Recent progresses in multiple sequence alignment: a survey

Cédric Notredame

Information Génétique et Structurale, UMR 1889, 31 Chemin Joseph Aiguier, 13 006 Marseille, France
Tel.: +33 (0)4 911 646 06
Fax: +33 (0)4 911 645 49
E-mail: cedric.notredame@igs.cnrs-mrs.fr

The assembly of a multiple sequence alignment (MSA) has become one of the most common tasks when dealing with sequence analysis. Unfortunately, the wide range of available methods and the differences in the results given by these methods makes it hard for a non-specialist to decide which program is best suited for a given purpose. In this review we briefly describe existing techniques and expose the potential strengths and weaknesses of the most widely used multiple alignment packages

Keywords: please provide

Ashley Publications Ltd
www.ashley-pub.com

Introduction

Sequence alignment is by far the most common task in bioinformatics. Procedures relying on sequence comparison are diverse and range from database searches [1] to secondary structure prediction [2]. Sequences can be compared two by two to scour databases for homologues, or they can be multiply aligned to visualize the effect of evolution across a whole protein family. In this study we will focus on the later methods, dedicated to the global simultaneous comparison of more than two sequences. Special emphasis will be given to the most recently described techniques.

The many uses of MSAs

Multiple alignments constitute an extremely powerful means of revealing the constraints imposed by structure and function on the evolution of a protein family. They make it possible to ask a wide range of important biological questions and they will each be discussed in turn.

Phylogenetic analyses

Phylogenetic trees are instrumental in elucidating the evolutionary relationships that exist among various organisms. Nowadays, highly accurate phylogenetic trees rely on molecular data. Their computation typically involves four steps:

- collection of a set of orthologous sequences in a database
- multiple alignment of the sequences
- measure of pair-wise phylogenetic distances on the multiple alignment and computation of a distance matrix
- computation of the tree by applying a clustering algorithm [3] to the distance matrix

As an alternative to the last two bullets the tree may also be computed using maximum likelihood [4]. In both cases, the role of multiple alignment is to provide a very accurate estimation of pair-wise distances and to make it possible to estimate the reliability of each branch by bootstrapping [5].

Identification of conserved motifs and domains

MSAs make it possible to identify motifs preserved by evolution that play an important role in the structure and function of a group of related proteins. Within a multiple alignment, these elements often appear as columns with a lower level of variation than their surroundings. When coupled with experimental data, these motifs constitute a very powerful means of characterizing sequences of unknown function. Important databases like PROSITE [6] or PRINTS [7] rely on this principle. When a motif is too subtle to be defined with a standard pattern, one may use another type of descriptor known as a profile [8] or a hidden Markov model (HMM) [9]. These are meant to exhaustively summarize (column by column) the properties of a protein family or a domain. Profiles and HMMs make it possible to identify very distant members of a protein family when searching a database. Their sensitivity and specificity is much higher than that provided by a single sequence or a pattern. In practice, one can derive their own profile from multiple alignments using packages such as: the PFTOOLS [10], pre-established collections like Pfam [11], or compute the profiles on the fly with PSI-BLAST [12] the position specific version of BLAST. The specificity and sensitivity of a profile are tightly correlated to the biological quality of the multiple alignment it was derived from.

Structure prediction

Structure prediction is another important use of multiple alignments. Secondary and tertiary

 ISSN 1462-2416

structure prediction aim at predicting the role a residue plays in a protein structure (buried or exposed, helix or strand etc.). Secondary structure predictions based on a single sequence yield a low accuracy (in the order of 60%) [13], while predictions based on a MSA go much higher (in the order of 75%) [2,14,15]. The rationale behind such improvements is that the pattern of substitutions observed in a column directly reflects the type of constraints imposed on that position in the course of evolution. In the context of tertiary structure determination or when predicting non-local contacts, multiple alignments can also help to identify correlated mutations. This approach has only given limited results when applied to proteins [16], it has been much more successful in RNA analysis where it allows highly accurate predictions [17] well confirmed by structural analysis.

Altogether, these very important applications explain the amount of attention dedicated to the MSA problem and any biologist should be aware that very few bioinformatics protocols bypass the multiple alignment stage. Unfortunately, available tools are only heuristics providing an approximate solution to a problem that remains largely open. These many heuristics are based on different paradigms, each well suited to a limited range of situations.

A complicated problem.

MSA is a complicated problem. It stands at the-cross road of three distinct technical difficulties:

- the choice of the sequences
- the choice of an objective function (i.e., a comparison model)
- the optimization of that function

Altogether, properly solving these three problems would require an understanding of statistics, biology and computer science that lies far beyond our grasp.

The choice of the sequences

The methods reviewed here (i.e., global MSA methods) only make sense if they are assumed to be dealing with a set of homologous sequences i.e., sequences sharing a common ancestor. Furthermore, with the exception of DiAlign [18], global methods require the sequences to be related over their whole length (or at least most of it). When that condition is not met, one should consider the use of local MSA methods such as the Gibbs sampler [19], Match-Box [20] or MACAW [21]. In any case, one should always be aware that given inappropriate sequences, most multiple alignment routines will nonetheless produce an alignment. It will be the responsibility of the biologist to realize that this alignment is meaningless. This is not an easy task, and a few years ago Henikoff reviewed a series of problems that can occur when one forces multiple alignments with unrelated sequences [22]. In order to recruit a set of homologous sequences, it is common practice to use one of the BLAST programs (WU-BLAST, PSI-BLAST, GAPPED BLAST etc.) [12], for searching within a database all the sequences similar to some query sequence. When doing so, an observed similarity is considered good when it is unlikely to arise by chance (given the database and the amino-acid frequencies). To make this estimation, BLAST uses powerful statistical models developed by Altschul and Karlin [23]. Of course, these statistical models merely approximate the biological reality, and homology may be misrepresented by similarity, leading to the incorporation of improper sequences within a multiple alignment.

The choice of an objective function

This is purely a biological problem that lies in the definition of correctness. What should a biologically correct alignment look like? Can we define its expected properties and will we recognize it when we see it? These intricate questions can only be answered by means of a mathematical function able to measure an alignment biological quality. We name this function an Objective Function (OF) because it defines the mathematical objective of the search. Given a perfect function, the mathematically optimal alignment will also be biologically optimal. Yet this is rarely the case, and while the function defines a mathematical optimum, we rarely have an argument that this optimum will also be biologically optimal.

Defining a proper objective function is a highly non-trivial task and an active research field of its own right. In theory, an OF should incorporate everything that is known about the sequences, including their structure, function and evolutionary history. This information is rarely at hand and is hard to use, so it is usually replaced with sequence similarity. Thus, a very simple general function is often used: the weighted sums-of-pairs with affine gap penalties [24]. Under this model, each sequence receives a weight proportional to the amount of independent information it contains [25] and the cost of the multiple alignment is equal to the sum of the

cost of all the weighted pair-wise substitutions. The substitution costs are evaluated using a predefined evolutionary model known as a substitution matrix [26], in which a score is assigned to every possible substitution or conservation according to its biological likeliness (i.e., rarely observed mutations receive a negative score while mutations observed more often would be expected by chance receive to a positive score). Insertions or deletions are scored using affine gap penalties that penalize a gap once for opening and then proportionally to its length. This penalty scheme is a major source of concern because it requires two parameters:

- The gap opening
- The gap extension penalty

whose adequate values can only be set empirically and may vary from one set of sequences to the next [27]. Although this function is clearly wrong from an evolutionary point of view [24], because it assumes every sequence within the set to be an ancestor of every other sequence, the ease of its implementation has made it popular with the most widely used MSA packages [28-30]. This validation was recently confirmed by a more thorough benchmarking [31] indicating that packages that rely on the sums-of-pairs are reasonable performers as judged by the biological quality of the alignments they produce. Very recently, a new variant of the sum-of-pairs function has been introduced that seems less likely to over-estimate evolutionary events [32].

