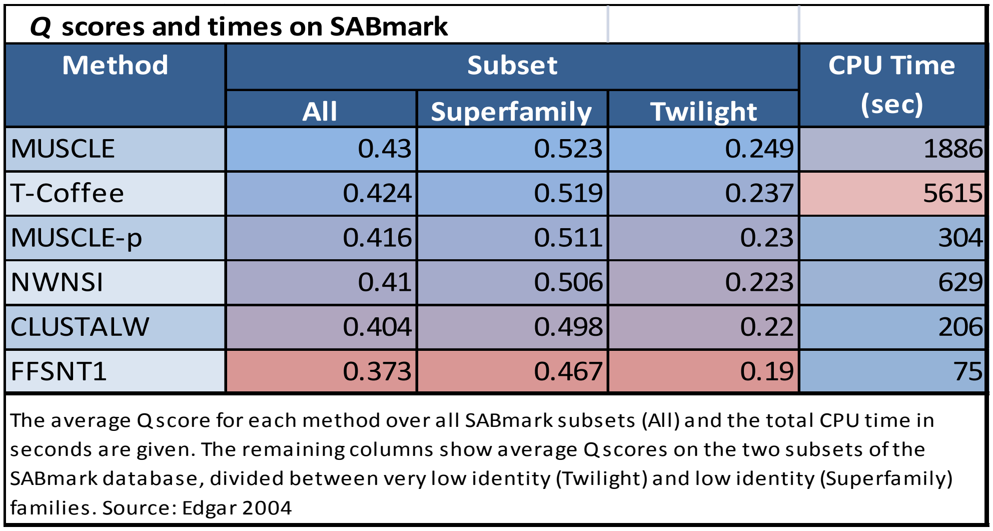


Q scores and times on PREFAB

Method	Pairwise Identity					CPU Time (sec)
	All	0–20%	20–40%	40–70%	70–100%	
MUSCLE	0.645	0.473	0.813	0.937	0.98	17000
MUSCLE-p	0.634	0.46	0.802	0.942	0.985	2000
T-Coffee	0.615	0.464	0.795	0.935	0.976	1000000
NWNSI	0.615	0.448	0.772	0.93	0.939	14000
FFTNS1	0.591	0.423	0.756	0.931	0.938	1000
CLUSTALW	0.563	0.382	0.732	0.916	0.93	33000

The average Q score for each method over all PREFAB alignments (All), and the total CPU time in seconds are given. The remaining columns show average Q scores on subsets in which the structure pairs fall within the given pairwise identity ranges. Note that T-Coffee required 10 CPU days to complete the test, compared with <5 h for MUSCLE and ~30 min for MUSCLE-p. Source: Edgar 2004

Table 3



Q scores and times on SABmark

Method	Subset			CPU Time (sec)
	All	Superfamily	Twilight	
MUSCLE	0.43	0.523	0.249	1886
T-Coffee	0.424	0.519	0.237	5615
MUSCLE-p	0.416	0.511	0.23	304
NWNSI	0.41	0.506	0.223	629
CLUSTALW	0.404	0.498	0.22	206
FFSNT1	0.373	0.467	0.19	75

The average Q score for each method over all SABmark subsets (All) and the total CPU time in seconds are given. The remaining columns show average Q scores on the two subsets of the SABmark database, divided between very low identity (Twilight) and low identity (Superfamily) families. Source: Edgar 2004

