25

Protein Multiple Sequence Alignment

Chuong B. Do and Kazutaka Katoh

Summary

Protein sequence alignment is the task of identifying evolutionarily or structurally related positions in a collection of amino acid sequences. Although the protein alignment problem has been studied for several decades, many recent studies have demonstrated considerable progress in improving the accuracy or scalability of multiple and pairwise alignment tools, or in expanding the scope of tasks handled by an alignment program. In this chapter, we review state-of-the-art protein sequence alignment and provide practical advice for users of alignment tools.

Key Words: Multiple sequence alignment; review; proteins; software.

1. Introduction

Sequence alignment is a standard technique in bioinformatics for visualizing the relationships between residues in a collection of evolutionarily or structurally related proteins (*see* **Note 1**). Given the amino acid sequences of a set of proteins to be compared, an alignment displays the residues for each protein on a single line, with gaps ("–") inserted such that "equivalent" residues appear in the same column. The precise meaning of equivalence is generally context dependent: for the phylogeneticist, equivalent residues have common evolutionary ancestry; for the structural biologist, equivalent residues correspond to analogous positions belonging to homologous folds in a set of proteins; for the molecular biologist, equivalent residues play similar functional roles in their corresponding proteins. In each case, an alignment provides a bird's eye view of the underlying evolutionary, structural, or functional constraints characterizing a protein family in a concise, visually intuitive format.

From: *Methods in Molecular Biology, vol. 484: Functional Proteomics: Methods and Protocols*
Edited by: J. D. Thompson et al., DOI: 10.1007/978-1-59745-398-1, © Humana Press, Totowa, NJ

In this chapter, we review state-of-the-art techniques for protein alignment. The literature is vast, and hence our presentation of topics is necessarily selective (*see* **Note 2**). Here, we address the problem of alignment construction: we survey the range of practical techniques for computing multiple sequence alignments, with a focus on practical methods that have demonstrated good performance on real-world benchmarks. We discuss current software tools for protein alignment and provide advice for practitioners looking to get the most out of their multiple sequence alignments.

2. Algorithms

Most modern programs for constructing multiple sequence alignments (MSAs) consist of two components: an *objective function* for assessing the quality of a candidate alignment of a set of input sequences, and an *optimization procedure* for identifying the highest scoring alignment with respect to the chosen objective function ***(1)***. In this section, we describe common themes in the architecture of modern MSA programs (*see* **Fig. 1**).

2.1. The Sum-of-Pairs Scoring Model

In the problem of pairwise sequence alignment, the score of a candidate alignment is typically defined as a summation of *substitution scores*, for matched

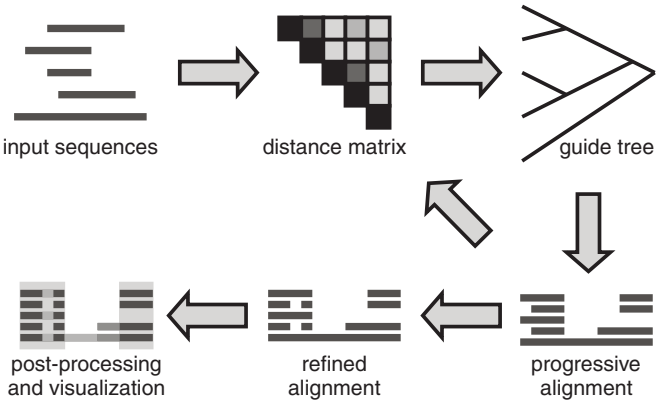

Fig. 1. Diagram of the basic steps in a prototypical modern multiple sequence alignment program: computation of matrix of distances between all pairs of input sequences; estimation of phylogenetic guide tree based on distance matrix; progressive alignment according to guide tree; guide tree reestimation and realignment; iterative refinement; and postprocessing and visualization.

pairs of characters in the sequences being aligned, and *gap penalties*, for consecutive substrings of gapped characters. Given a fixed set of scoring parameters, efficient dynamic programming algorithms (*see* **Note 3**) for computing the optimal alignment of two sequences in quadratic time and linear space have been known since the early 1980s ***(2–5)***.

In the case of multiple sequence alignment for N sequences, the multiple alignment score is usually defined to be the summed scores of all $N(N-1)/2$ pairwise projections of the original candidate MSA to each pair of input sequences. This is known as the standard *sum-of-pairs* (SP) scoring model ***(6)***. While other alternatives exist, such as consensus ***(7)***, entropy ***(8)***, or circular sum ***(9)*** scoring, most alignment methods rely on the SP objective and its variants. Unlike the pairwise case, multiple sequence alignment under the SP scoring model is NP-complete ***(10–13)***; direct dynamic programming methods for multiple alignment require time and space exponential in N.

Some strategies for dealing with the exponential cost of multiple alignment involve pruning the space of candidate multiple alignments. The "MSA" program ***(14,15)***, for instance, uses the Carrillo–Lipman bounds ***(16)*** in order to determine constraints on an optimal multiple alignment based on the projections of the alignment to all pairs of input sequences; similarly, the DCA program ***(17–21)*** employs a divide-and-conquer approach that uses pairwise projected alignments to identify suitable "cut" points for partitioning a large multiple alignment into smaller subproblems. In practice, however, these methods are impractical for more than a few sequences. Consequently, most current techniques for SP-based multiple alignment work by either applying heuristics to solve the original NP-complete optimization problem approximately, or replacing the SP objective entirely with another objective whose optimization is tractable.

2.2. Global Optimization Techniques

In general, finding a mathematically optimal multiple alignment of a set of sequences can be formulated as a complex optimization problem: given a set of candidate MSAs, identify the alignment with the highest score. Global optimization techniques, developed in applied mathematics and operations research, provide a generic toolbox for tackling complex optimization problems. Over the past several decades, application of these methods to the MSA problem has become routine.

Among these methods, *genetic algorithms* ***(22)***—which maintain a population of candidate alignments that are stochastically combined and mutated through a directed evolutionary process—have been particularly popular ***(23–28)***. In this technique, the SP objective (or an approximation thereof) provides a measure

of fitness for individual alignments within the population. Typical mutation operations involve local insertion, deletion, or shuffling of gaps; designing these operations in a manner that allows fast traversal of the space of candidate alignments while remaining efficient to compute is the main challenge in the development of effective genetic algorithm approaches for MSA. Sequence alignment programs based on genetic algorithms include SAGA ***(29)***, MAGA ***(30–33)***, and PHGA ***(34)***.

In *simulated annealing* ***(35)***, a candidate alignment is also iteratively modified via local perturbations in a stochastic manner, which tends toward alignments with high SP scores ***(36–38)***. Unlike genetic algorithms, simulated annealing approaches do not maintain a population of candidate solutions; rather, modifications made to candidate solutions may either improve or decrease the objective function, and the probability of applying a particular modification to a candidate alignment is dependent both on the resulting change in SP score and on a scaling constant known as the temperature. In theory, when using appropriately chosen temperature schedules, simulated annealing provably converges to optimal MSAs. The number of iterations required to reach an optimal alignment with appreciable probability, however, can often be exponentially large. The MSASA ***(37)*** program for simulated annealing-based alignment overcomes this barrier by using multiple alignments obtained via progressive alignment (described later) as a starting point.

Search-based strategies form a third class of global optimization techniques that have been applied to multiple alignment. In these methods, multiple alignment is typically formulated as a shortest path problem, where the initial state is the empty alignment (containing no columns), goal states are the set of all possible alignments of the given sequences, intermediate states represent candidate partial alignments of sequence prefixes, and state transition costs represent the change in score resulting from the addition of a column to an existing partial alignment. Despite the large state space, search techniques such as A* and branch-and-bound use heuristics to prune the set of searched alignments ***(39,40)***. The MSA ***(14,15)*** and DCA/OMA ***(17,19–21,41–43)*** programs are two examples of methods based on this strategy.

2.3. Progressive Alignment

While global optimization techniques are powerful in their general applicability, they are less commonly used in modern MSA programs due to their computational expense (*see* **Note 4**). In this section, we examine a heuristic, known as *progressive alignment*, that solves the intractable problem of MSA approximately via a sequence of tractable subproblems. Unlike the techniques discussed in the last section, which find good multiple alignments directly,

progressive alignment works indirectly, relying on variants of known algorithms for pairwise alignment.

In the popular *progressive alignment* strategy ***(44–46)***, the sequences to be aligned are each assigned to separate leaves in a rooted binary tree (known as an alignment guide tree, *see* **Section 2.4.1**). Next, the internal nodes of the tree are visited in a bottom-up order, and each visited node is associated with an MSA of the sequences in its corresponding subtree. At the end of the traversal, the MSA associated with the root node is returned. By restricting MSAs at each internal node to preserve the aligned columns in the MSAs associated with their children nodes, the overall procedure reduces to a sequence of pairwise alignment computations: here, each pairwise alignment operates on a pair of alignments rather than a pair of sequences.

Under the most common gap scoring schemes, aligning a pair of alignments to optimize the SP score exactly is theoretically NP-hard ***(47)***. Here, the complication arises from the fact that a gap opening character for some sequence in an MSA may not necessarily be present in every projected pairwise alignment involving that sequence. In practice, aligning alignments can be accomplished via procedures that optimize upper or lower bounds on the SP score ***(48)***, which use a "quasinatural gap" approximation to the full SP score ***(49)***, or which approximate each set of input alignments as a *profile*—a matrix of character frequencies at each position in the alignment ***(50,51)***. Progressive alignment is the foundation of several alignment programs including DFALIGN ***(44)***, MULTAL ***(45,46)***, MAP ***(52)***, PCMA ***(53)***, PIMA ***(54)***, PRIME ***(55)***, PRRP ***(56)***, MULTALIN ***(57)***, CLUSTALW ***(58–60)***, MAFFT ***(50,61)***, MUSCLE ***(51, 62)***, T-Coffee ***(63,64)***, KAlign ***(65)***, POA ***(66–68)***, PROBCONS ***(69)***, and MUMMALS/PROMALS ***(70,71)***.

Profile–profile alignment techniques are routinely used in classification tasks such as remote homology detection and fold recognition ***(72–75)***. In this literature, a considerable amount of effort has been placed in identifying profile–profile scoring functions that discriminate well between weakly homologous sequences and nonhomologous sequences ***(76–81)***. While one might expect that a profile–profile scoring function that works well for classification should give accurate multiple sequence alignments, empirical tests have revealed only minor differences in alignment quality resulting from various profile–profile scoring schemes ***(62,82–84)***.

2.4. Extensions to Progressive Alignment

The efficiency and simplicity of progressive algorithms for sequence alignment account for their widespread use in modern sequence alignment

tools. Given a guide tree over N sequences, MSA construction requires $N - 1$ pairwise merge steps, hence rendering the cost of alignment effectively linear in the number of sequences (*see* **Note 5**). Nonetheless, progressive alignment strategies may also suffer from inaccuracies in the constructed guide trees or the accumulation of errors from the early pairwise alignment stages. In this section, we describe a number of heuristics used in modern MSA programs to overcome the shortcomings of vanilla progressive alignment.