Over the last years, new OFs were described that seem to be less sensitive to gap penalty estimation thanks to the incorporation of local information. These include the segment-based evaluation of DiAlign [33] and the consistency objective function of T-Coffee [34]. HMMs [9,35] constitute another line of thought recently explored. HMMs describe the multiple alignment in a statistical context, using a Bayesian approach. Although from a formal point of view they provide us with the most attractive solution, their performances for *ab initio* alignments have so far been disappointing and recent work shows that carefully tuned HMM packages barely outperform ClustalW [36]. Other statistically-based methods that attempt to associate a P-value to the multiple alignment have been described [19,37]. Unfortunately, these measures are restricted to ungapped MSAs.

All things considered, one should be well aware that there is no such thing as the ideal OF and every available scheme suffers from major drawbacks. In an ideal world, a perfect OF would be available for every situation. In practice, this is not the case and the user is always left to make a decision when choosing the method that is most suitable to the problem.

Computational

The third problem associated with MSAs is computational. Assuming we have at our disposal an adequate set of sequences and a biologically perfect objective function, the computation of a mathematically optimal alignment is too complex a task for an exact method to be used [38]. Even if the function we are interested in was as simple as a maximization of the number of perfect identities within each column, the problem would already be out of reach for more than three sequences. This is why all the current implementations of multiple alignment algorithms are heuristics and that none of them guarantee a full optimization. Considering their most obvious properties, it is convenient to classify existing algorithms in three main categories: exact, progressive and iterative. *Exact algorithms* are high quality heuristics that deliver an alignment usually very close to optimality [28,39], sometimes but not always within well-defined boundaries. They can only handle a small number of sequences (< 20) and are limited to the sums-of-pairs objective function. *Progressive alignments* are by far the most widely used [34,40,41]. They depend on a progressive assembly of the multiple alignment [42-44] where sequences or alignments are added one by one so that never more than two sequences (or multiple alignments) are simultaneously aligned using dynamic programming [45]. This approach has the great advantage of speed and simplicity combined with reasonable sensitivity, even if it is by nature a heuristic that does not guarantee any level of optimization. Other progressive alignment methods exist such as DiAlign [18] or Match-Box [20], which assemble the alignment in a sequence-independent manner by combining segment pairs in an order dictated by their score, until every residue of every sequence has been incorporated in the multiple alignment. *Iterative alignment* methods depend on algorithms able to produce an alignment and to refine it through a series of cycles (iterations) until no more improvements can be made. Iterative methods can be deterministic or stochastic, depending on the strategy used to improve the alignment. The simplest iterative strategies are deterministic. They involve extracting sequences one by one

from a multiple alignment and realigning them to the remaining sequences [46,47], some of these methods can even be a mixture of progressive and iterative strategies [48]. The procedure is terminated when no more improvement can be made (convergence). Stochastic iterative methods include HMM training [49] and simulated annealing or genetic algorithms [50-56]. The main advantage is to allow for a good conceptual separation between optimization processes and OF. Recent examples of algorithms belonging to these three categories are reviewed in the next section.

Review

The number of available MSA methods has steadily increased over the last 20 years. Being exhaustive on these will not be possible within the scope of this work, and this review should be seen as complementary with another recent review [57]. Furthermore, it should be pointed out that only a minority of the methods described in the literature have found their way towards regular usage. There are many reasons for failure, but the main one stems from a simple fact: there is no satisfactory theoretical framework in sequence analysis, in this context an algorithm is only as good as it is useful. Improvements are driven by results and not theory, so that programs with badly designed interfaces or poor portability have been disgarded by natural selection, leaving their algorithms to be reinvented by later generations.

Over the last few years, the field of MSA has undergone drastic evolutionary changes with the introduction of several new algorithms and new evaluation methods. Some of the methods used for mutiple sequence alignments are listed in **Table 1**. Among all this, two new trends have emerged:

- the increasing use of iterative optimisation strategies (stochastic or non-stochastic)
- the use of consistency-based scoring schemes

In this section, we review some of these new algorithms, their main characteristics and potential shortcomings. Another major trend, that will not be extensively covered here, has been the introduction of HMMs methods [9,35]. A very detailed account on HMM-based methods for MSAs may be found in [58].

The progressive algorithms.

Progressive alignment constitutes one of the simplest and most effective ways of multiply aligning a set of sequences in little time and with little memory. This algorithm was initially described by Hogeweg [42] and later re-invented by Feng [43] and Taylor [44]. The most widely used MSA packages are based on an implementation of this algorithm, which include: Pileup, a part of the GCG package [59], MultAlign [41] and ClustalW [29] that has become the standard method for multiple alignments.

ClustalW is a non-iterative, deterministic algorithm that attempts to optimize the weighted sums-of-pairs with affine gap penalties. It is a straightforward progressive alignment strategy where sequences are added one by one to the multiple alignment according to the order indicated by a pre-computed dendrogram. Sequence addition is made using a pair-wise sequence alignment algorithm [45]. The main shortcoming of this strategy is that once a sequence has been aligned, that alignment will never be modified even if it conflicts with sequences added later in the process as shown in **Figure 1**. ClustalW also includes many highly specialized heuristics meant to maximally exploit sequence information:

- local gap penalties
- automatic substitution matrix choice
- automatic gap penalty adjustment
- the delaying of the alignment of distantly related sequences

Benchmarking tests, carried out on BAliBASE [31], a database of reference multiple sequence alignments. In general, ClustalW performs better when the Phylogenetic tree is relatively dense without any obvious outlier. It does not matter how widely the sequences are spread just as long as every sequence remains close enough (a bit like crossing a river stepping from stone to stone). Long insertions or deletions also cause trouble, due to the intrinsic limitation of the affine penalty scheme used by ClustalW.

The latest improvement to the progressive alignment algorithm is T-Coffee, a novel strategy where sequences are aligned in a progressive manner but using a consistency-based objective function that makes it possible to minimize potential errors, especially in the early stages of the alignment assembly. T-Coffee is reviewed in more detail in the consistency-based algorithm section.

Exact algorithms

As mentioned earlier, progressive alignment is only an approximate solution. In order to use the

Table 1. Some recent and less recent available methods for MSAs.