Table 4

Table 5

	M-Coffee8	ClustalW	Dialign-T	FINSI	Muscle 6	PCMA	POA	Probcons	T-Coffee
Homstrad	67.75	61.15	57.92	64.22	66.04	63.73	51.9	66.41	65.37
Prefab <10%	27.19	18.25	15.51	24.86	24.14	25.53	9.09	24.81	23.41
Prefab 10 to <20%	59.8	43.27	44.11	58.76	54.76	55.96	32.26	56.21	55.28
Prefab 20 to <30%	84.58	74.79	75.28	83.76	82.09	81.47	64.42	82.85	82.39
Prefab 30 to <40%	92.54	87.27	85.62	91.81	90.42	89.84	79.96	91.68	91.51
Prefab 40 to <100%	97.05	94.91	96.07	96.92	96.17	95.03	94.3	96.2	96.68
Prefab total	72.91	61.68	62.05	72.01	69.56	69.76	52.61	70.54	69.97
BaliBase Set: 11	43.18	22.68	25.32	38.95	34.37	37.45	11.18	39.55	32.68
BaliBase Set: 12	85.91	71.43	72.57	82.68	84.8	82.61	51.05	84.8	83
BaliBase Set: 20	43.12	21.68	29.2	45.85	36.49	44.83	13.37	37.78	39.68
BaliBase Set: 30	59.19	25.48	35.19	57.59	41.04	58.15	7.89	47.26	47.48
BaliBase Set: 40	58.17	39.04	44.75	60.02	48.42	53.83	14.42	51.25	55.58
BaliBase Set: 50	59.81	33.69	44.25	57.69	50.56	59.88	21.63	55.25	57.31
BaliBase Set: S11	59.5	40.76	33.34	50.63	59.37	44.76	31.37	58.45	47.61
BaliBase Set: S12	86.59	79.05	76.2	84.02	86.95	82.91	68.14	87.05	83.75
BaliBase Set: S2	56.76	44.37	36.9	53.85	55.78	51.85	35.24	54.46	49.78
BaliBase Set: S3	69.41	49.69	47.31	63.83	63.14	64.1	36.14	65.03	64.45
BaliBase Set: S5	60.6	43.27	45.47	57.73	60.33	56.73	28.47	59.8	55.67
BaliBase total	62.02	42.83	44.59	59.34	56.47	57.92	29	58.24	56.1

Comparison of popular MSA programs. Source: Wallace 2006

Structure-based Global MSA

Expresso and 3D-Coffee

The functionality of Expresso (Armougom 2006) was accessible as early as 2004 via 3D-Coffee, an extension of the TCoffee engine (O'Sullivan 2004). The original 3D-Coffee interface required the user to input explicit structures to be considered along with sequences whereas Expresso uses a BLAST search of the PDB database to automatically locate suitable structural templates, greatly simplifying and streamlining the alignment process (Armougom 2006).

Structural information was incorporated into TCoffee using three methods: Fugue, SAP, and LSQman. 3D-Coffee submits sequence/structure pairs to the Fugue server[1] and automatically retrieves the corresponding pair-wise alignments. SAP pair-wise alignments are computed based on non-rigid structure superposition whereas LSQman on the other hand computes pairwise alignments based on rigid structure superposition. The SAP and LSQman alignments are highly accurate, so much so that 3D-Coffee assigns the highest possible weight to them, but only the threading method Fugue is able to incorporate single structures, i.e. when structural information is only available for one out of the pair of sequences being compared. (O'Sullivan 2004).

When combined with the Fugue threading algorithm given a single structure, TCoffee-Fugue (TC-Fugue) outperformed TCoffee by 4% and ClustalW by over 8% in MSA of a set of 39 related proteins; improvements increased with higher ratios of structures:sequences. With two structures, TC-LSQman and TC-SAP could be compared; results yielded the same direct 4% improvement for TC-Fugue, a slightly higher 5% improvement for TC-LSQ, a considerable 8.5% improvement over TC for TC-SAP and a 10.3% direct improvement for the consensus method incorporating all three methods, TCoffee-3D. Induced improvement, which is the improvement in accuracy of sequence alignments other than the protein whose structure was provided, followed a similar patter, with the combination method TC-3D having the highest accuracy, and both methods showed modest increases in improvement with more structures incorporated. Of note is the small and at times non-significant induced improvements, meaning that these methods are not able to extrapolate the structural information of a given sequence to neighboring sequences; the relationship between structures and sequences is not one of diminishing returns but is linear (O'Sullivan 2004).