2.4.1. Guide Tree Construction

In most progressive alignment programs, the guide tree used to determine the merging order for sequence groups is taken to be the phylogenetic tree relating the input sequences. Distance matrix methods for tree construction, such as the UPGMA ***(85,86)*** or neighbor-joining ***(87,88)*** algorithms, work by first estimating the evolutionary time between each pair of sequences. Then, a greedy procedure is used to construct a tree whose edge lengths correspond to evolutionary distances between points of divergence in the evolutionary history of the input sequences.

Problems with alignment guide trees generally result from either errors in the computed distance matrices or violated assumptions associated with the used tree reconstruction technique. The former case is especially common as many modern multiple alignment programs (e.g., MUSCLE, MAFFT, and MUMMALS/PROMALS) use fast approximate distance measures, such as *k*-mer counting, to form distance matrices for progressive alignment ***(50,58,89,90)***. Replacing these measures with more sensitive distance-estimation methods based on full pairwise alignment can be effective but slow ***(60)***. Recently, the Wu–Manber algorithm for fast inexact string matching ***(91)***, as employed in the KAlign program, has been shown to be significantly more sensitive than simple *k*-mer approaches for especially distant sequences ***(65)***.

Alternatively, *guide tree reestimation* can be effective for obtaining more accurate distance measures; given an approximate multiple alignment generated from the progressive alignment algorithm, it is generally possible to compute evolutionary trees of higher quality than the original guide trees formed using simple distance measures ***(50,56)***. In practice, alignment programs that use guide tree reestimation (e.g., MAFFT, MUSCLE, PRIME, PRRP, and MUMMALS/PROMALS) compute new distance matrices using an MSA obtained by progressive alignment. This revised distance matrix is then used to construct a new guide tree, which is in turn used in a second round of progressive alignment. The procedure may be iterated as many times as desired (or until convergence).

2.4.2. Modified Objective Functions

Even with perfect guide trees, errors can still occur in the pairwise merge steps of the progressive alignment. Errors made at early stages of the progressive alignment are particularly detrimental as they provide a distorted view of sequence homology that increases the chances of incorrect pairwise alignments at all higher levels of the tree. C*onsistency-based* objective functions focus on improved scoring of matches in early alignments by incorporating information from outgroup sequences during each pairwise merge step ***(92–95)***. In particular, when performing a pairwise alignment of two sequences *x* and *y*, knowing that the *k*th residue of an outgroup sequence *z* aligns well with the *i*th residue of *x* and the *j*th residue of *y* provides strong evidence that the *i*th position of *x* and *j*th position of *y* should align with each other—i.e., pairwise alignments induced by a multiple alignment should be consistent (*see* **Fig. 2A**). Based on this transitivity condition, consistency-based objective functions typically modify the score for matching positions in an alignment of two groups during pairwise alignment by considering the relationship of each group to sequences not involved in the pairwise merge. Consistency-based scoring is used in the T-Coffee, DIALIGN, PROBCONS, PCMA, MUMMALS, PROMALS, and Align-m ***(96,97)*** alignment algorithms.

A number of modern programs (e.g., CLUSTALW, MUSCLE, and MAFFT) also use *position-specific gap penalties* to bias alignment algorithms toward placing gaps where previous gaps were opened during each pairwise merge step. Here, the rationale is that gap opening events that occur simultaneously in a group of sequences likely represent a single evolutionary event and hence should not be overpenalized. In addition, for globular protein sequences, hydrophobic residues are abundant in core regions where sequence indels are likely to affect proper folding, whereas hydrophilic residues are abundant on the protein surface, where extra loops are more likely to be tolerated (*see* **Fig. 2B**). CLUSTALW and MUSCLE attempt to make use of this signal by heuristically increasing gap penalties in hydrophobic regions and decreasing them in hydrophilic regions, though in practice the impact of hydropathy-based scoring on these methods is small. Recently, however, the CONTRAlign program ***(98)*** has demonstrated that rigorous statistical estimation of hydropathy-based gap penalty modifications can result in improvements in alignment accuracy of several percent for distant sequences; similar results have also been observed for detection of homology via profile alignments ***(99)***.

Sequence weighting is another common modification of the traditional SP multiple alignment objective applicable when the representation of sequence subgroups in a multiple alignment is highly skewed (*see* **Fig. 2C**). For

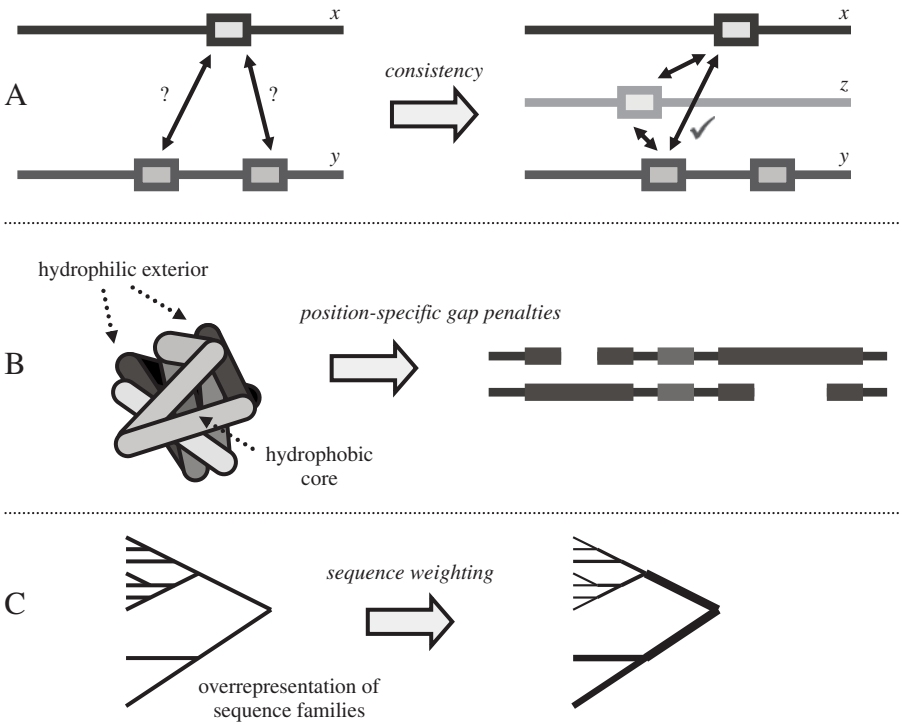

Fig. 2. Modified objective functions for sum-of-pairs alignment. (**A**) To aid in the alignment of two sequences *x* and *y*, consistency-based aligners use alignments of *x* and *y* to a third sequence *z*. (**B**) Gaps occur more frequently in the hydrophilic exterior than the hydrophobic core of globular proteins; position-specific gap penalties are higher in regions with hydrophobic residues and lower in regions with hydrophilic residues. (**C**) Sequence weighting corrects for sequence family overrepresentation.

example, in a multiple alignment of *K* sequences, if a large number of copies of a single sequence are added to the input, then an unweighted SP optimizer will emphasize the alignments of the redundant sequence to the other *K* – 1 sequences, thus effectively generating a biologically incorrect star alignment. While numerous schemes for computing sequence weights exist ***(92,100–108)***, the best choice of weights for alignment programs is unclear. In practice, the exact choice of weighting technique is generally a second-order effect; most reasonable sequence weighting techniques can greatly improve the accuracy of alignments in situations of sequence overrepresentation.

2.4.3. Postprocessing

In many cases, no amount of preprocessing is sufficient to prevent errors during progressive alignment. Postprocessing procedures, generally known as *iterative refinement* techniques, deal with progressive alignment errors by making changes to an existing alignment obtained from progressive alignment. For instance, iterative realignment techniques work by repeatedly dividing an alignment into two groups of aligned sequences, and realigning the groups ***(56, 109–111)***. In practice, iterative realignment can greatly improve the quality of an existing multiple alignment while requiring little extra programming effort. Alignment programs that make use of iterative realignment procedures include ITERALIGN ***(112)***, TULLA ***(113)***, AMPS/AMULT ***(114,115)***, MULTAN ***(116)***, OMA ***(42)***, PRRP, PROBCONS, MUSCLE, and MAFFT.

Other refinement techniques focus on correcting local errors in alignments by pattern matching or stochastic optimization, and bear strong similarity to the global optimization strategies introduced earlier ***(110,117–119)***. While global optimization techniques are generally considered less efficient than heuristic strategies such as progressive alignment in constructing multiple alignments, they can, nonetheless, be extremely effective given a good initial starting point (i.e., an existing multiple alignment).

2.5. Local Alignment

Most protein sequence alignment tools make the implicit assumption of *global homology*—the assumption that the sequences being aligned are generally related over their entire length. In many practical situations, however, two proteins may simply share a few common domains interspersed with regions of little to no homology. In these scenarios, variants of dynamic programming can be used for pairwise alignment ***(3)***. A space-efficient formulation of the dynamic programming algorithm, in particular, forms the basis of the SIM and LALIGN pairwise local alignment programs ***(120,121)***.

When speed is essential, indexing-based techniques can also be used for local alignment. These methods work by identifying segments of fixed length (known as *seeds* or *k-mers*) that are shared between two sequences; seeds meeting a certain threshold score are either chained or extended to form local alignments. This strategy is employed by the BLASTP ***(122,123)*** and LFASTA ***(124–126)*** programs.

For the problem of multiple local alignment, the DIALIGN ***(127–130)*** and DIALIGN-T ***(131)*** programs work by identifying homologous ungapped segments using a unique probabilistic segment scoring system that does not explicitly penalize for indels. Segments are then selected for inclusion in the multiple alignment via a greedy procedure that requires conserved segments to

be present in the same order in each sequence. Related procedure for finding conserved "boxes" or for identifying high-confidence matches are used in the MATCH-BOX ***(132,133)*** and AMAP ***(134)*** programs.

In some proteins, however, conserved domains may appear multiple times in a single sequence (known as repeats) or may appear in a different order in different sequences (known as rearrangements). Repeated domains can generally be identified via local alignment of a sequence to itself ***(135)***; programs that specialize in the identification and alignment of protein repeats include Mocca ***(136)***, RADAR ***(137)***, REPRO ***(138)***, and TRUST ***(139)***. A more recent program called RAlign ***(140)*** performs global alignments while taking into account repeat structure.

Constructing multiple local alignments with both repeats and rearrangements is an extremely difficult problem that is usually done manually. Motif finders, such as GIBBS ***(141,142)***, MOTIF ***(143,144)***, MEME ***(145)***, and CONSENSUS ***(141)***, in principle can detect local ungapped homologies between several protein sequences. In practice, however, these methods are usually slow and can find only short, well-conserved gap-free segments of fixed length. Existing domain finding programs, such as DOMAINER ***(146)*** and MACAW ***(147)***, have similar restrictions, and the latter also requires significant manual intervention. Recently, a number of programs have addressed the challenges of representing multiple local alignments of protein sequences using partial-order ***(66)*** and A-Bruijn ***(148)*** graphs; some recent attempts to completely automate multiple local alignment construction include the ABA ***(149)*** and ProDA ***(150)*** alignment tools.