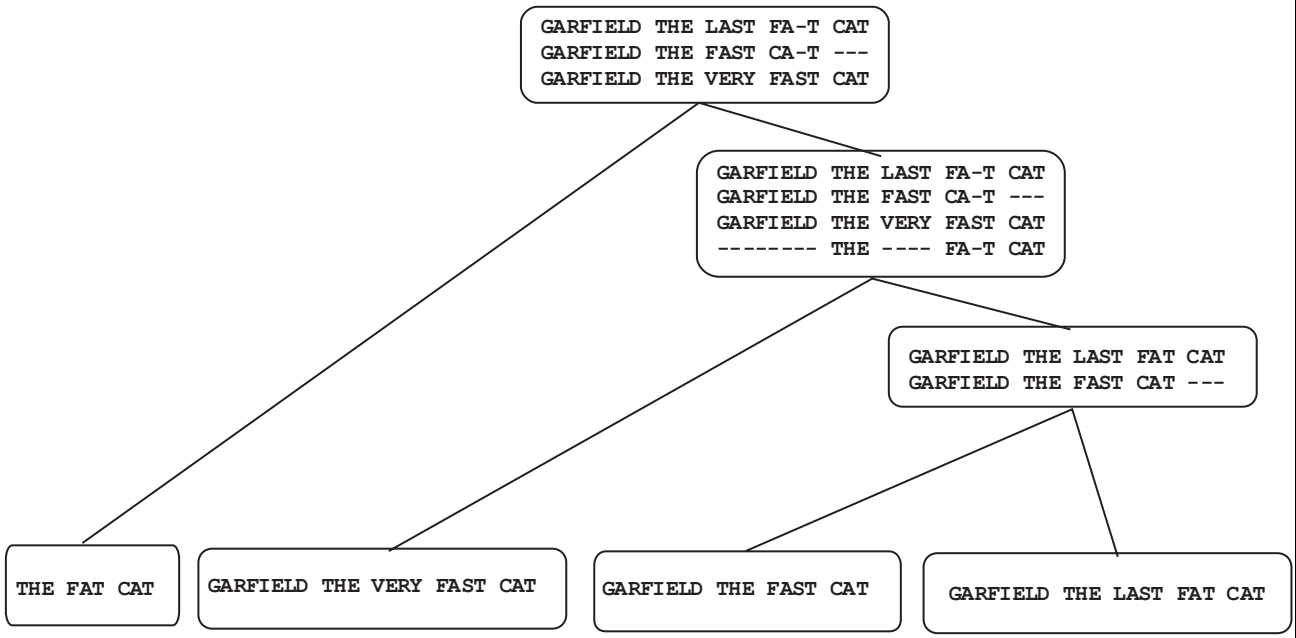
Name	Algorithm	URL	Ref.
MSA	Exact	http://www.ibc.wustl.edu/ibc/msa.html	[28]
DCA	Exact (requires MSA)	http://bibiserv.techfak.uni-biefield.de/dca	[39]
OMA	Iterative DCA	http://bibiserv.techfak.uni-biefield.de/oma	[61]
ClustalW, ClustalX	Progressive	ftp://ftp-igbmc.u-strasbg.fr/pub/clustalW or clustalX	[29]
MultAlin	Progressive	http://www.toulouse.inra.fr/multalin.html	[41]
DiAlign	Consistency-based	http://www.gsf.de/biodv/dialign.html	[18]
ComAlign	Consistency-based	http://www.daimi.au.df/~ ocaprani	[75]
T-Coffee	Consistency-based/progressive	http://igs-server.cnrs-mrs.fr/~ cnotred	[66]
Praline	Iterative/progressive	jhering@nimr.mrc.ac.uk	[48]
IterAlign	Iterative	http://giotto.Stanford.edu/~ luciano/iteralign.html	[70]
Prrp	Iterative/Stochastic	ftp://ftp.genome.ad.jp/pub/genome/saitama-cc/	[47]
SAM	Iterative/Stochastic/HMM	rph@cse.ucsc.edu	[84]
HMMER	Iterative/Stochastic/HMM	http://hmmer.wustl.edu/	[68]
SAGA	Iterative/Stochastic/GA	http://igs-server.cnrs-mrs.fr/~ cnotred	[51]
GA	Iterative/Stochastic/GA	czhang@watnow.uwaterloo.ca	[52]

signal contained in the sequences properly, one would like to simultaneously align them, rather than adding them one by one to a multiple alignment. This would be especially useful when dealing with sets of extremly divergent sequences whose pair-wise alignments are all likely to be incorrect. Unfortunately, to align several sequences, one would need to generalize the Needlman and Wunsch algorithm [45] to a multi-dimensional space and for practical reasons (time and memory) this is only possible for a maximum of three sequences. That limit can be pushed a bit further if one finds a way to identify in advance the portion of the hyperspace that does not contribute to the solution and exclude it from computation. This is achieved in the MSA program, an implementation of the Carrillo and Lipman algorithm [60] that makes it possible to align up to ten closely related sequences [28]. It should be stressed here that, contrary to a widespread belief, the MSA program is only a heuristic implementation of the Carillo and Lipman algorithm, that is not guaranteed to reach the mathematical optimum. MSA uses lower and upper bounds tighter than the guaranteed ones (Altschul, personal communication). Even so, the high memory requirement, the lengthy computational time and the limitation on the number of sequences explain why the MSA program quickly gave way to ClustalW. Yet, MSA met again with popularity when Stoye described a new divide and conquer algorithmDCA [39] that sits on the top of MSA and extends its capabilities. The DCA algorithm cuts the sequences in subsets of segments that are small enough to be fed to MSA. The sub-alignments are later reassembled by DCA. The trick is to cut the sequences at the right points so that the produced alignment remains as close as possible to optimality. The way it is done in DCA is slightly heuristic albeit fairly accurate. Benchmarking on BAliBASE indicated that the DCA strategy does slightly better that ClustalW, even if the four largest BAliBASE test sets could not be computed with DCA (Notredame, unpublished results). Even when MSA is coupled to DCA, strong limitations remain on the number of sequences that can be handled (20–30) and on their phylogenetic spread. Recently, an iterative implementation of DCA [61], optimal multiple alignment (OMA) was described that is meant to speed up the DCA strategy and decreases its memory requirements.

Iterative algorithms

Iterative algorithms are based on the idea that the solution to a given problem can be computed by modifying an already existing sub-optimal solution. Each 'modification' step is an iteration. In the examples considered here, modifications can be made using dynamic programming or various random protocols. While the dynamic programming-based protocols can also include elements of randomization, we distinguish them from more traditional stochastic iterative meth-

Figure 1. Limits of the progressive strategy.



This example shows how a progressive alignment strategy can be misled. In the initial alignment of sequences 1 and 2, ClustalW has a choice between aligning CAT with CAT and making an internal gap or making a mismatch between C and F and having a terminal gap. Since terminal gaps are much cheaper than internals, the ClustalW scoring schemes prefers the former. In the next stage, when the extra sequence is added, it turns out that properly aligning the two CATs in the previous stage would have led to a better scori ng sums-of-pairs multiple alignment.

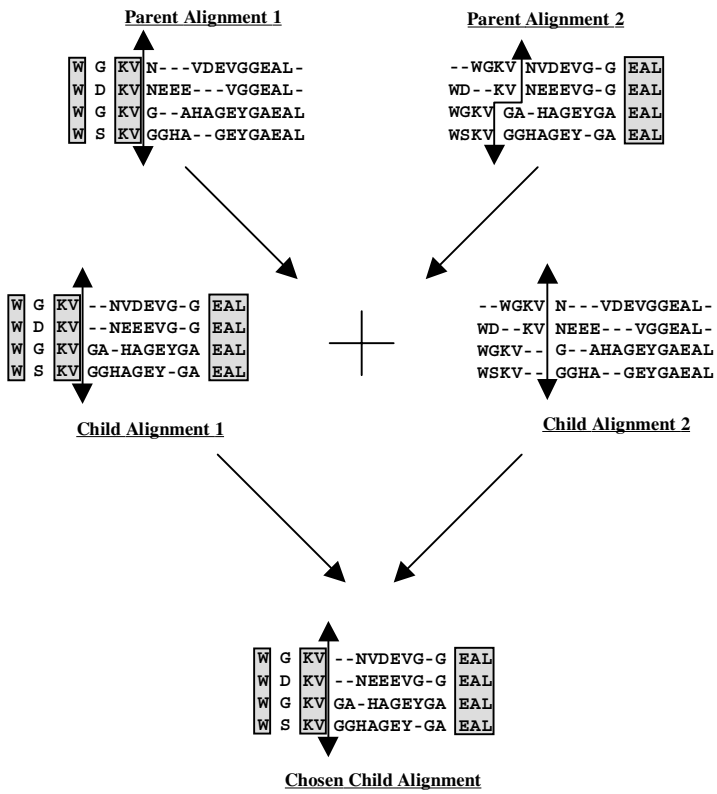
ods such as simulated annealing (SA) [62] or genetic algorithms (GA) [63].