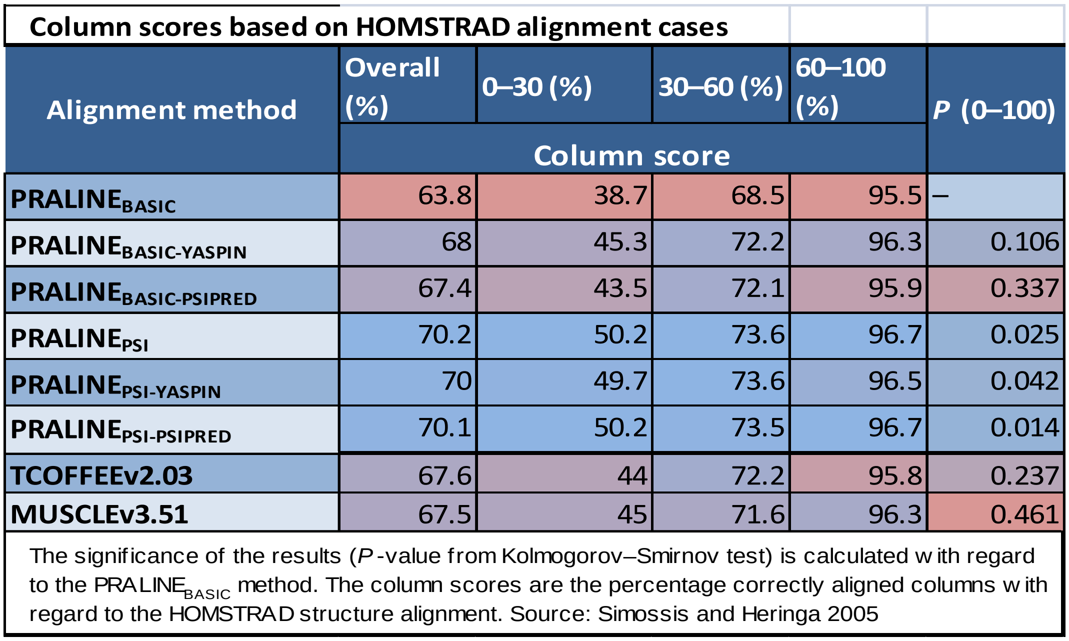
[1] http://www.cryst.bioc.cam.ac.uk/˜fugue/

PRALINE and SPEM

PRALINE (Simossis and Heringa 2005) and SPEM (Zhou and Zhou 2005) are two different programs that use PSI-BLAST to search for homologous sequences and build a structural profile for each sequence, to aid in computing the final MSA, taking into account predicted secondary structure as well as sequence homology. For proteins without PDB structures, PRALINE can incorporate DSSP predicted secondary structure information, or, when no such information is available, a choice of seven secondary strcutre prediction methods to arrive at a hypothesized structure (Pirovano and Heringa 2008)

Both PRALINE and SPEM are important MSA programs in their own right, but their distinguishing characteristic is incorporation of structure information. SPEM is a rather inflexible program in terms of user-specification, but follows a similar algorithm to PRALINE when incorporating structural information: the difference is that SPEM uses dynamic programming to apply structure-dependent gap penalties, while PRALINE adds the use of structure-specific residue exchange matrices. Similar to both TCoffee and ProbCons, both PRALINE and SPEM make considerable use of consistency-based scoring (see)(Pirovano and Heringa 2008)

Table 6 is a comparison using the HOMSTRAD benchmark set, showing considerable improvements over TCoffee and MUSCLE when PRALINE is combined with PSI-BLAST as well as prediction methods PSIPRED and YASPIN. The gains are more pronounced at lower identity levels, as is expected (Simossis and Herringa 2005). PRALINE and SPEM's use of an external program (PSI-BLAST) greatly increases running time and processing power; these programs are primarily useful for limited numbers of proteins (Edgar and Batzoglou 2006).



Column scores based on HOMSTRAD alignment cases

Alignment method	Overall (%)	0–30 (%)	30–60 (%)	60–100 (%)	*P* (0–100)
	Column score				
PRALINE$_{BASIC}$	63.8	38.7	68.5	95.5	–
PRALINE$_{BASIC-YASPIN}$	68	45.3	72.2	96.3	0.106
PRALINE$_{BASIC-PSIPRED}$	67.4	43.5	72.1	95.9	0.337
PRALINE$_{PSI}$	70.2	50.2	73.6	96.7	0.025
PRALINE$_{PSI-YASPIN}$	70	49.7	73.6	96.5	0.042
PRALINE$_{PSI-PSIPRED}$	70.1	50.2	73.5	96.7	0.014
TCOFFEEv2.03	67.6	44	72.2	95.8	0.237
MUSCLEv3.51	67.5	45	71.6	96.3	0.461

The significance of the results (*P*-value from Kolmogorov–Smirnov test) is calculated with regard to the PRALINE$_{BASIC}$ method. The column scores are the percentage correctly aligned columns with regard to the HOMSTRAD structure alignment. Source: Simossis and Heringa 2005