2.6. Probabilistic Models

While most alignment techniques rely abstractly on a scoring scheme that uses substitution scores and gap penalties, they do not develop an explicit model of the evolutionary process. In this section, we consider the class of probabilistic methods for aligner construction that has garnered much recent interest. Probabilistic techniques for multiple alignment generally come in three main varieties: complex evolutionary models of insertion, deletion, and mutation in multiple sequences; fixed dimensionality profile models for representing specific protein families; and hybrid methods that combine probabilistic models with traditional ad hoc alignment techniques.

Of the three approaches, evolutionary models for statistical alignment provide the most explicit representation of change in biological sequences as a stochastic process ***(151,152)***. Research in statistical alignment typically derive from the classic Thorne–Kishino–Felsenstein (TKF) pairwise alignment model ***(153)*** in which amino acid substitutions follow a time-reversible Markov process

and single-gap creation and deletion are treated as birth/death processes over imaginary "links" separating letters in a sequence. Subsequent work on statistical alignment has focused on modeling multiresidue, overlapping indels ***(154–159)***, extending the TKF model to multiple alignment ***(160–167)***, and the even more complex task of coestimating alignment and sequence phylogeny ***(164,168–172)***. Unlike traditional score-based alignment approaches, statistical alignment methods provide a natural framework for estimating the parameters underlying stochastic evolutionary processes ***(173)***. However, the resulting models are often quite complex. While dynamic programming is sometimes possible, these models often require sampling-based inference procedures ***(174)*** that share many of the disadvantages of simulated annealing approaches discussed earlier. The accuracy of TKF-based techniques in alignment construction is unclear as few methods based on this approach have been comparatively benchmarked against standard programs; one exception is the Handel ***(162,163)*** program for statistical multiple alignment, which achieves substantially lower accuracy (i.e., 13% fewer correctly aligned residue pairs) than CLUSTALW, the prototypical score-based modern sequence aligner.

A second class of probabilistic modeling techniques is the profile hidden Markov model (profile HMM), a sophisticated variant of the character frequency profile matrices that takes into account position-specific indel probabilities ***(8,175–179)***. To construct a profile HMM given a set of unaligned sequences, a length is chosen for the initial profile, as well as initial emission probabilities for each position in the profile and transition probabilities for indel creation and extension after each position. Next, the model is optimized according to a likelihood criterion using an expectation–maximization (EM)-based Baum–Welch procedure ***(8)***, simulated annealing ***(38)***, deterministic annealing ***(180)***, or approximate gradient descent ***(181,182)***. Finally, all sequences are aligned to the profile using the Viterbi algorithm ***(183)*** for finding the most likely correspondence between each individual sequence and the profile, and the correspondences of each sequence to the profile are accumulated to form the multiple alignment. Profile HMMs and their variants ***(184)*** form the basis of many remote homology detection techniques ***(185–187)*** and have been used to characterize protein sequence families ***(188)***. Empirically, profile HMMs ***(177,189)*** have great appeal in practice as they provide a principled probabilistic framework, and, when properly tuned ***(190,191)***, achieve good empirical performance close to that of CLUSTALW ***(192,193)***.

Finally, hybrid techniques combine the rigor of probabilistic model parameter estimation with standard heuristics for multiple alignment. The ProAlign ***(194)***, COACH ***(81)***, and SATCHMO ***(195,196)*** progressive alignment tools, for instance, all achieve CLUSTALW accuracy; the recent PRANK aligner ***(197)*** has revealed the benefits of scoring insertions and deletions differently for the

purposes of indel distribution estimation. A separate promising direction has been the development of the maximum expected accuracy (MEA) algorithm for pairwise alignment based on posterior match probabilities ***(198)***, which was generalized to consistency-based multiple alignment in the PROBCONS algorithm ***(69)***. Other programs based on the public domain PROBCONS source code include AMAP ***(199)***, which optimizes an objective function that rewards for correctly placed gaps, and ProbAlign ***(200)***, which uses a physics-inspired modification of the posterior probability calculations in PROBCONS. Finally, the MUMMALS program ***(70)***, which extends the PROBCONS approach to allow for more sophisticated HMM structures, has achieved the highest reported accuracies to date of all modern stand-alone multiple alignment programs.

2.7. Computation-Intensive Methods

In recent years, a new category of computation-intensive methods has risen in importance. Typically, these methods are designed not for high-throughput scenarios but rather for situations in which accuracy is paramount and abundant computing resources are available. Such scenarios arise in protein structure prediction, where alignment quality is the bottleneck in fold prediction accuracy, and the need for high-speed alignment is less important.

Ensemble methods (often known as meta-prediction methods in the protein structure prediction community) consider the predictions of a number of separate individual methods in order to form an aggregate prediction. M-Coffee ***(201)*** places input alignments into an alignment library and then assembles a multiple alignment using the T-Coffee progressive algorithm for solving the maximum-weight trace problem ***(202–204)***. A similar program called meta_align is also available as part of the MUMMALS package ***(70)***. In both cases, the resulting alignments generated by the ensemble predictor are more accurate than those made by any individual prediction technique.

Finally, *database-aided methods* add external information to help the aligner resolve ambiguities in alignment decisions. For instance, adding homologous sequences found in a large sequence database when the number of input sequences is small has been shown to be effective for methods such as MAFFT, PRALINE ***(205,206)***, and DbClustal ***(207)***. Alternatively, adding extra experimental or predicted information regarding the structural properties of the sequences being aligned can also improve accuracy. For example, the NdPASA ***(208)***, HHAlign ***(75)***, and PrISM.1 ***(209)*** pairwise aligners and the PSI-PRALINE ***(205)*** and SPEM ***(210)*** multiple aligners all make use of known or predicted secondary structure; similarly, the 3D-Coffee ***(211,212)*** multiple aligner incorporates structural alignments when they are available. In general, the specific program used for performing the alignment tends to be less

important than the data incorporated by each alignment approach. Given this, the best database-aided method to use in any given alignment situation should generally be based on the data available.

3. Other Considerations

In studies of multiple sequence alignment, the algorithms used can be important, but they are not the only consideration that must be made. In this section, we provide a brief overview of aligner performance assessment and recent developments in parameter estimation.

3.1. Benchmarking

Techniques for assessing aligner performance typically have one of four goals: (1) demonstrating the effectiveness of a particular heuristic strategy for SP objective optimization; showing that a particular software package achieves good accuracy relative to "gold standard" reference alignments of either (2) real or (3) simulated proteins; or (4) quantifying alignment accuracy on real data in a reference-independent manner. For comparing software packages relying on different objective functions, the first validation scheme is not applicable. In this subsection, we focus on the latter three methods of aligner validation.

In real protein sequences, the true alignment of a set of sequences based on structural considerations is not necessarily the same as the true alignment based on evolutionary or functional considerations. In practice, structural alignments are relatively easy to obtain for proteins of known structure, and hence, are the *de facto* standard in most real-world benchmarks of alignment tools. Popular databases of hand-curated structural alignments include BAliBASE version 2 ***(213,214)*** and HOMSTRAD ***(215)***. Because of the difficulty and lack of reproducibility of hand curation, a number of modern alignment databases rely on automated structural alignment protocols, including SABmark ***(216)***, PREFAB ***(51)***, OxBench ***(217)***, and to a large extent, BAliBASE version 3 ***(218)***. Because the correct protein structural alignment can sometimes also be ambiguous, most alignment databases annotate select portions of their provided alignments as "core blocks"—regions for which structural alignments are known to be reliable—and measures of accuracy such as the Q score [defined as the proportion of pairwise matches in a reference alignment predicted by the aligner; other measures of accuracy also exist ***(219)***] are computed with respect to only core blocks.

The difficulties of ambiguity in structural alignments can be avoided when benchmarking with simulated evolution programs, such as SIMPROT ***(220,221)*** or Rose ***(222)***. In simulation studies, the true "evolutionary" relationships

between positions in a set of a sequences are completely known. Besides allowing for the construction of large testing sets, simulation-based validation also has the advantage of enabling detailed studies of aligner performance in specific settings; for example, the IRMBase database ***(131)***, created using the Rose simulator, was built to evaluate the ability of local alignment methods to identify short implanted conserved motifs within nonhomologous sequences. Despite these advantages, simulation studies are highly prone to parameter overfitting. Furthermore, the performance of a method on simulated proteins may not be representative of its performance on real proteins, especially if the simulator fails to properly model all of the biological features used by the aligner. For instance, a method that accounts for gap enrichment in hydrophilic regions of proteins will perform relatively worse on simulations that do not account for hydropathy properties of protein sequences than on real proteins for which hydropathy plays an important role.

Finally, it is possible to avoid dealing with ambiguities in reference alignments using techniques that directly assess the quality of an alignment in terms of the resulting structural superposition. For a pair of proteins, the coordinate root-mean-square-distance (coordinate RMSD) between positions identified as "equivalent" according to an alignment (after the two protein structures have been appropriately rotated and translated) is a common measure for evaluating structural alignment quality. Several RMSD variants exist ***(223)***, including variants that account for protein length ***(224)***, that examine pairwise distances between residues in a protein ***(225)***, or that rely on alternate representations of protein backbones ***(226)***. Another recently proposed metric is the APDB measure ***(227)***, an approximation of the Q score that judges the "correctness" of aligned residue pairs based on the degree to which nearby aligned residues have similar local geometry in the sequences being aligned.

3.2. Parameter Estimation

For traditional score-based sequence alignment procedures, estimation of substitution matrices and gap penalties are usually treated separately (*see* **Note 6**). Briefly, substitution matrices are generally estimated from databases of alignments known to be reliable. Statistical estimation procedures for constructing log-odds substitution matrices vary in their details, but most methods nonetheless tend to generate sets of matrices approximately parameterized by some notion of evolutionary distance for which that matrix is optimal. Popular matrices include the BLOSUM ***(228)***, PAM ***(229,230)***, JTT ***(89)***, MV ***(231)***, and WAG ***(232)*** matrices; matrices derived from structural alignments for use with low-identity sequences also exist ***(233)***. For gap parameters,

an empirical trial-and-error approach ***(234)*** is common as the number of parameters to be estimated is low.

Probabilistic models have the advantage that the maximum likelihood principle provides a natural mechanism for estimating gap parameters when example alignments are available ***(235)***; when only unaligned sequences are available, unsupervised estimation of gap parameters can still be effective ***(69)***. Alternatively, Bayesian methods ***(236,237)*** automatically combine the results obtained when using multiple varying parameter sets and thus avoid the need for deciding on fixed parameter sets.

Recently, the problem of parameter estimation has been the subject of renewed attention, stemming from the influence of the convex optimization and machine learning communities. Kececioglu and Kim ***(238)*** described a simple cutting-plane algorithm for *inverse alignment*—the problem of identifying a parameter set for which an aligner aligns each sequence in a training set correctly. Their algorithm is fast in practice, though the biological accuracy of the resulting alignments on unseen test data is unclear. Do *et al*. ***(98)*** developed a machine learning-based method based on pair conditional random fields (pair-CRFs) called CONTRAlign, which achieves significantly better generalization performance than existing methods for pairwise alignment of distant sequences. Most recently, Yu *et al*. ***(239)*** described a fast approach for training protein threading models based on support vector machines ***(240)***, which shares many of the generalization advantages of CONTRAlign.