Stochastic iterative algorithms

SA was the first stochastic iterative method described for simultaneously aligning a set of sequences. Various schemes have been published [50,64], which all involve the same chain of processes: an alignment is randomly modified, its score assessed, it is kept or discarded according to an acceptance function that gets more stringent while the iteration number increases (by analogy with a decreasing temperature during crystallization), the process goes on until a finishing criteria such as convergence is met. In practice, despite being intellectually very attractive, SA is too slow for making *ab initio* alignment and it can only be used as an alignment improver. GAs constitute an interesting alternative to SA as shown in SAGA [51], a GA dedicated to MSA. Like SA, SAGA is an optimization black box in which any OF invented can be tested. The principle of SAGA is very straightforward and follows closely the 'simple GA' [65]: randomly generated multiple alignments of a given set of sequences evolve under some selection pressure.

These alignments are in competition with each other for survival (survival of the fittest) and reproduction. Within SAGA, fitness depends on the score measured by the objective function (the better the score, the fitter the multiple alignment). Over a series of cycles known as generations, alignments will die or survive, depending on their fitness. They can also improve and reproduce through some stochastic modifications known as mutations and crossovers. Mutations randomly insert or shift gaps while crossovers combine the content of two alignments (**Figure 2**). Overall, 20 operators co-exist in SAGA and compete for usage. The program does not guarantee optimality but has been shown to equal or outperform MSA from a mathematical point of view on 13 test sets (using exactly the same OF in both programs). The complete disconnection between the operators and the original OF made it possible to seamlessly modify the original OF in order to test SAGA with a new OF named COFFEE (Consistency Objective Function For align-

Figure 2. One point crossover in SAGA.



This figure illustrates the manner in which two alignments are combined into one in SAGA, a genetic algorithm that evolves a population of alignments toward optimality. The principle is to cut straight one of the alignments and to cut the second one so that compatible ends are generated.

mEnt Evaluation) [66]. This series of studies revealed the suitability of GAs to become investigation tools but also made it clear that GAs were too slow a strategy for large-scale projects or everyday use. Another similar MSA GA was later introduced by Zhang and Wong [52]. The authors report a very high efficiency for their GA but these results must be considered with care since their strategy (especially the mutations) is driven by the presence of completely conserved segments that guide the assembly of the alignments. The assumption that such segments will always exist when aligning proteins is not realistic. This method appears to be appropriate when comparing very long highly similar sequences (such as portions of genomes). SAGA was later parallelized by two independent groups [67,53], in order to improve its efficiency. The model described in SAGA has been met with considerable interest in the evolutionary programming community and, in recent years, at least three algorithms based on the SAGA principle have been published [54-56].

The Gibbs sampler is another interesting stochastic iterative strategy [19]. It is a local multiple alignment method that finds ungapped motifs among a set of unaligned sequences. From a multiple alignment perspective, the most interesting feature of the Gibbs sampler is its OF. The algorithm aims to build an alignment with a good P-value (i.e., a low probability of having been generated by chance). At each iteration, segments are removed or added according to the probability that the current model (the rest of the alignment) could have generated them. If that probability is high enough, the model is then updated with the new segments and the algorithm proceeds toward the next iteration. The overall result is an alignment that has a good P-value and maximizes the probability of the data it contains (i.e., each sequence fits well within the alignment). This Bayesian idea of simultaneously maximizing the data and the model is also central to HMMs [9,35], thus it is not surprising to find that HMMs can also be trained by expected maximization [49,68]. However, like GAs, HMMs proved rather disappointing when it came to *ab initio* alignments. Today, HMMs such as those found in Pfam [11] are no longer generated from unaligned sequences. State of the art protocols are much more inclined toward turning a pre-computed alignment into an HMM and further refining it using HMMER [49] or SAM [68].

Non stochastic iterative algorithms.

The first non-stochastic iterative algorithms date back to the origins of MSAs [46]. The idea is simple and attractive: since mistakes may arise in the early stages of a progressive alignment, why not correct them later by re-aligning each sequence in turn to the multiple alignment using standard dynamic programming algorithms [45]. The procedure terminates when iterations consistently fail to improve the alignment. This very simple algorithm constitutes most of the iterative strategies described in the early 1990s. The main scope for variation is the way sequences are divided into two groups before being re-aligned. In AMPS [46], sequences are chosen according to their input order and re-aligned one by one. In the algorithm of Berger and Munsen [69], the choice is made in a random manner and sequences are divided into two groups that can contain more than one sequence. The element of

randomization makes the algorithm more robust and improves its accuracy. Few of these early iterative methods have been properly benchmarked, making it hard to estimate their true biological significance.

The most sophisticated DP-based iterative algorithm available was recently described by Gotoh [47]. It is a double nested iterative strategy with randomization that optimizes the weighted sums-of-pairs with affine gap penalties (**Figure 3**). The originality of this algorithm is that the weights and the alignment are simultaneously optimised. The inner iteration optimizes the weighted sums of pairs while the outer iteration optimizes the weights that are calculated on a phylogenetic tree estimated from the current alignment [25]. The algorithm terminates when the weights have converged. Prrp was the first multiple alignment program to be extensively benchmarked, using JOY, a database of structural alignments. The results were confirmed on BAliBASE [31,34]. Prrp significantly out-performs most of the traditional progressive methods as well as some of the most recent iterative strategies (**Table 2**).

Two other iterative alignment methods were recently described: Praline [48] and IterAlign [70]. These two methods share very similar protocols. They both start with a preprocessing of the sequences to align. In IterAlign, sequences are *'ameliorated'(sic)*, this means that each sequence is locally compared to others and that every segment that shows high similarity with other proteins is replaced by a consensus. One round of 'amelioration' constitutes one iteration. Other iterations are run on the new set of 'ameliorated' sequences, until the collection of consensus converges. Consistent blocks are then extracted from the consensus collection and these blocks are chained in order to produce the final alignment. Praline uses a very similar protocol: sequences are replaced with a complete profile made from a multiple alignment that only includes their closest relatives. That profile step is iterated until the collection of profiles converges. This collection of profiles is conceptually similar to the 'ameliorated' set of sequences used by IterAlign. The multiple alignment is then assembled by using a straightforward progressive algorithm where sequences are replaced with profiles. One of the most interesting consequences of the protocol used in Praline is the possibility of measuring the consistency between the final alignment and the collection of profiles used for its assembly. There may be some correlation between this measure and the true accuracy of the alignment.

Regardless of the potential performances of these two methods (neither have been properly bench marked), some emphasis should be given to the novel concepts they incorporate:

- the first one is the use of local information in IterAlign, in order to decrease sensitivity to the gap penalty parameterization
- the second key concept is consistency

Sequences are preprocessed so that the regions consistently conserved across the family see their signal enhanced and become more likely to drive the alignment. This search for consistency has been one of the strongest trend in recent developments of MSA. It is also central to the non-iterative methods.