Table 6 – Column scores based on HOMSTRAD, comparing PRALINE, TCoffee, and MUSCLE

Conclusion and Future Directions

Performance Differences between MSA Programs

As is clear from the tables presented above, there can be significant differences between programs in the accuracy of their determined multiple sequence alignment. In particular, especially at lower levels of identity, consistency-based and structure-based methods handily outperform classic methods such as ClustalW, and tend to modestly outperform iterative methods such as MUSCLE. ProbCons is generally the winner in most accuracy-based tests (Blackshields 2006, Essoussi 2008). More drastic are the differences between programs in processing time: while the most accurate algorithms are marginally more accurate than the average, they often require orders of magnitude more time and processing power.

Future Directions

The trend of including all available data in determining sequence homology will continue: as more structural and functional information is made available, and more computing power increases the ability to handle complex algorithms, programs that incorporate more and more data will be able to more closely approximate true biological relationships. There will be also be better utilization of phylogenetic relationships and incorporation of models of protein sequence evolution. Existing data will be verified and corrected, preventing uncertainties in structural data to cloud sequence alignment. Furthermore, there will be a trend away from progressive methods towards considering all sequences at once, clearly an impossible computational problem today, but one that is being solved step-by-step through methods such as consistency-based alignment.

The Biologist's Choice

Recent developments have included the development of "consensus methods" such as M-Coffee (Wallace 2006) as well as template-based methods such as Expresso (Armougom 2006). M-Coffee uses a combination of MSA programs to arrive at a consensus that incorporates every one, and is shown to be considerably more accurate than the most accurate single program (Wallace 2006). Given that each program has its own strengths and weaknesses, M-Coffee is an easy way for the biologist to apply widely different perspectives to the same problem. Structure-based methods such as Expresso take advantage of the fact that structures evolve slower than sequences, meaning that even when sequences have diverged beyond recognition, structural-based homology based on 3D comparisons of protein folding can identify evolutionary relationships. This means mistakes in structural alignment would impact sequence alignment negatively, thus the role of the supervising biologist is paramount.

The final choice of program hinges on the biologist's criteria: if a fast response time is required, or if the sequence bank being analyzed is large, then MUSCLE and MAFFT are the best starting points. If the goal is accuracy without compromise, then a joint analysis using the best heuristics algorithm, ProbCons, as well as all available structure data, as in Expresso, is the best way forward. According to this analysis, there is limited scope for continuing to use ClustalW other than as a reference; other algorithms outperform it both in terms of speed and in terms of accuracy. TCoffee is a highly accurate program but its computational requirements are too onerous: reports of slow running times, failed tests and incomplete benchmark results are rife in the literature (Edgar 2004, Blackshields 2006).

References

Aiyar A. The use of Clustal W and Clustal X for multiple sequence alignment. Methods Mol Biol. 132:221-241, 2000

Armougom F, Moretti S, Poirot O, Audic S, Dumas P, Schaeli B, Keduas V, Notredame C. Expresso: automatic incorporation of structural information in multiple sequence alignments using 3D-Coffee. Nucleic Acids Res. 34:604-608. July 1 2006

Blackshields G, Wallace IM, Larkin M, Higgins DG. Analysis and comparison of benchmarks for multiple sequence alignment. In Silico Biol 6:321-339. 2006

Brutlag D. Biochemistry 218: Computational Molecular Biology. Stanford University. Course lectures 2008-2009.

de Bakker PI, Bateman A, Burke DF, Miguel RN, Mizuguchi K, Shi J, Shirai H, Blundell TL. HOMSTRAD: adding sequence information to structure-based alignments of homologous protein families. Bioinformations applications note. 17(8):748-749, April 6 2001

Do CB, Katoh K. Functional proteomics: protein multiple sequence alignment. Humana Press (2008) 379-413

Edgar RC. MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res. 32(5):1792-1797, March 19 2004

Elias I. Settling the intractability of multiple alignment. J Comput Biol. 13(7):1323-1339. Sep 2006

Essoussi N, Boujenfa K, Limam M. A comparison of MSA tools. Bioinformation. 2(9):452-455, 2008