4. Advice for Practitioners

Given the multitude of choices, it can be difficult for a user of multiple alignment software to understand the situations in which a particular alignment tool is or is not appropriate. When aligning a small number (<20) of globally homologous sequences with high percent identity ($>40\%$), most modern alignment programs will have no difficulty in returning a correct multiple sequence alignment, and no special consideration is needed. When all of these conditions do not hold, however, choosing the appropriate tools and configuration, while keeping in mind the tradeoff between accuracy and computational cost, can be difficult. In this section, we provide a list of currently popular alignment software (*see* **Table 1**) and give advice on tool selection (*see* **Fig. 3**) and effective use of alignments.

4.1. The Extreme Cases

Extreme cases for sequence alignment programs involve scenarios typically not encountered in most alignment benchmarking studies. The spectrum of

Table 1
MSA Programs

Tool	URL
CLUSTALW	http://www.clustal.org/
DIALIGN	http://bibiserv.techfak.uni-bielefeld.de/dialign/
MAFFT	http://align.bmr.kyushu-u.ac.jp/mafft/software/
MUMMALS	http://prodata.swmed.edu/mummals/
MUSCLE	http://www.drive5.com/muscle/
PRALINE	http://zeus.cs.vu.nl/programs/pralinewww/
PRIME	http://prime.cbrc.jp/
ProbAlign	http://probalign.njit.edu/standalone.html
PROBCONS	http://probcons.stanford.edu/
ProDA	http://proda.stanford.edu/
PROMALS	http://prodata.swmed.edu/promals/
SPEM	http://sparks.informatics.iupui.edu/
T-Coffee, M-Coffee, 3D-Coffee	http://www.tcoffee.org/

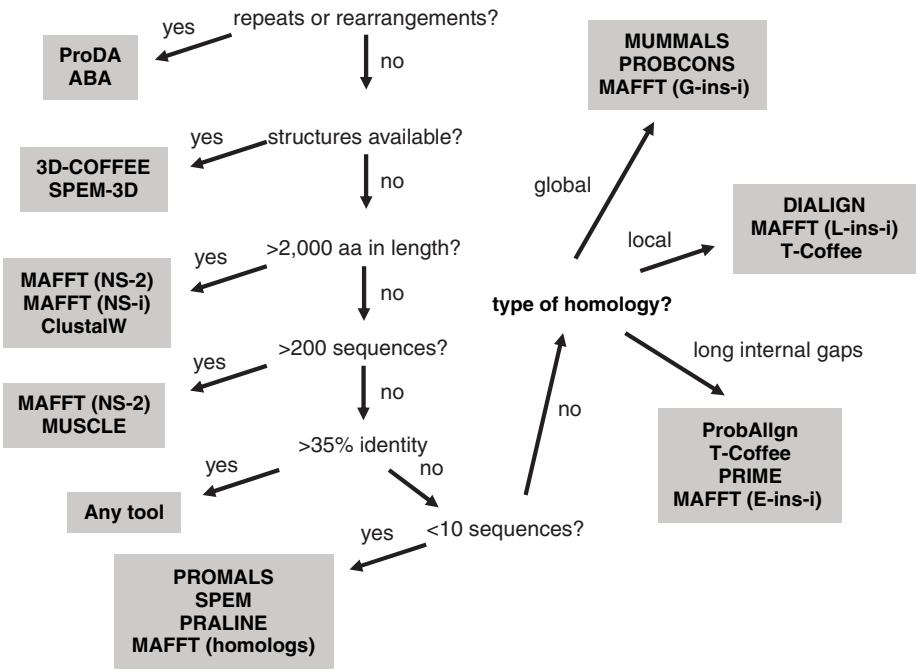

Fig. 3. Decision tree for selecting an appropriate MSA tool.

applicable tools in extreme alignment cases is generally small. We distinguish three particular situations: (1) repeated or rearranged protein domains, (2) high-throughput alignment of large numbers (>200) of input sequences, and (3) extremely long sequences (>2000 amino acids).

Currently, few programs adequately deal with alignments involving proteins with repeated or rearranged domains. While some repeat finding programs can be used for identifying repeats in protein alignments, these programs do not present a complete view of the homology in a collection of protein sequences. To date, the only programs that attempt to address this issue are ABA ***(149)*** and ProDA ***(150)***, of which we recommend the latter based on its significant advantage in accuracy on real data. While these methods are far more effective than traditional global alignment methods on sequences with repeats and rearrangements, they obtain lower accuracy on sequences where no rearrangements or repeats occur.

In high-throughput alignment scenarios, program speed can be a major bottleneck. In particular, when the number of sequences is between 200 and 1000, $O(N^2)$ distance matrix calculation (where N is the number of sequences) is generally the time-limiting factor, so progressive alignment methods with fast distance calculation, such as MAFFT (FFT-NS-2), MUSCLE (progressive), or KAlign, are recommended. For extremely large numbers of sequences (>10,000), even these fast distance calculation methods can be slow. In these cases, the PartTree ***(241)*** option in MAFFT, which relies on approximate guide tree construction in $O(N \log N)$ time based on a restricted portion of the distance matrix, is currently the only realistic option. In practice, MAFFT (PartTree), which uses approximate tree construction, achieves Q scores on average 2–3% lower than MAFFT (FFT-NS-2), which uses a full UPGMA guide tree.

For extremely long sequences (>2000 amino acids), space complexity is the main consideration in choosing an aligner. In particular, most recent multiple alignment programs tend to use dynamic programming algorithms with $O(L^2)$ memory usage (where L is the average sequence length), which is fine for most scenarios considered in benchmarking studies. For longer sequences, more efficient linear space algorithms ***(5)***, as implemented in CLUSTALW, MAFFT (FFT-NS-2), and MAFFT (FFT-NS-i), are available.

4.2. Sequences with Low Similarity

For sequences with less than 35% identity, benchmark studies under various conditions ***(221,225,242)*** have consistently identified T-Coffee, PROBCONS, and MAFFT (L-ins-i) as being the most accurate stand-alone programs currently available. More recently developed programs based on the PROBCONS framework, including MUMMALS, ProbAlign, and AMAP, have been reported

to obtain even higher accuracies. In general, however, stand-alone programs tend to perform poorly for low-identity sequences. Here, we outline two main strategies for obtaining quality alignments from the point of view of an end user: careful identification of alignment scenarios and incorporation of external information to improve alignment quality.

In general, low-identity alignments may be characterized as (1) global homology over the entire length of the protein (N-terminus to C-terminus), (2) local homology surrounded by nonhomologous flanking regions, or (3) short patches of homology interrupted by long internal gaps (*see* **Fig. 4**). Case 1 is the simplest of the three situations for which the best alignment accuracy can be expected; in these situations, MUMMALS and PROBCONS are typically the most accurate. However, when large N-terminal or C-terminal extensions exist in one or more sequences (i.e., case 2), these global methods tend to perform less well than techniques that make use of local alignment; in particular, DIALIGN, T-Coffee, and MAFFT (L-ins-i) are recommended; additionally, ProbAlign is reported to work well for these situations. Finally, the third case (case 3) occurs for highly divergent sequences in which sequence similarity remains only around functionally important residues but the order of conserved regions is identical in all sequences. Here, MAFFT (E-ins-i), T-Coffee, PRIME, and DIALIGN are recommended; these methods typically make use of more sophisticated gap penalties, such as the generalized affine gap cost ***(243,244)*** in the case of MAFFT (E-ins-i), or piecewise linear gap costs in the case of PRIME.

In general, we recommend using methods tailored for case 3 when aligning full-length proteins. Once an initial alignment is obtained, then trimming the

Fig. 4. Types of alignment homology. "-" represents a gap, "X" represents an aligned amino acid residue, and "o" is an unalignable residue. (**A**) Global homology. (**B**) Local homology. (**C**) Long internal gaps.

alignment to include only the relevant homologous parts can be done manually, and then a method designed for case 1 can be applied to give the best possible accuracy. For even more accuracy, ensemble approaches, such as the M-Coffee mode of T-Coffee or the meta_align program in MUMMALS, merge numerous independently calculated multiple sequence alignments into a single combined alignment. Clearly, ensemble aligners will not perform well if the input individual multiple alignments are poor, but in general can give modest improvements in accuracy over their component aligners.

Usually, however, the best way to improve alignment accuracy is not by more sophisticated algorithms or more careful program tuning, but rather by incorporation of external information when present. For example, the structural similarity of homologous proteins is generally conserved even after sequence similarity becomes nondetectable over the course of evolution. Therefore, sequence alignment tools that make use of structural information, such as 3D-Coffee and SPEM-3D, can achieve significantly better accuracies than tools relying solely on sequence data. Additionally, when speed is not critical and the number of input sequences is small (<10), database-aided methods can achieve better accuracy by recruiting additional homologs from a sequence database. This sort of analysis is supported by DbClustal, MAFFT-homologs, PRALINE, SPEM, and PROMALS. By enhancing site-specific evolutionary constraints, homologs can improve accuracy to a level comparable to the benefits of adding structural information.

4.3. Postprocessing and Visualization

Once an alignment has been generated, visualization tools allow manual identification of regions with reliably predicted homology; many of these tools also allow for interactive alignment editing. For alignments of sequences with low similarity, postprocessing is extremely important as most regions in a low-identity alignment will not be reliably alignable. Typically, high confidence aligned regions can be identified by looking for groups of residues with strongly conserved physicochemical properties (e.g., hydropathy, polarity, and volume), using alternative alignment objective functions for identifying reliable columns, using posterior confidences generated by alignment programs such as PROBCONS, using the consensus of several alignment methods, or even better, cross-referencing aligned positions with amino acid residues in three-dimensional protein structures. Tools for integrating structural and functional information with sequence data for alignments, such as MACSIMS ***(245)***, can also be helpful for analyzing multiple alignments. Other freely available alignment visualization and editing programs are listed in **Table 2**.

Table 2
Alignment Visualization Tools

Tool	URL
Jalview	http://www.jalview.org/
SeaView	http://pbil.univ-lyon1.fr/software/seaview.html
CINEMA	http://www.bioinf.manchester.ac.uk/dbbrowser/CINEMA2.1/
Kalignvu	http://msa.cgb.ki.se/
GeneDoc	http://www.nrbsc.org/gfx/genedoc/
STRAP	http://www.charite.de/bioinf/strap/
ClustalX	http://www.clustal.org/
BoxShade	http://www.ch.embnet.org/software/BOX_form.html
ALTAVIST	http://bibiserv.techfak.uni-bielefeld.de/altavist/

5. Conclusions

Despite its long history, research in sequence alignment continues to flourish. Each year, dozens of articles describing new methods for protein alignments are published. Although many of these approaches rely on the same basic principles, the details of the implementations can have dramatic effects on the performance, both in terms of accuracy and speed. A primary reason for this continued interest in protein sequence alignment is the centrality of comparative sequence analysis in modern computational biology: accurate alignments form the basis of many bioinformatics studies, and advances in alignment methodology can confer sweeping benefits in a wide variety of application domains.