Consistency-based algorithm

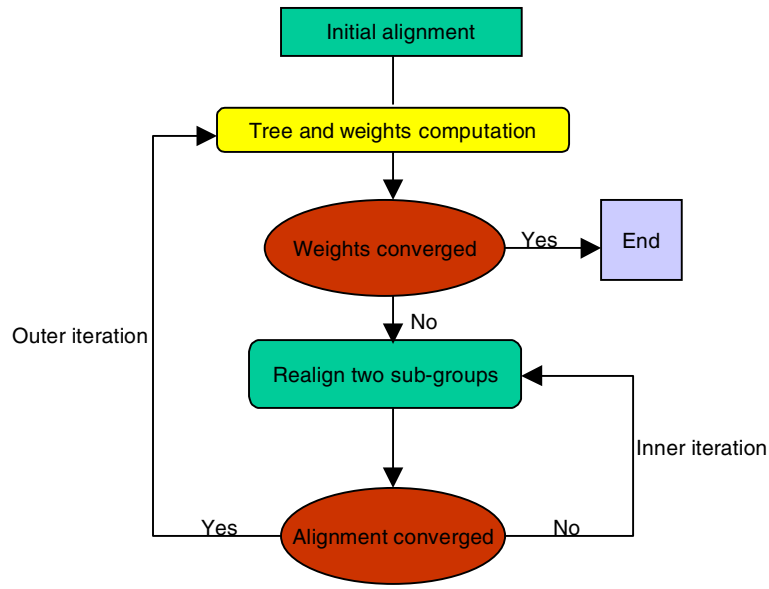
The first consistency-based MSA method was described by Kececioglu in the 80s [71]. Given a set of sequences, the optimal MSA is defined as the one that agrees the most with all the possible optimal pair-wise alignments. Computing that alignment is an NP complete problem that can only be solved for a small number of related sequences, using an MSA-like algorithm. Nonetheless, there are at least three good reasons that make consistency-based OFs very attractive:

- firstly, they do not depend on a specific substitution matrix but rather on any method or collection of methods able to align two sequences at a time
- secondly, the consistency-based scheme is position dependant, given the collection of pair-wise alignments. This means that the score associated with the alignment of two residues depends on their indexes (position within the protein sequence) rather than their individual nature
- the third reason is more general and has to do with consistency. Experience shows that given a set of independent observations, the most consistent are often closer to the truth

This principle generally holds well in biology and can be loosely connected to the observation that, given a series of measurements, noise spreads while signal accumulates.

Although the first consistency-based OF was described in 1983, it took several more years to develop heuristic algorithms able to deal with optimization and it is only recently that a GA, (SAGA [51]) was used to show the biological

Figure 3. Layout of Prrp.



This figure shows the layout of Prrp, a double-nested strategy for optimizing multiple alignments. When the inner iteration has converged, new sequence weights are estimated. The convergence of these weights is the criteria for the outer iteration to stop.

Table 2. Some elements of validation on BAliBASE.

Method	Ref1	Ref2	Ref3	Ref4	Ref5	Total
DiAlign	71.0	25.2	35.1	74.7	80.4	57.3
ClustalW	78.5	32.2	42.5	65.7	74.3	58.7
Prrp	78.6	32.5	50.2	51.1	82.7	59.0
T-Coffee	80.7	37.3	52.9	83.2	88.7	68.7

Each method in the Method column was used to align the 141 test-sets contained in BAliBASE. The alignments were then compared with the reference BAliBASE alignment using aln_compare [34]. Ref1–5 indicates the five BAliBASE categories. Results obtained in each category were averaged. All the observed differences are statistically significant, as assessed by the Wilcoxon rank-based test [34,47]. Ref1 contains a homogenous set of sequences, ref2 contains a homogenous group of sequences and an outlayer, ref3 contains two distantly related groups of sequences. Ref4 contains sequences that require long internal gaps to be properly aligned and ref5 contains sequences that require long-terminal gaps to be properly aligned. Total is the average of ref1–5.

advantages of such a function, COFFEE [66], which emulates the maximum weight trace problem. In SAGA-COFFEE, the collection of weighted pair-wise alignments is named a library and SAGA is used to compute the alignment that has the highest level of consistency with the library. In practice, the library may contain more than one alignment for each pair of sequences, the information it contains may be redundant, conflicting and may originate from sources as various as one wishes (structure analysis, sequence comparison, database search, experimental knowledge etc.). Although SAGA-COFFEE yielded interesting results, the GA was too slow for everyday use. This prompted the development of a new heuristic algorithm to optimize the COFFEE function in a time efficient manner: T-Coffee (**Figure 4**). In T-Coffee, the COFFEE library is turned into a so-called 'extended library', a position-specific substitution matrix where the score associated with each pair of residues depends on the compatibility of that pair with the rest of the library. T-Coffee uses a procedure reminiscent of Vingron's Dot matrix multiplication [72] and Morgenstern overlapping weights [73]. The multiple alignment is assembled using a progressive alignment algorithm similar to the one used in ClustalW:

- pair-wise distances are computed
- a neighbour joining tree is estimated [3]
- the sequences are aligned one by one following the topology of the tree

The main difference between T-Coffee and ClustalW is that in T-Coffee, the extended library replaces a substitution matrix. Another important characteristic of T-Coffee is that its primary library is made of a mixture of global alignments (produced with ClustalW) and local

alignments (produced with Lalign [74]). The bench-marking carried out on BAliBASE shows that this combination of local and global information makes the T-Coffee implementation able to outperform Prrp, ClustalW and DiAlign on the five categories of test-sets contained in this reference database [34]. These results were obtained without tuning, since T-Coffee does not have any parameters of its own. Due to the library extension, T-Coffee does more than simply compute a consensus alignment. Nonetheless, given a collection of multiple alignments, it can be interesting to combine them into a single consensus multiple alignment. This is what the ComAlign program does [75] by combining several multiple alignments into a single, often improved, multiple alignment.

Although the details differ, T-Coffee bears some similarity to DiAlign [73], another consistency-based algorithm that attempts to use local information in order to guide a global multiple alignment. DiAlign starts with an identification of highly homologous segment-pairs. The weight of each of these pairs is defined by a P-value comparable to the P-values used in BLAST. Each of these segment-pairs receives another score proportional to its compatibility with the complete set of segment-pairs. This score is named an overlapping weight and segment-pairs weighted this way are very reminiscent of the extended library. The multiple alignment is then progressively assembled by adding the pairs of segments according to their weight. Assembly is made in a sequence independent order, as opposed to the ClustalW-style progressive alignment strategy. Non-compatible segment-pairs are discarded, hence the importance of the order induced by the weights. According to the authors, DiAlign is especially good at properly aligning sequences where local homology is the driving signal. This has been confirmed by BAliBASE benchmarking [31,34]. Overall, DiAlign is not as accurate as ClustalW or Prrp but it does very well in categories 4 and 5 of BAliBASE, which require very long insertions to be properly aligned. Over the past few years, the DiAlign algorithm has been modified on numerous occasions for improved efficiency [76].

Conclusion and expert opinion

Ten years ago, when schemes such as MSA were developed, there was very little data available and the main problem was to use every bit of available information properly. Today the situation has dramatically changed. We are overwhelmed by 'relevant' information and in fact there is so much of it that by choosing the data, one can suit the needs of almost any method (progressive, iterative etc.). Ironically, one could be tempted to say that data has improved faster than mutiple aligment methods.