Holm L, Sander C. Touring protein fold space with Dali/FSSP. Nucleic Acids Res., 26: 316–319. 1998

Larkin MA, Blackshields G, Brown NP, Chenna R, McGettigan PA, McWilliam H, Valentin F, Wallace IM, Wilm A, Lopez R, Thompson JD, Gibson TJ, Higgins DG. Clustal W and Clustal X version 2.0. Bioinformatics. 23(21):2947-2948. September 10 2007

Lassmann T, Sonnhammer EL. Quality assessment of multiple alignment programs. FEBS Letters. 529(1):126-130, October 2 2002

Murzin A, Brenner S, Hubbard T, Chothia C. SCOP: a structural classification of proteins database for the investigation of sequences and structures. Journal of Molecular Biology. 247:536-540. 1995

Notredame C. Recent evolutions of multiple sequence alignment algorithms. PLoS Comp Biol. 3(8)1405-1408. August 2007

Notredame C, Higgins DG, Heringa J. T-Coffee: a novel method for fast and accurate multiple sequence alignment. Journal of Molecular Biology. 302:205–217, 2000

O’Sullivan O, Suhre K, Abergel C, Higgins DG, Notredame C. 3DCoffee: Combining Protein Sequences and Structures within Multiple Sequence Alignments. J Mol Bio. 340(2):385-395. 2 July 2004

Pei J. Multiple protein sequence alignment. Curr Opin Struct Biol. 18 :382-386. May 2008

Perrodou E, Chica C, Poch O, Gibson TJ, Thompson JD. A new protein linear motif benchmark for multiple sequence alignment software. BMC Bioinformatics. 9:213-228, 2008

Pirovano W, Heringa J. Bioinformatics, volume I: data, sequence analysis, and eveolution, vol. 452. Chapter 7: multiple sequence alignment. Humana press (Totowa, NJ), 143-161, 2008

Puntervoll P, Linding R, Gemund C, Chabanis S, Mattingsdal M, Cameron S, Martin DM, Ausiello G, Brannetti B, Costantini A, Ferre F, Maselli V, Via A, Cesareni G, Diella F, Superti-Furga G, Wyrwicz L, Ramu C, McGuigan C, Gudavalli R, Letunic I, Bork P, Rychlewski L, Kuster B, Helmer-Citterich M, Hunter WN, Aasland R, Gibson TJ. ELM server: a new resource for investigating short functional sites in modular eukaryotic proteins. Nucleic Acids Research. 31(13):3625-3630. 2003

Raghava GP, Searle SM, Audley PC, Barber JD, Barton GJ. OXBench: A benchmark for evaluation of protein multiple sequence alignment accuracy. BMC Bioinformatics. 4(47), 10 October 2003

Russell RB, Barton GJ Multiple protein sequence alignment from tertiary structure comparison: assignement of global and residue confidence levels. Proteins. 14:309-323, 1992

Shindyalov IN, Bourne PE. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. Protein Eng. 11, 739–747. 1998

Simossis VA, Heringa J. PRALINE: a multiple sequence alignment toolbox that integrates homology-extended and secondary structure information. Nucleic Acids Res. 33:289-294, 2005

Stebbings LA, Mizuguchi K. HOMSTRAD: recent developments of the Homologous Protein Structure Alignment Database. Nucleic Acids Research. 32:203-207, 2004

Subramanian AR, Weyer-Menkhoff J, Kaufmann M, Morgenstern B. DIALIGN-T: an improved algorithm for segment-based multiple sequence alignment. BMC Bioinformatics. 6:66, 2005

Thomson J, Higgins D, Gibson T. Clustal W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Res. 22:4673-4690. 1994

Van Walle I, Lasters I, Wyns L. SABmark – a benchmark for sequence alignment that covers the entire known fold space. Bioinformatics. 21(7):1267-1268. 2005

Wallace IM, O’Sullivan O, Higgins DG, Notredame C. M-Coffee: Combining multiple sequence alignment methods with T-Coffee. Nucleic Acids Res. 34:1692-1699. 2006

Wang L, Jiang T. On the complexity of multiple sequence alignment. J Comput Biol. 1(4):337-348. 1994 Winter

Zhou H, Zhou Y. SPEM: improving multiple sequence alignment with sequence profiles and predicted secondary structures. Bioinformatics. 21:3615-3621. 2005