In recent years, trends in the alignment field have included the development of efficient tools suited for high-throughput processing on a single PC (e.g., MUSCLE, MAFFT, POA, KAlign), the application of machine learning techniques for parameter estimation and sequence modeling (e.g., PROBCONS, CONTRAlign, MUMMALS), and the exploitation of publicly available sequence databases to improve accuracy of low-identity alignments (e.g., PRALINE, MAFFT, PROMALS). Furthermore, recent attempts to build alignment algorithms for dealing with proteins containing repeats and rearrangements (e.g., ABA, PRODA) push the boundaries of the types of scenarios considered by aligner developers. Finally, a number of groups have recognized the growing importance of integrating multiple alignments with other forms of data for presentation to biologists [e.g., MAO ***(246)***, MACSIMS]. While it is impossible to predict all the advances in sequence alignment research to come, their implications for practitioners is clear: the next generation of protein alignment tools will be faster, more accurate, and easier to use.

6. Notes

1. In this chapter, we focus on the problem of sequence alignment, which we distinguish from the related topic of homology search, in which we would like to identify homologs of a "query" sequence among a collection of "database" sequences. Unlike sequence alignment tools, homology search tools, such as BLASTP ***(122)*** or PSI-BLAST ***(123)***, rely extensively on approximate string matching techniques but do not focus on providing accurate residue-level alignments of the returned sequences.
2. We refer the reader to a number of other recent reviews on protein sequence alignment techniques ***(1,247–251)*** and their applications ***(252)***.
3. *Dynamic programming* (DP) refers to a class of algorithms that decomposes the solution for a complex optimization problem into overlapping solutions for smaller subproblems ***(253)***. By exploiting these overlaps, DP algorithms search an exponentially large space (e.g., the space of all possible alignments) by solving a small polynomial number of subproblems.
4. The SAGA algorithm, for example, was found to be 100–1000× slower than CLUSTALW in a number of typical multiple alignments ***(29)***.
5. As previously pointed out ***(62)***, although the progressive alignment procedure may be linear in the number of sequences N, typical algorithms for tree construction require $O(N^3)$ time. For large numbers of sequences (e.g., 10,000), this is intractable. An approximate $O(N^2)$ UPGMA tree construction algorithm that produces reasonable trees in practice has been described; alternatively, exact worst-case quadratic time algorithms for UPGMA ***(254)*** and neighbor-joining ***(255)*** tree construction exist. For situations with very large N, the recent PartTree algorithm ***(241)*** computes approximate trees in $O(N \log N)$ time.
6. *Parametric alignment* ***(256–258)*** is an attempt to abandon the need for parameter estimation altogether by computing optimal sequence alignments for all possible parameter sets. However, the resulting algorithms are often computationally expensive, and for most biologists, the generated alignment sets are of limited benefit when alignment quality is difficult to judge manually.

Acknowledgments

We thank Karen Ann Lee for help in preparing the manuscript. C.B.D was funded by an NDSEG fellowship.

References

1. Notredame, C. (2002) Recent progress in multiple sequence alignment: a survey. *Pharmacogenomics* **3,** 131–144.
2. Needleman, S. B. and Wunsch, C. D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol*. **48,** 443–453.

3. Smith, T. F. and Waterman, M. S. (1981) Identification of common molecular subsequences. *J. Mol. Biol*. **147,** 195–197.
4. Gotoh, O. (1982) An improved algorithm for matching biological sequences. *J. Mol. Biol*. **162,** 705–708.
5. Myers, E. W. and Miller, W. (1988) Optimal alignments in linear space. *Comput. Appl. Biosci*. **4,** 11–17.
6. Murata, M., Richardson, J. S., and Sussman, J. L. (1985) Simultaneous comparison of three protein sequences. *Proc. Natl. Acad. Sci. USA* **82,** 3073–3077.
7. Waterman, M. S. and Jones, R. (1990) Consensus methods for DNA and protein sequence alignment. *Methods Enzymol*. **183,** 221–237.
8. Durbin, R., Eddy, S. R., Krogh, A., and Mitchison, G. (1999) *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge.
9. Gonnet, G. H., Korostensky, C., and Benner, S. (2000) Evaluation measures of multiple sequence alignments. *J. Comput. Biol*. **7,** 261–276.
10. Wang, L. and Jiang, T. (1994) On the complexity of multiple sequence alignment. *J. Comput. Biol*. **1,** 337–348.
11. Bonizzoni, P. and Della Vedova, G. (2001) The complexity of multiple sequence alignment with SP-score that is a metric. *Theor. Comput. Sci*. **259,** 63–79.
12. Just, W. (2001) Computational complexity of multiple sequence alignment with SP-score. *J. Comput. Biol*. **8,** 615–623.
13. Elias, I. (2006) Settling the intractability of multiple alignment. *J. Comput. Biol*. **13,** 1323–1339.
14. Lipman, D. J., Altschul, S. F., and Kececioglu, J. D. (1989) A tool for multiple sequence alignment. *Proc. Natl. Acad. Sci. USA* **86,** 4412–4415.
15. Gupta, S. K., Kececioglu, J. D., and Schaffer, A. A. (1995) Improving the practical space and time efficiency of the shortest-paths approach to sum-of-pairs multiple sequence alignment. *J. Comput. Biol*. **2,** 459–472.
16. Carrillo, H. and Lipman, D. (1988) The multiple sequence alignment problem in biology. *SIAM J. Appl. Math*. **48,** 1073–1082.
17. Dress, A., Fullen, G., and Perrey, S. (1995) A divide and conquer approach to multiple alignment. *Proc. Int. Conf. Intell. Syst. Mol. Biol*. **3,** 107–113.
18. Stoye, J., Perrey, S. W., and Dress, A. W. M. (1997) Improving the divide-and-conquer approach to sum-of-pairs multiple sequence alignment. *Appl. Math. Lett*. **10,** 67–73.
19. Stoye, J., Moulton, V., and Dress, A. W. (1997) DCA: an efficient implementation of the divide-and-conquer approach to simultaneous multiple sequence alignment. *Comput. Appl. Biosci*. **13,** 625–626.
20. Stoye, J. (1998) Multiple sequence alignment with the divide-and-conquer method. *Gene* **211,** GC45–56.
21. Reinert, K., Stoye, J., and Will, T. (2000) An iterative method for faster sum-of-pairs multiple sequence alignment. *Bioinformatics* **16,** 808–814.
22. Holland, J. H. (1975) *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor.

23. Zhang, C. and Wong, A. K. (1997) A genetic algorithm for multiple molecular sequence alignment. *Comput. Appl. Biosci*. **13,** 565–581.
24. Anbarasu, L. A., Narayanasamy, P., and Sundararajan, V. (1998) Multiple sequence alignment using parallel genetic algorithms. SEAL.
25. Chellapilla, K. and Fogel, G. B. (1999) Multiple sequence alignment using evolutionary programming. Congress on Evolutionary Computation.
26. Gonzalez, R. R., Izquierdo, C. M., and Seijas, J. (1999) Multiple protein sequence comparison by genetic algorithms. SPIE-98.
27. Cai, L., Juedes, D., and Liakhovitch, E. (2000) Evolutionary computation techniques for multiple sequence alignment. Congress on Evolutionary Computation.
28. Zhang, G.-Z. and Huang, D.-S. (2004) Aligning multiple protein sequence by an improved genetic algorithm. IEEE International Joint Conference on Neural Networks.
29. Notredame, C. and Higgins, D. G. (1996) SAGA: sequence alignment by genetic algorithm. *Nucleic Acids Res*. **24,** 1515–1524.
30. Isokawa, M., Takahashi, K., and Shimizu, T. (1996) Multiple sequence alignment using a genetic algorithm. *Genome Inform*. **7,** 176–177.
31. Harada, Y., Wayama, M., and Shimizu, T. (1997) An inspection of the multiple alignment methods with use of genetic algorithm. *Genome Inform*. **8,** 272–273.
32. Hanada, K., Yokoyama, T., and Shimizu, T. (2000) Multiple sequence alignment by genetic algorithm. *Genome Inform*. **11,** 317–318.
33. Yokoyama, T., Watanabe, T., Taneda, A., and Shimizu, T. (2001) A web server for multiple sequence alignment using genetic algorithm. *Genome Inform*. **12,** 382–383.
34. Nguyen, H. D., Yoshihara, I., Yamamori, K., and Yasunaga, M. (2002) A parallel hybrid genetic algorithm for multiple protein sequence alignment. *Evol. Comput*. **1,** 309–314.
35. Kirkpatrick, S., Gelatt, J., C. D., and Vecchi, M. P. (1983) Optimization by simulated annealing. *Science* 220, 671–680.
36. Ishikawa, M., Toya, T., Hoshida, M., Nitta, K., Ogiwara, A., and Kanehisa, M. (1993) Multiple sequence alignment by parallel simulated annealing. *Comput. Appl. Biosci*. **9,** 267–273.
37. Kim, J., Pramanik, S., and Chung, M. J. (1994) Multiple sequence alignment using simulated annealing. *Comput. Appl. Biosci*. **10,** 419–426.
38. Eddy, S. R. (1995) Multiple alignment using hidden Markov models. *Proc. Int. Conf. Intell. Syst. Mol. Biol*. **3,** 114–120.
39. Ikeda, T. and Imai, H. (1999) Enhanced A* algorithms for multiple alignments: optimal alignments for several sequences and k-opt approximate alignments for large cases. *Theor. Comput. Sci*. **210,** 341–374.
40. Horton, P. (2001) Tsukuba BB: a branch and bound algorithm for local multiple alignment of DNA and protein sequences. *J. Comput. Biol*. **8,** 283–303.
41. Reinert, K., Lenhof, H.-P., Mutzel, P., Mehlhorn, K., and Kececioglu, J. D. (1997) A branch-and-cut algorithm for multiple sequence alignment. RECOMB.

42. Reinert, K., Stoye, J., and Will, T. (1999) Combining divide-and-conquer, the A*-algorithm and successive realignment approaches to speed up multiple sequence alignment. German Conference on Bioinformatics.
43. Lermen, M. and Reinert, K. (2000) The practical use of the A* algorithm for exact multiple sequence alignment. *J. Comput. Biol*. **7,** 655–671.
44. Feng, D. F. and Doolittle, R. F. (1987) Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol*. **25,** 351–360.
45. Taylor, W. R. (1987) Multiple sequence alignment by a pairwise algorithm. *Comput. Appl. Biosci*. **3,** 81–87.
46. Taylor, W. R. (1988) A flexible method to align large numbers of biological sequences. *J. Mol. Evol*. **28,** 161–169.
47. Kececioglu, J. and Starrett, D. (2004) Aligning alignments exactly. RECOMB.
48. Kececioglu, J. and Zhang, W. (1998) Aligning alignments. CPM.
49. Altschul, S. F. (1989) Gap costs for multiple sequence alignment. *J. Theor. Biol*. **138,** 297–309.
50. Katoh, K., Misawa, K., Kuma, K., and Miyata, T. (2002) MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Res*. **30,** 3059–3066.
51. Edgar, R. C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res*. **32,** 1792–1797.
52. Huang, X. (1994) On global sequence alignment. *Comput. Appl. Biosci*. **10,** 227–235.
53. Pei, J., Sadreyev, R., and Grishin, N. V. (2003) PCMA: fast and accurate multiple sequence alignment based on profile consistency. *Bioinformatics* **19,** 427–428.
54. Smith, R. F. and Smith, T. F. (1992) Pattern-induced multi-sequence alignment (PIMA) algorithm employing secondary structure-dependent gap penalties for use in comparative protein modelling. *Protein Eng*. 5, 35–41.
55. Yamada, S., Gotoh, O., and Yamana, H. (2006) Improvement in accuracy of multiple sequence alignment using novel group-to-group sequence alignment algorithm with piecewise linear gap cost. *BMC Bioinform.* 7, 524.
56. Gotoh, O. (1996) Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments. *J. Mol. Biol*. **264,** 823–838.
57. Corpet, F. (1988) Multiple sequence alignment with hierarchical clustering. *Nucleic Acids Res*. **16,** 10881–10890.
58. Higgins, D. G. and Sharp, P. M. (1988) CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. *Gene* **73,** 237–244.
59. Higgins, D. G. and Sharp, P. M. (1989) Fast and sensitive multiple sequence alignments on a microcomputer. *Comput. Appl. Biosci*. **5,** 151–153.
60. Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994) 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,** 4673–4680.