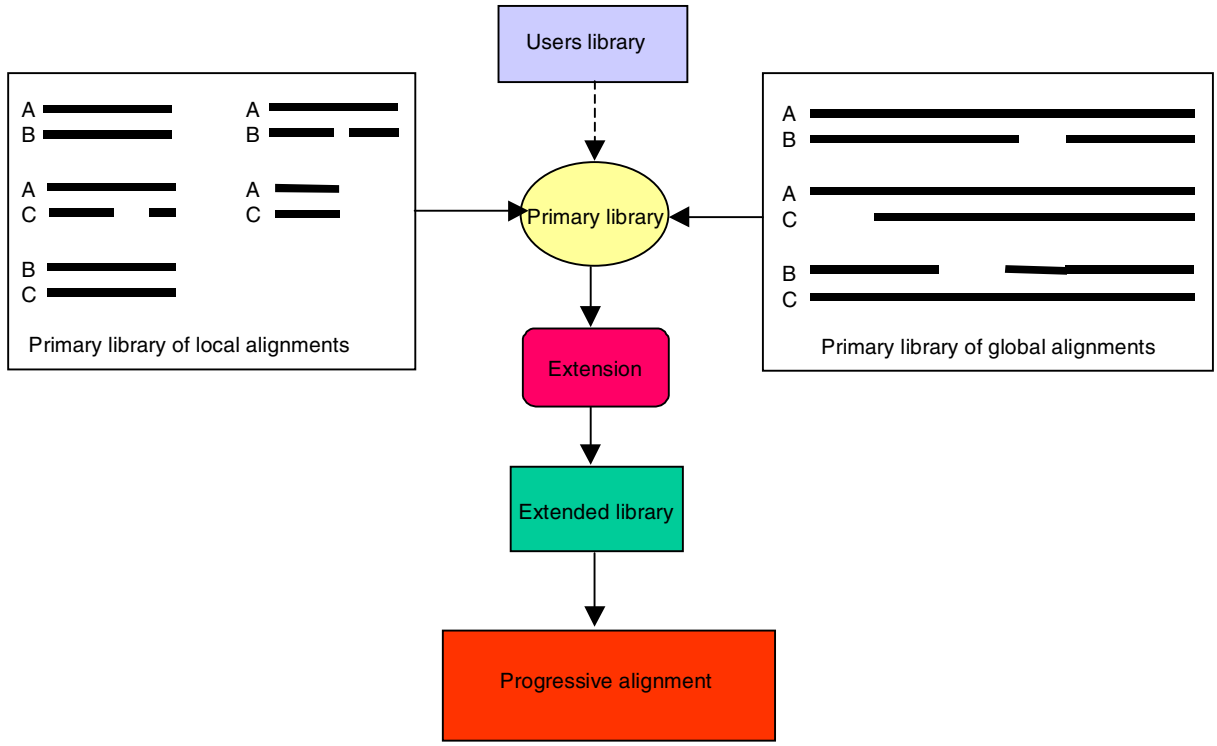
As a consequence, the real challenge is not so much the multiple alignment itself but rather the choice of a subset of sequences that will yield the most biologically correct and informative alignment, given one method or another. There are two good reasons for not using all the available sequences:

- Alignments with a large number of sequences are slow to compute and hard to analyze. Whenever possible, an alignment should fit on a single sheet of A4 paper.
- Limitations of existing programs. Although they all use weighting schemes meant to minimize the effect of similar or highly correlated sequences, none of these schemes are entirely satisfactory, and over-represented sub-groups always end up dominating the alignment or the profile.

This can prevent the proper alignment of less well represented sequence sub-groups that may be just as important. Careful user's trimming is still the best available way around that effect. Unfortunately, the increased sensitivity of database search tools coupled with the increase in database size has rendered this process very tedious.

The second major change that has occurred over the last years is the increasing number of available 3D structures. Although the proportion of protein sequences with a known 3D structure is getting smaller and smaller, the situation is very different from a protein family perspective and the proportion of protein families where at least one member has a known 3D structure increases regularly. This means that in most cases, multiple alignment modelling could benefit from the incorporation of 3D structural information, in order to enhance very remote homologies, or to guide the choice of local penalties [77]. Very few of the packages avaliable are able to mix structure and sequences within a multiple alignment. While ClustalW is able to use SwissProt secondary structure information for gap penalty estimation, a proper tool is still lacking for the simultaneous alignment of sequences and structures. Two of the methods introduced here are good candidates for such a combination. The consistency-based algorithms

Figure 4. Layout of T-Coffee.



This figure indicates the layout of T-Coffee. Local and global pairwise alignments are first computed and then combined into a primary library that is extended in order to be used for computing the multiple sequence alignment in a progressive manner.

have the advantage of having few requirements on the origin of their libraries. For instance, DALI, the database of structural multiple alignments [78] relies on T-Coffee to assemble the collection of pair-wise alignments produced by the DALI algorithm into a multiple alignment. The double dynamic programming algorithm introduced by Taylor [79] is also a good candidate for that purpose. While it has been shown that this algorithm is suitable for structure-to-structure alignments [80], recent results indicate that it could also be used in the context of MSA and possibly as a means to mix sequences and structures [81].

The third major obstacle on the road that leads toward an informative multiple alignment is the processing of repeats. Repeated sequences (in tandem or not) are renowned for confusing all the existing MSA methods. When dealing with sequences that contain such repeats, the only solution is to pre-process the sequences, extract the repeats and only align homologous regions. This extraction can be made using any local multiple alignment tool such as the Gibbs sampler [19], Mocca [82] or Repro [83]. Unfortunately, none of these tools are well integrated within a global multiple alignment procedure. The Gibbs sampler and Mocca have the advantage of providing the user with some estimation of the biological relevance of their output.

The fourth point that needs to be raised here is computation. While elegant solutions have been found to parallelize database searches, the parallelization of a MSA algorithm remains a difficult task. The operations involved in the implementation of these algorithms require complex schemes of memory sharing that are not suitable to Linux-farms and other clusters. When dealing with large sets of data of long sequences, supercomputers are still required for multiple alignment programs.

The last important point is the estimation of local accuracy. The common property of all the methods introduced here is that no one in particular is the best. They may all be out-performed by the others on one protein family or another. For that reason, we feel that it is more important to be able to assess the exact level of

Highlights

- MSAs are essential bioinformatics tools. They are required for phylogenetic analysis, to scan databases for remote members of a protein family and structure prediction.
- No perfect method exists for assembling a multiple sequence alignment and all the available methods do approximations (heuristics).
- The most commonly used methods for doing multiple sequence alignments use a progressive alignment algorithm (ClustalW [31]).
- Recent progress have focused on the design of iterative (Prrp [30], SAGA[51]) and consistency based methods (DiAlign [33], T-Coffee [34], Praline [48] and IterAlign [70]).
- Benchmarking on a collection of reference alignments (BAliBASE [31]) indicates that ClustalW [31] performs reasonably well on a wide range of situations while DiAlign is more appropriate for sequences with long insertions/deletions. These tests also indicate that T-Coffee [34] is on average the best available method among those evaluated that way.
- Future methods should be able to integrate structural information within the multiple alignments and to allow some estimation of their local reliability.

accuracy of an alignment, rather than improving the average performances of each method. To our knowledge, only four packages, incorporate an estimation of the alignment local quality: ClustalX (the X-Window interface of ClustalW), Praline [48], T-Coffee [34] and Match-Box [20]. None of these methods for estimating local accuracy have been thoroughly benchmarked and properly validated for estimation.

To conclude, a multiple alignment is merely a very constrained model. It is a powerful way to spot inconsistencies amongst a data set and to visualize relationships that may exist among seemingly independent pieces of information. Multiple alignment may be driven by any available source of information for instance structure, sequence, experimental knowledge and so on.

Outlook

Are multiple sequence alignments here to stay? The answer is yes, without any doubt. While we enter the area of comparative genomics, the simultaneous comparison of a large number of homologous biological objects will become more and more important in our understanding of biology and there is no doubt than in 5 to 10 years, multiple alignments will be as central to the biological analysis as they are now. There is no doubt in my mind that MSA will remain central to sequence-based biology.