61. Katoh, K., Kuma, K., Toh, H., and Miyata, T. (2005) MAFFT version 5: improvement in accuracy of multiple sequence alignment. *Nucleic Acids Res*. **33,** 511–518.
62. Edgar, R. C. (2004) MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinform.* **5,** 113.
63. Notredame, C., Holm, L., and Higgins, D. G. (1998) COFFEE: an objective function for multiple sequence alignments. *Bioinformatics* **14,** 407–422.
64. Notredame, C., Higgins, D. G., and Heringa, J. (2000) T-Coffee: A novel method for fast and accurate multiple sequence alignment. *J. Mol. Biol*. **302,** 205–217.
65. Lassmann, T. and Sonnhammer, E. L. (2005) Kalign–an accurate and fast multiple sequence alignment algorithm. *BMC Bioinform*. **6,** 298.
66. Lee, C., Grasso, C., and Sharlow, M. F. (2002) Multiple sequence alignment using partial order graphs. *Bioinformatics* **18,** 452–464.
67. Lee, C. (2003) Generating consensus sequences from partial order multiple sequence alignment graphs. *Bioinformatics* **19,** 999–1008.
68. Grasso, C. and Lee, C. (2004) Combining partial order alignment and progressive multiple sequence alignment increases alignment speed and scalability to very large alignment problems. *Bioinformatics* **20,** 1546–1556.
69. Do, C. B., Mahabhashyam, M. S., Brudno, M., and Batzoglou, S. (2005) ProbCons: Probabilistic consistency-based multiple sequence alignment. *Genome Res*. **15,** 330–340.
70. Pei, J. and Grishin, N. V. (2006) MUMMALS: multiple sequence alignment improved by using hidden Markov models with local structural information. *Nucleic Acids Res*. **34,** 4364–4374.
71. Pei, J. and Grishin, N. V. (2007) PROMALS: towards accurate multiple sequence alignments of distantly related proteins. *Bioinformatics* **23,** 802–808.
72. Gribskov, M., McLachlan, A. D., and Eisenberg, D. (1987) Profile analysis: detection of distantly related proteins. *Proc. Natl. Acad. Sci. US A* **84,** 4355–4358.
73. von Ohsen, N., Sommer, I., and Zimmer, R. (2003) Profile-profile alignment: a powerful tool for protein structure prediction. *Pac. Symp. Biocomput*. 252–263.
74. von Ohsen, N., Sommer, I., Zimmer, R., and Lengauer, T. (2004) Arby: automatic protein structure prediction using profile-profile alignment and confidence measures. *Bioinformatics* **20,** 2228–2235.
75. Soding, J. (2005) Protein homology detection by HMM-HMM comparison. *Bioinformatics* **21,** 951–960.
76. von Ohsen, N. and Zimmer, R. (2001) Improving profile-profile alignments via log-average scoring. WABI.
77. Yona, G. and Levitt, M. (2002) Within the twilight zone: a sensitive profile-profile comparison tool based on information theory. *J. Mol. Biol*. **315,** 1257–1275.
78. Heger, A. and Holm, L. (2003) Exhaustive enumeration of protein domain families. *J. Mol. Biol*. **328,** 749–767.
79. Mittelman, D., Sadreyev, R., and Grishin, N. (2003) Probabilistic scoring measures for profile-profile comparison yield more accurate short seed alignments. *Bioinformatics* **19,** 1531–1539.

80. Sadreyev, R. and Grishin, N. (2003) COMPASS: a tool for comparison of multiple protein alignments with assessment of statistical significance. *J. Mol. Biol*. **326,** 317–336.
81. Edgar, R. C. and Sjolander, K. (2004) COACH: profile-profile alignment of protein families using hidden Markov models. *Bioinformatics* **20,** 1309–1318.
82. Rychlewski, L., Jaroszewski, L., Li, W., and Godzik, A. (2000) Comparison of sequence profiles. Strategies for structural predictions using sequence information. *Protein Sci*. **9,** 232–241.
83. Edgar, R. C. and Sjolander, K. (2004) A comparison of scoring functions for protein sequence profile alignment. *Bioinformatics* **20,** 1301–1308.
84. Ohlson, T., Wallner, B., and Elofsson, A. (2004) Profile-profile methods provide improved fold-recognition: a study of different profile–profile alignment methods. *Proteins* **57,** 188–197.
85. Sokal, R. R. and Michener, C. D. (1958) A statistical method for evaluating systematic relationships. *Univ. Kans. Sci. Bull*. **28,** 1409–1438.
86. Sneath, P. H. and Sokal, R. R. (1962) Numerical taxonomy. *Nature* **193,** 855–860.
87. Saitou, N. and Nei, M. (1987) The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol*. **4,** 406–425.
88. Studier, J. A. and Keppler, K. J. (1988) A note on the neighbor-joining algorithm of Saitou and Nei. *Mol. Biol. Evol*. **5,** 729–731.
89. Jones, D. T., Taylor, W. R., and Thornton, J. M. (1992) The rapid generation of mutation data matrices from protein sequences. *Comput. Appl. Biosci*. **8,** 275–282.
90. Edgar, R. C. (2004) Local homology recognition and distance measures in linear time using compressed amino acid alphabets. *Nucleic Acids Res*. **32,** 380–385.
91. Wu, S. and Manber, U. (1992) Fast text searching allowing errors. *Commun. ACM* **35,** 83–91.
92. Vingron, M. and Argos, P. (1989) A fast and sensitive multiple sequence alignment algorithm. *Comput. Appl. Biosci*. **5,** 115–121.
93. Vingron, M. and Argos, P. (1990) Determination of reliable regions in protein sequence alignments. *Protein Eng*. **3,** 565–569.
94. Vingron, M. and Argos, P. (1991) Motif recognition and alignment for many sequences by comparison of dot-matrices. *J. Mol. Biol*. **218,** 33–43.
95. Gotoh, O. (1990) Consistency of optimal sequence alignments. *Bull. Math. Biol*. **52,** 509–525.
96. Van Walle, I., Lasters, I., and Wyns, L. (2003) Consistency matrices: quantified structure alignments for sets of related proteins. *Proteins* **51,** 1–9.
97. Van Walle, I., Lasters, I., and Wyns, L. (2004) Align-m–a new algorithm for multiple alignment of highly divergent sequences. *Bioinformatics* **20,** 1428–1435.
98. Do, C. B., Gross, S. S., and Batzoglou, S. (2006) CONTRAlign: discriminative training for protein sequence alignment. RECOMB.
99. Lolkema, J. S. and Slotboom, D. J. (1998) Hydropathy profile alignment: a tool to search for structural homologues of membrane proteins. *FEMS Microbiol. Rev*. **22,** 305–322.

100. Altschul, S. F., Carroll, R. J., and Lipman, D. J. (1989) Weights for data related by a tree. *J. Mol. Biol*. **207,** 647–653.
101. Vingron, M. and Sibbald, P. R. (1993) Weighting in sequence space: a comparison of methods in terms of generalized sequences. *Proc. Natl. Acad. Sci. USA* **90,** 8777–8781.
102. Sibbald, P. R. and Argos, P. (1990) Weighting aligned protein or nucleic acid sequences to correct for unequal representation. *J. Mol. Biol*. **216,** 813–818.
103. Henikoff, S. and Henikoff, J. G. (1994) Position-based sequence weights. *J. Mol. Biol*. **243,** 574–578.
104. Eddy, S. R., Mitchison, G., and Durbin, R. (1995) Maximum discrimination hidden Markov models of sequence consensus. *J. Comput. Biol*. **2,** 9–23.
105. Gotoh, O. (1995) A weighting system and algorithm for aligning many phylogenetically related sequences. *Comput. Appl. Biosci*. **11,** 543–551.
106. Krogh, A. and Mitchison, G. (1995) Maximum entropy weighting of aligned sequences of proteins or DNA. *Proc. Int. Conf. Intell. Syst. Mol. Biol*. **3,** 215–221.
107. Karchin, R. and Hughey, R. (1998) Weighting hidden Markov models for maximum discrimination. *Bioinformatics* **14,** 772–782.
108. May, A. C. (2001) Optimal classification of protein sequences and selection of representative sets from multiple alignments: application to homologous families and lessons for structural genomics. *Protein Eng*. **14,** 209–217.
109. Hirosawa, M., Totoki, Y., Hoshida, M., and Ishikawa, M. (1995) Comprehensive study on iterative algorithms of multiple sequence alignment. *Comput. Appl. Biosci*. **11,** 13–18.
110. Wang, Y. and Li, K. B. (2004) An adaptive and iterative algorithm for refining multiple sequence alignment. *Comput. Biol. Chem*. **28,** 141–148.
111. Wallace, I. M., O'Sullivan, O., and Higgins, D. G. (2005) Evaluation of iterative alignment algorithms for multiple alignment. *Bioinformatics* **21,** 1408–1414.
112. Brocchieri, L. and Karlin, S. (1998) A symmetric-iterated multiple alignment of protein sequences. *J. Mol. Biol*. **276,** 249–264.
113. Subbiah, S. and Harrison, S. C. (1989) A method for multiple sequence alignment with gaps. *J. Mol. Biol*. **209,** 539–548.
114. Barton, G. J. and Sternberg, M. J. (1987) A strategy for the rapid multiple alignment of protein sequences. Confidence levels from tertiary structure comparisons. *J. Mol. Biol*. **198,** 327–337.
115. Barton, G. J. and Sternberg, M. J. (1987) Evaluation and improvements in the automatic alignment of protein sequences. *Protein Eng*. **1,** 89–94.
116. Bains, W. (1986) MULTAN: a program to align multiple DNA sequences. *Nucleic Acids Res*. 14, 159–177.
117. Thompson, J. D., Thierry, J. C., and Poch, O. (2003) RASCAL: rapid scanning and correction of multiple sequence alignments. *Bioinformatics* **19,** 1155–1161.
118. Chakrabarti, S., Lanczycki, C. J., Panchenko, A. R., Przytycka, T. M., Thiessen, P. A., and Bryant, S. H. (2006) State of the art: refinement of multiple sequence alignments. *BMC Bioinform*. **7,** 499.

119. Chakrabarti, S., Lanczycki, C. J., Panchenko, A. R., Przytycka, T. M., Thiessen, P. A., and Bryant, S. H. (2006) Refining multiple sequence alignments with conserved core regions. *Nucleic Acids Res*. **34,** 2598–2606.
120. Huang, X. Q., Hardison, R. C., and Miller, W. (1990) A space-efficient algorithm for local similarities. *Comput. Appl. Biosci*. **6,** 373–381.
121. Huang, X. and Miller, W. (1991) A time-efficient, linear-space local similarity algorithm. *Adv. Appl. Math*. **12,** 337–357.
122. Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990) Basic local alignment search tool. *J. Mol. Biol*. **215,** 403–410.
123. Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., et al. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*. **25,** 3389–3402.
124. Pearson, W. R. (1998) Empirical statistical estimates for sequence similarity searches. *J. Mol. Biol*. **276,** 71–84.
125. Pearson, W. R. (1990) Rapid and sensitive sequence comparison with FASTP and FASTA. *Methods Enzymol*. **183,** 63–98.
126. Pearson, W. R. (2000) Flexible sequence similarity searching with the FASTA3 program package. *Methods Mol. Biol*. **132,** 185–219.
127. Morgenstern, B., Dress, A., and Werner, T. (1996) Multiple DNA and protein sequence alignment based on segment-to-segment comparison. *Proc. Natl. Acad. Sci. USA* **93,** 12098–12103.
128. Morgenstern, B., Frech, K., Dress, A., and Werner, T. (1998) DIALIGN: finding local similarities by multiple sequence alignment. *Bioinformatics* **14,** 290–294.
129. Morgenstern, B. (1999) DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment. *Bioinformatics* **15,** 211–218.
130. Morgenstern, B. (2004) DIALIGN: multiple DNA and protein sequence alignment at BiBiServ. *Nucleic Acids Res*. **32,** W33–36.
131. Subramanian, A. R., Weyer-Menkhoff, J., Kaufmann, M., and Morgenstern, B. (2005) DIALIGN-T: an improved algorithm for segment-based multiple sequence alignment. *BMC Bioinform*. **6,** 66.
132. Depiereux, E. and Feytmans, E. (1992) MATCH-BOX: a fundamentally new algorithm for the simultaneous alignment of several protein sequences. *Comput. Appl. Biosci*. **8,** 501–509.
133. Depiereux, E., Baudoux, G., Briffeuil, P., Reginster, I., De Bolle, X., Vinals, C., et al. (1997) Match-Box_server: a multiple sequence alignment tool placing emphasis on reliability. *Comput. Appl. Biosci*. **13,** 249–256.
134. Schwartz, A. S. and Pachter, L. (2007) Multiple alignment by sequence annealing. *Bioinformatics* **23,** e24–29.
135. Pellegrini, M., Marcotte, E. M., and Yeates, T. O. (1999) A fast algorithm for genome-wide analysis of proteins with repeated sequences. *Proteins* **35,** 440–446.
136. Notredame, C. (2001) Mocca: semi-automatic method for domain hunting. *Bioinformatics* **17,** 373–374.
137. Heger, A. and Holm, L. (2000) Rapid automatic detection and alignment of repeats in protein sequences. *Proteins* **41,** 224–237.

138. Heringa, J. and Argos, P. (1993) A method to recognize distant repeats in protein sequences. *Proteins* **17,** 391–341.
139. Szklarczyk, R. and Heringa, J. (2004) Tracking repeats using significance and transitivity. *Bioinformatics* **20(Suppl 1)**, I311–I317.
140. Sammeth, M. and Heringa, J. (2006) Global multiple-sequence alignment with repeats. *Proteins* **64,** 263–274.
141. Lawrence, C. E., Altschul, S. F., Boguski, M. S., Liu, J. S., Neuwald, A. F., and Wootton, J. C. (1993) Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science* **262,** 208–214.
142. Neuwald, A. F., Liu, J. S., and Lawrence, C. E. (1995) Gibbs motif sampling: detection of bacterial outer membrane protein repeats. *Protein Sci*. **4,** 1618–1632.
143. Henikoff, S., Henikoff, J. G., Alford, W. J., and Pietrokovski, S. (1995) Automated construction and graphical presentation of protein blocks from unaligned sequences. *Gene* **163,** GC17–26.
144. Smith, H. O., Annau, T. M., and Chandrasegaran, S. (1990) Finding sequence motifs in groups of functionally related proteins. *Proc. Natl. Acad. Sci. USA* **87,** 826–830.
145. Bailey, T. L. and Elkan, C. (1994) Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc. Int. Conf. Intell. Syst. Mol. Biol*. **2,** 28–36.
146. Sonnhammer, E. L. and Kahn, D. (1994) Modular arrangement of proteins as inferred from analysis of homology. *Protein Sci*. **3,** 482–492.
147. Schuler, G. D., Altschul, S. F., and Lipman, D. J. (1991) A workbench for multiple alignment construction and analysis. *Proteins* **9,** 180–190.
148. Pevzner, P. A., Tang, H., and Tesler, G. (2004) De novo repeat classification and fragment assembly. *Genome Res*. **14,** 1786–1796.
149. Raphael, B., Zhi, D., Tang, H., and Pevzner, P. (2004) A novel method for multiple alignment of sequences with repeated and shuffled elements. *Genome Res*. **14,** 2336–2346.
150. Phuong, T. M., Do, C. B., Edgar, R. C., and Batzoglou, S. (2006) Multiple alignment of protein sequences with repeats and rearrangements. *Nucleic Acids Res*. **34,** 5932–5942.
151. Bishop, M. J. and Thompson, E. A. (1986) Maximum likelihood alignment of DNA sequences. *J. Mol. Biol*. **190,** 159–165.
152. Hein, J., Wiuf, C., Knudsen, B., Moller, M. B., and Wibling, G. (2000) Statistical alignment: computational properties, homology testing and goodness-of-fit. *J. Mol. Biol*. **302,** 265–279.
153. Thorne, J. L., Kishino, H., and Felsenstein, J. (1991) An evolutionary model for maximum likelihood alignment of DNA sequences. *J. Mol. Evol*. **33,** 114–124.
154. Thorne, J. L., Kishino, H., and Felsenstein, J. (1992) Inching toward reality: an improved likelihood model of sequence evolution. *J. Mol. Evol*. **34,** 3–16.
155. Miklos, I. and Toroczkai, Z. (2001) An improved model for statistical alignment. WABI.

156. Miklos, I. (2003) Algorithm for statistical alignment of sequences derived from a Poisson sequence length distribution. *Disc. Appl. Math*. **127,** 79–84.
157. Miklos, I., Lunter, G. A., and Holmes, I. (2004) A "Long Indel" model for evolutionary sequence alignment. *Mol. Biol. Evol*. **21,** 529–540.
158. Knudsen, B. and Miyamoto, M. M. (2003) Sequence alignments and pair hidden Markov models using evolutionary history. *J. Mol. Biol*. **333,** 453–460.
159. Metzler, D. (2003) Statistical alignment based on fragment insertion and deletion models. *Bioinformatics* **19,** 490–499.
160. Hein, J. (2001) A generalisation of the Thorne-Kishino-Felsenstein model of statistical alignment to k sequences related by a binary tree. PSB.
161. Hein, J., Jensen, J. L., and Pedersen, C. N. (2003) Recursions for statistical multiple alignment. *Proc. Natl. Acad. Sci. USA* **100,** 14960–14965.
162. Holmes, I. and Bruno, W. J. (2001) Evolutionary HMMs: a Bayesian approach to multiple alignment. *Bioinformatics* **17,** 803–820.
163. Holmes, I. (2003) Using guide trees to construct multiple-sequence evolutionary HMMs. *Bioinformatics* **19(Suppl 1),** i147–157.
164. Steel, M. and Hein, J. (2001) Applying the Thorne-Kishino-Felsenstein model to sequence evolution on a star-shaped tree. *Appl. Math. Lett*. **14,** 679–684.
165. Miklos, I. (2002) An improved algorithm for statistical alignment of sequences related by a star tree. *Bull. Math. Biol*. **64,** 771–779.
166. Lunter, G. A., Miklos, I., Song, Y. S., and Hein, J. (2003) An efficient algorithm for statistical multiple alignment on arbitrary phylogenetic trees. *J. Comput. Biol*. **10,** 869–889.
167. Jensen, J. L. and Hein, J. (2005) Gibbs sampler for statistical multiple alignment. *Stat. Sin*. **15,** 889–907.
168. Hein, J. (1990) Unified approach to alignment and phylogenies. *Methods Enzymol*. **183,** 626–645.
169. Vingron, M. and von Haeseler, A. (1997) Towards integration of multiple alignment and phylogenetic tree construction. *J. Comput. Biol*. **4,** 23–34.
170. Fleissner, R., Metzler, D., and von Haeseler, A. (2005) Simultaneous statistical multiple alignment and phylogeny reconstruction. *Syst. Biol*. **54,** 548–561.
171. Lunter, G., Miklos, I., Drummond, A., Jensen, J. L., and Hein, J. (2005) Bayesian coestimation of phylogeny and sequence alignment. *BMC Bioinform*. **6,** 83.
172. Redelings, B. D. and Suchard, M. A. (2005) Joint Bayesian estimation of alignment and phylogeny. *Syst. Biol*. **54,** 401–418.
173. Metzler, D., Fleissner, R., Wakolbinger, A., and von Haeseler, A. (2001) Assessing variability by joint sampling of alignments and mutation rates. *J. Mol. Evol*. **53,** 660–669.
174. Allison, L. and Wallace, C. S. (1994) The posterior probability distribution of alignments and its application to parameter estimation of evolutionary trees and to optimization of multiple alignments. *J. Mol. Evol*. **39,** 418–430.
175. Krogh, A., Brown, M., Mian, I. S., Sjolander, K., and Haussler, D. (1994) Hidden Markov models in computational biology. Applications to protein modeling. *J. Mol. Biol*. **235,** 1501–1531.