This being said, MSA methods will also need to evolve. They will need to integrate heterogeneous information such as structures, results of database searches, experimental data and in general, anything that may come from expression data and proteomic analysis, including regulatory information. Integrating such heterogeneous information is a complex task. When the data is heterogeneous, knowing who is right and who is wrong becomes an art. Addressing that type of questions will be difficult and essential. The appropriate method will have to do this in a transparent way, letting the user control every bit of extra information that goes into his alignment. This ideal method should also allow the user to inject into his model some of his own knowledge. Doing so should be made an easy task. These ideas have been central to development of the underlying philosophy in the T-Coffee package [34].

In any case, these future methods are bound to be memory and CPU hungry. Compared with database searches, multiple sequence alignment protocols are hard to optimize. Special hardware may need to be adapted and the code may have to be redesigned. Several computer manufacturers are currently looking at this problem. One can easily imagine that a powerful multiple sequence alignment server will soon be a feature of most laboratories, just like PCR machines made their appearance in the 90s.

Acknowledgements

The author wishes to thank Dr Jean Michel Claverie for helpful advice and comments. He also wishes to thank the two referees for interesting comments and for putting to his attention several of the most recent references included in this work.

Bibliography

Papers of special note have been highlighted as either of interest (•) or of considerable interest (••) to readers.

1. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ: Basic local alignment search tool. *J. Mol. Biol.* 215, 403-410 (1990).
2. Rost B, Sander C, Schneider R: PHD - an automatic server for protein secondary structure prediction. *CABIOS* 10, 53-60 (1994).
3. Saitou N, Nei M: The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.* 4, 406-425 (1987).
4. Saitou N: Maximum likelihood methods. *Meth. Enzymol.* 183, 584-598 (1990).
5. Felsenstein J: Confidence limits on phylogenies: an approach using the bootstrap. *Evolution* 39, 783-791 (1985).
6. Bairoch A, Bucher P, Hofmann K: The PROSITE database its status in (1997). *Nucleic Acids Res.* 25, 217-221 (1997).
7. Attwood TK, Croning MD, Flower DR *et al.*: PRINTS-S: the database formerly known as PRINTS. *Nucleic Acids Res.* 28(1), 225-227 (2000).
8. Gribskov M, Luethy R, Eisenberg D: Profile analysis. *Meth. Enzymol.* 183, 146-159 (1990).

9. Haussler D, Krogh A, Mian IS, Sjölander K: Protein modeling using hidden markov models: analysis of globins. In: *Proceedings for the 26th Hawaii International Conference on Systems Sciences*. Wailea HI U.S.A.: Los Alamitos CA: IEEE Computer Society Press (1993).
10. Luthy R, Xenarios I, Bucher P: Improving the sensitivity of the sequence profile method. *Protein Sci.* 3(1), 139-146 (1994).
11. Bateman A, Birney E, Durbin R, Eddy SR, Howe KL, Sonnhammer E: The Pfam protein families database. *Nucleic Acids Res.* 28(1), 263-266 (2000).
12. Altschul SF, Madden TL, Schaffer AA *et al.*: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* 25(17), 2289-3402 (1997).
13. Garnier J, Gibrat J-F, Robson B: GOR method for predicting protein secondary structure from amino acid sequence. *Meth. Enzymol.* 266, 540-553 (1996).
14. Jones DT: Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.* 292(2), 195-202 (1999).
15. Rost B: Review: protein secondary structure prediction continues to rise. *J. Struct. Biol.* 134(2-3), 204-18 (2001).
16. Goebel U, Sander C, Schneider R, Valencia A: Correlated mutations and residue contacts in proteins. *Proteins: Structure Function and Genetics* 18(4), 309-317 (1994).
17. Gutell RR, Weiser B, Woese CR, Noller HF: Comparative anatomy of 16S-like ribosomal RNA. *Prog. Nucleic Acid Res. Mol. Biol.* 32, 155-216 (1985).
18. Morgenstern B, Dress A, Wener T: Multiple DNA and protein sequence based on segment-to-segment comparison. *Proc. Natl. Acad. Sci. USA* 93, 12098-12103 (1996).

•• **The first method described that does not require arbitrary gap penalties.**

19. Lawrence CE, Altschul SF, Boguski MS, Liu JS, Neuwald AF, Wootton JC: Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science* 262, 208-214 (1993).
20. Depiereux E, Baudoux G, Briffeuil P *et al.*: Match-Box_server: a multiple sequence alignment tool placing emphasis on reliability. *Comput. Appl. Biosci.* 13(3), 249-56 (1997).
21. Schuler GD, Altschul SF, Lipman DJ: A workbench for multiple alignment construction and analysis. *Proteins* 9(3), 180-90 (1991).
22. Henikoff S: Playing with blocks: some pitfalls of forcing multiple alignments. *The New Biologist* 3(12), 1148-1154 (1991).
23. Karlin S, Bucher P, Brendel V, Altschul SF: Statistical methods and insight for protein and DNA sequences. *Annu. Rev.Biophys. Biophys. Chem.* 20, 175-203 (1991).
24. Altschul SF, Lipman DJ:Trees stars and multiple biological sequence alignment. *SIAM J. Appl. Math.* 49, 197-209 (1989).
25. Altschul SF, Carroll RJ, Lipman DJ: Weights for data related by a tree. *J. Mol. Biol.* 207, 647-653 (1989).
26. Dayhoff MO, Schwarz RM, Orcutt BC: A model of evolutionary change in proteins. Detecting distant relationships: computer methods and results. In: *Atlas of Protein Sequence and Structure* Dayhoff MO (Eds.), xNational Biomedical Research Foundation: Washington D.C. USA 353-358 (1979).
27. Vingron M, Waterman MS: Sequence alignment and penalty choice. *J. Mol. Biol.* 235, 1-12 (1994).
28. Lipman DJ, Altschul SF, Kececioglu JD: A tool for multiple sequence alignment. *Proc. Natl. Acad. Sci. USA* 86, 4412-4415 (1989).
29. Thompson 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).

•• **The most widely used method for making multiple sequence alignments.**

30. Gotoh O: Further improvement in methods of group-to-group sequence alignment with generalized profile operations. *Comput. Appl. Biosci.* 10(4), 379-87 (1994).

•• **The first attempt to systematically assess the accuracy of a MSA method by comparison wit reference structural alignment. Also the most complex dynamic programming based iterative method.**

31. Thompson JD, Plewniak F, Poch O: A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Res.* 27(13), 2682-2690 (1999).
32. Gonnet GH, Korostensky C, Benner S: Evaluation measures of multiple sequence alignments. *J. Comput. Biol.* 7(1-2), 261-76 (2000).
33. Morgenstern B: DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment [In Process Citation]. *Bioinformatics* 15(3), 211-8 (1999).
34. Notredame C, Higgins DG, Heringa J: T-Coffee: A novel algorithm for multiple sequence alignment. *J. Mol. Biol.* 302, 205-217 (2000).
35. Baldi P, Chauvin Y, Hunkapiller T, McClure MA: Hidden Markov models of biological primary sequence information. *Proc. Nat. Acad. Sci.* 91, 1059-1063 (1994).
36. Karplus K, Hu B: Evaluation of protein multiple alignments by SAM-T99 using the BAliBASE multiple alignment test set. *Bioinformatics* 17(8), 713-20 (2001).
37. Hertz GZ, Stormo GD: Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics* 15(7-8), 563-77 (1999).
38. Wang L, Jiang T: On the complexity of multiple sequence alignment. *J. Comput. Biol.* 1(4), 337-348 (1994).
39. Stoye J, Moulton V, Dress AW: DCA: an efficient implementation of the divide-and-conquer approach to simultaneous multiple sequence alignment. *Comput. Appl. Biosci.* 13(6), 625-6 (1997).
40. Higgins DG, Sharp PM: Clustal: a package for performing multiple sequence alignment on a microcomputer. *Gene* 73, 237-244 (1988).
41. Corpet F: Multiple sequence alignment with hierarchical clustering. *Nucleic Acids Res.* 16, 10881-10890 (1988).
42. Hogeweg P, Hesper B: The alignment of sets of sequences and the construction of phylogenetic trees. An integrated method. *J. Mol. Evol.* 20, 175-186 (1984).

• **The first description of the progressive algorithm.**

43. Feng D-F, Doolittle RF: Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol.* 25, 351-360 (1987).
44. Taylor WR: A flexible method to align large numbers of biological sequences. *J. Mol. Evol.* 28, 161-169 (1988).
45. Needleman SB, Wunsch CD: A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* 48, 443-453 (1970).
46. Barton GJ, Sternberg MJE: A strategy for the rapid multiple alignment of protein sequences: confidence levels from tertiary structure comparisons. *J. Mol. Biol.* 198, 327-337 (1987).
47. Gotoh O: Significant Improvement in Accuracy of Multiple Protein Sequence Alignments by Iterative Refinements as Assessed by Reference to Structural Alignments. *J. Mol. Biol.* 264(4), 823-838 (1996).
48. Heringa J: Two strategies for sequence comparison: profile-preprocessed and secondary structure-induced multiple

alignment. *Computers and Chemistry* 23, 341-364 (1999).

49. Krogh A, Brown M, Mian IS, Sjölander K, Haussler D: Hidden Markov Models in Computational Biology: Applications to Protein Modeling. *J. Mol. Biol.* 235, 1501-1531 (1994).
50. Kim J, Pramanik S, Chung MJ: Multiple Sequence Alignment using Simulated Annealing. *Comp. Applic. Biosci*. 10(4), 419-426 (1994).
51. Notredame C, Higgins DG: SAGA: sequence alignment by genetic algorithm. *Nucleic Acids Res*. 24, 1515-1524 (1996).

• **One of the first attempts to apply genetic algorithms to sequence analysis.**

52. Zhang C, Wong AK: A genetic algorithm for multiple molecular sequence alignment. *Comput. Appl. Biosci.* 13(6), 565-81 (1997).
53. Anabarasu LA: Multiple sequence alignment using parallel genetic algorithms. In *The Second Asia-Pacific Conference on Simulated Evolution (SEAL-98).* Canberra Australia (1998).
54. Gonzalez RR: Multiple protein sequence comparison by genetic algorithms. In *SPIE-98* (1999).
55. Chellapilla K, Fogel GB: Multiple sequence alignment using evolutionary programming. In: *Congress on Evolutionary Computation.* (1999).
56. Cai L, Juedes D, Liakhovitch E: Evolutionary computation techniques for multiple sequence alignment. In: *Congress on Evolutionary Computation.* (2000).
57. Duret L, Abdeddaim S: Multiple Alignment for Structural Functional or phylogenetic analyses of Homologous Sequences. In: *Bioinformatics Sequence structure and databanks*. Higgins D, Taylor W (Eds.), Oxford University Press: Oxford UK (2000).
58. Durbin R *et al.*: *Biological Sequence Analysis*. Cambridge University Press. Cambridge, UK (1998).

•• **One of the most comrehensive textbook on the algorithms dedicated to sequence analysis.**

59. Devereux J, Haeberli P, Smithies O: GCG package. *Nucleic Acids Res.* 12, 387-395 (1984).
60. Carrillo H, Lipman DJ: The multiple sequence alignment problem in biology. *SIAM J. Appl. Math.* 48, 1073-1082 (1988).
61. Reinert K, Stoye J, Will T: An iterative method for faster sum-of-pair multiple sequence alignment. *Bioinformatics* 16(9), 808-814 (2000).
62. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E: Equation of state calculations by fast computing machines. *J. Chem. Phys.* 21, 1087-1092 (1953).
63. Holland JH: Adaptation in natural and artificial systems. Ann Arbour MI: University of Michigan Press (1975).
64. Ishikawa M, Toya T, Hoshida M, Nitta K, Ogiwara A, Kanehisa M: Multiple sequence alignment by parallel simulated annealing. *Comp. Applic. Biosci.* 9, 267-273 (1993).
65. Goldberg DE: Genetic Algorithms. In: Search Optimization and Machine Learning. Goldberg DE (Eds), Addison-Wesley. New York USA (1989).
66. Notredame C, Holm L, Higgins DG: COFFEE: an objective function for multiple sequence alignments. *Bioinformatics* 14(5), 407-422 (1998).
67. Notredame C, O'Brien EA, Higgins DG: RAGA: RNA sequence alignment by genetic algorithm. *Nucleic Acids Res.* 25(22), 4570-4580 (1997).
68. Eddy SR: Multiple alignment using hidden Markov models. In: *Third international conference on intelligent systems for molecular biology (ISMB).* Cambridge England: Menlo Park CA: AAAI Press (1995).
69. Berger MP, Munson PJ: A novel randomized iterative strategy for aligning multiple protein sequences. *Comput. Appl. Biosci.* 7, 479-484 (1991).
70. Brocchieri L, Karlin S: Asymetric-iterated multiple alignment of protein sequences. *JMB* 276, 249-264 (1998).
71. Kececioglu JD: The maximum weight trace problem in multiple sequence alignment. Lecture notes in *Computer Science* 684, 106-119 (1983).
72. Vingron M, Argos P: Motif recognition and alignment for many sequences by comparison of dot-matrices. *J. Mol. Biol.* 218, 33-43 (1991).
73. Morgenstern B, Frech BK, Dress A, Werner T: DIALIGN: finding local similarities by multiple sequence alignment. *Bioinformatics* 14(3), 290-294 (1998).
74. Huang X, Miller W: A time-efficient linear-space local similarity algorithm. *Adv. Appl. Math.* 12, 337-357 (1991).
75. Bucka-Lassen K, Caprani O, Hein J: Combining many multiple alignments in one improved alignment. *Bioinformatics*. 15(2), 122-130 (1999).
76. Lenhof HP, Morgenstern B, Reinert K: An exact solution for the segment-to-segment multiple sequence alignment problem. *Bioinformatics* 15(3), 203-210 (1999).
77. Jennings AJ, Edge CM, Sternberg MJ: An approach to improving multiple alignments of protein sequences using predicted secondary structure. *Protein Eng.* 14(4), 227-31 (2001).
78. Dietmann S, Park J, Notredame C, Heger A, Lappe M, Holm L: A fully automatic evolutionary classification of protein folds: dali domain dictionary version 3. *Nucleic Acids Res.* 29(1), 55-7 (2001).
79. Taylor WR, Saelensminde G, Eidhammer I: Multiple protein sequence alignment using double-dynamic programming. *Comput. Chem.* 24(1), 3-12 (2000).
80. Orengo CA, Taylor WR: A rapid method of protein structure alignment. *J. Theor. Biol.* 147, 517-551 (1990).
81. Eidhammer I, Jonassen I, Taylor WR: Structure comparison and structure patterns. *J. Comput. Biol.* 7(5), 685-716 (2000).
82. Notredame C: Mocca: semi-automatic method for domain hunting. *Bioinformatics* 17(4), 373-374 (2001).
83. Heringa J, Argos P: A method to recognzse distant repeats in protein sequences. *Proteins: Structure Function and Genetics* 17, 391-411 (1993).
84. Hughey R, Krogh A: Hidden Markov models for sequence analysis: extension and analysis of the basic method. *Computer Applications in Biological Science* 12, 95-107 (1996).