176. Krogh, A. (1998) An introduction to hidden Markov models for biological sequences. In *Computational Methods in Molecular Biology* (Salzberg, S., Searls, D., Kasif, S., eds.). Elsevier Science, St. Louis, MO, pp. 45–63.
177. Hughey, R. and Krogh, A. (1996) Hidden Markov models for sequence analysis: extension and analysis of the basic method. *Comput. Appl. Biosci*. **12,** 95–107.
178. Eddy, S. R. (1996) Hidden Markov models. *Curr. Opin. Struct. Biol*. **6,** 361–365.
179. Eddy, S. R. (1998) Profile hidden Markov models. *Bioinformatics* **14,** 755–763.
180. Mamitsuka, H. (2005) Finding the biologically optimal alignment of multiple sequences. *Artif. Intell. Med*. **35,** 9–18.
181. Baldi, P. and Chauvin, Y. (1994) Smooth on-line learning algorithms for hidden Markov models. *Neural Comput*. **6,** 307–318.
182. Baldi, P., Chauvin, Y., Hunkapiller, T., and McClure, M. A. (1994) Hidden Markov models of biological primary sequence information. *Proc. Natl. Acad. Sci. USA* **91,** 1059–1063.
183. Viterbi, A. J. (1967) Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inform. Theory* **It13,** 260.
184. Grundy, W. N., Bailey, T. L., Elkan, C. P., and Baker, M. E. (1997) Meta-MEME: motif-based hidden Markov models of protein families. *Comput. Appl. Biosci*. **13,** 397–406.
185. Bucher, P., Karplus, K., Moeri, N., and Hofmann, K. (1996) A flexible motif search technique based on generalized profiles. *Comput. Chem*. **20,** 3–23.
186. Karplus, K., Barrett, C., and Hughey, R. (1998) Hidden Markov models for detecting remote protein homologies. *Bioinformatics* **14,** 846–856.
187. Park, J., Karplus, K., Barrett, C., Hughey, R., Haussler, D., Hubbard, T., *et al*. (1998) Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. *J. Mol. Biol*. **284,** 1201–1210.
188. Sonnhammer, E. L., Eddy, S. R., Birney, E., Bateman, A., and Durbin, R. (1998) Pfam: multiple sequence alignments and HMM-profiles of protein domains. *Nucleic Acids Res*. **26,** 320–322.
189. Eddy, S. R. HMMER: a profile hidden Markov modeling package, available from http://hmmer.janelia.org/.
190. Sjolander, K., Karplus, K., Brown, M., Hughey, R., Krogh, A., Mian, I. S., *et al.* (1996) Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology. *Comput. Appl. Biosci*. **12,** 327–345.
191. Barrett, C., Hughey, R., and Karplus, K. (1997) Scoring hidden Markov models. *Comput. Appl. Biosci*. **13,** 191–199.
192. McClure, M. A., Smith, C., and Elton, P. (1996) Parameterization studies for the SAM and HMMER methods of hidden Markov model generation. *Proc. Int. Conf. Intell. Syst. Mol. Biol*. **4,** 155–164.
193. Karplus, K. and Hu, B. (2001) Evaluation of protein multiple alignments by SAM-T99 using the BAliBASE multiple alignment test set. *Bioinformatics* **17,** 713–720.
194. Loytynoja, A. and Milinkovitch, M. C. (2003) A hidden Markov model for progressive multiple alignment. *Bioinformatics* **19,** 1505–1513.

195. Edgar, R. C. and Sjolander, K. (2003) Simultaneous sequence alignment and tree construction using hidden Markov models. *Pac. Symp. Biocomput*. 180–191.
196. Edgar, R. C. and Sjolander, K. (2003) SATCHMO: sequence alignment and tree construction using hidden Markov models. *Bioinformatics* **19,** 1404–1411.
197. Loytynoja, A. and Goldman, N. (2005) An algorithm for progressive multiple alignment of sequences with insertions. *Proc. Natl. Acad. Sci. USA* **102,** 10557–10562.
198. Holmes, I. and Durbin, R. (1998) Dynamic programming alignment accuracy. *J. Comput. Biol*. **5,** 493–504.
199. Schwartz, A. S., Myers, E., and Pachter, L. (2006) Alignment metric accuracy. *arXiv 2006:q-bio.QM/0510052*.
200. Roshan, U. and Livesay, D. R. (2006) Probalign: multiple sequence alignment using partition function posterior probabilities. *Bioinformatics* **22,** 2715–2721.
201. Wallace, I. M., O'Sullivan, O., Higgins, D. G., and Notredame, C. (2006) M-Coffee: combining multiple sequence alignment methods with T-Coffee. *Nucleic Acids Res*. **34,** 1692–1699.
202. Kececioglu, J. D. (1993) The maximum weight trace problem in multiple sequence alignment. CPM.
203. Kececioglu, J. D., Lenhof, H.-P., Mehlhorn, K., Mutzel, P., Reinert, K., and Vingron, M. (2000) A polyhedral approach to sequence alignment problems. *Disc. Appl. Math*. **104,** 143–186.
204. Koller, G. and Raidl, G. R. (2004) An evolutionary algorithm for the maximum weight trace formulation of the multiple sequence alignment problem. In *LNCS*, 3242, pp. 302–311.
205. Simossis, V. A. and Heringa, J. (2005) PRALINE: a multiple sequence alignment toolbox that integrates homology-extended and secondary structure information. *Nucleic Acids Res*. **33,** W289–294.
206. Simossis, V. A., Kleinjung, J., and Heringa, J. (2005) Homology-extended sequence alignment. *Nucleic Acids Res*. **33,** 816–824.
207. Thompson, J. D., Plewniak, F., Thierry, J., and Poch, O. (2000) DbClustal: rapid and reliable global multiple alignments of protein sequences detected by database searches. *Nucleic Acids Res*. **28,** 2919–2926.
208. Wang, J. and Feng, J. A. (2005) NdPASA: a novel pairwise protein sequence alignment algorithm that incorporates neighbor-dependent amino acid propensities. *Proteins* **58,** 628–637.
209. Yang, A. S. (2002) Structure-dependent sequence alignment for remotely related proteins. *Bioinformatics* **18,** 1658–1665.
210. Zhou, H. and Zhou, Y. (2005) SPEM: improving multiple sequence alignment with sequence profiles and predicted secondary structures. *Bioinformatics* **21,** 3615–3621.
211. O'Sullivan, O., Suhre, K., Abergel, C., Higgins, D. G., and Notredame, C. (2004) 3DCoffee: combining protein sequences and structures within multiple sequence alignments. *J. Mol. Biol*. **340,** 385–395.

212. Armougom, F., Moretti, S., Poirot, O., Audic, S., Dumas, P., Schaeli, B., *et al*. (2006) Expresso: automatic incorporation of structural information in multiple sequence alignments using 3D-Coffee. *Nucleic Acids Res*. **34,** W604–608.
213. Thompson, J. D., Plewniak, F., and Poch, O. (1999) BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics* **15,** 87–88.
214. Thompson, J. D., Plewniak, F., and Poch, O. (1999) A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Res*. **27,** 2682–2690.
215. Mizuguchi, K., Deane, C. M., Blundell, T. L., and Overington, J. P. (1998) HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Sci*. **7,** 2469–2471.
216. Van Walle, I., Lasters, I., and Wyns, L. (2005) SABmark–a benchmark for sequence alignment that covers the entire known fold space. *Bioinformatics* **21,** 1267–1268.
217. Raghava, G. P., Searle, S. M., Audley, P. C., Barber, J. D., and Barton, G. J. (2003) OXBench: a benchmark for evaluation of protein multiple sequence alignment accuracy. *BMC Bioinform*. **4,** 47.
218. Thompson, J. D., Koehl, P., Ripp, R., and Poch, O. (2005) BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark. *Proteins* **61,** 127–136.
219. Sauder, J. M., Arthur, J. W., and Dunbrack, R. L., Jr. (2000) Large-scale comparison of protein sequence alignment algorithms with structure alignments. *Proteins* **40,** 6–22.
220. Pang, A., Smith, A. D., Nuin, P. A., and Tillier, E. R. (2005) SIMPROT: using an empirically determined indel distribution in simulations of protein evolution. *BMC Bioinform*. **6,** 236.
221. Nuin, P. A., Wang, Z., and Tillier, E. R. (2006) The accuracy of several multiple sequence alignment programs for proteins. *BMC Bioinform*. **7,** 471.
222. Stoye, J., Evers, D., and Meyer, F. (1998) Rose: generating sequence families. *Bioinformatics* **14,** 157–163.
223. Eidhammer, I., Jonassen, I., and Taylor, W. R. (2000) Structure comparison and structure patterns. *J. Comput. Biol*. **7,** 685–716.
224. Carugo, O. and Pongor, S. (2001) A normalized root-mean-square distance for comparing protein three-dimensional structures. *Protein Sci*. **10,** 1470–1473.
225. Armougom, F., Moretti, S., Keduas, V., and Notredame, C. (2006) The iRMSD: a local measure of sequence alignment accuracy using structural information. *Bioinformatics* **22,** e35–39.
226. Chew, L. P., Huttenlocher, D., Kedem, K., and Kleinberg, J. (1999) Fast detection of common geometric substructure in proteins. *J. Comput. Biol*. **6,** 313–325.
227. O'Sullivan, O., Zehnder, M., Higgins, D., Bucher, P., Grosdidier, A., and Notredame, C. (2003) APDB: a novel measure for benchmarking sequence alignment methods without reference alignments. *Bioinformatics* **19(Suppl 1),** i215–221.

228. Henikoff, S. and Henikoff, J. G. (1992) Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA* **89,** 10915–10919.
229. Dayhoff, M. O., Eck, R. V., and Park, C. M. (1972) A model of evolutionary change in proteins. In *Atlas of Protein Sequence and Structure* (Dayhoff, M. O., ed.). National Biomedical Research Foundation, Washington, DC, pp. 89–99.
230. Dayhoff, M. O., Schwartz, R. M., and Orcutt, B. C. (1978) A model of evolutionary change in proteins. In *Atlas of Protein Sequence and Structure* (Dayhoff, M. O., ed.). National Biomedical Research Foundation, Washington, DC, pp. 345–352.
231. Muller, T. and Vingron, M. (2000) Modeling amino acid replacement. *J. Comput. Biol*. **7,** 761–776.
232. Whelan, S. and Goldman, N. (2001) A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach. *Mol. Biol. Evol*. **18,** 691–699.
233. Prlic, A., Domingues, F. S., and Sippl, M. J. (2000) Structure-derived substitution matrices for alignment of distantly related sequences. *Protein Eng*. **13,** 545–550.
234. Reese, J. T. and Pearson, W. R. (2002) Empirical determination of effective gap penalties for sequence comparison. *Bioinformatics* **18,** 1500–1507.
235. Arribas-Gil, A., Gassiat, E., and Matias, C. (2006) Parameter estimation in pair-hidden Markov models. *Scand. J. Stat*. **33,** 651–671.
236. Liu, J. S., Neuwald, A. F., and Lawrence, C. E. (1995) Bayesian models for multiple local sequence alignment and Gibbs sampling strategies. *J. Am. Stat. Assoc*. **90,** 1156–1170.
237. Zhu, J., Liu, J. S., and Lawrence, C. E. (1998) Bayesian adaptive sequence alignment algorithms. *Bioinformatics* **14,** 25–39.
238. Kececioglu, J. and Kim, E. (2007) Simple and fast inverse alignment. RECOMB.
239. Yu, C.-N., Joachims, T., Elber, R., and Pillardy, J. (2007) Support vector training of protein alignment models. RECOMB.
240. Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. (2005) Large margin methods for structured and interdependent output variables. *J. Mach. Learn. Res*. **6,** 1453–1484.
241. Katoh, K. and Toh, H. (2007) PartTree: an algorithm to build an approximate tree from a large number of unaligned sequences. *Bioinformatics* **23,** 372–374.
242. Ahola, V., Aittokallio, T., Vihinen, M., and Uusipaikka, E. (2006) A statistical score for assessing the quality of multiple sequence alignments. *BMC Bioinform*. **7,** 484.
243. Altschul, S. F. (1998) Generalized affine gap costs for protein sequence alignment. *Proteins* **32,** 88–96.
244. Zachariah, M. A., Crooks, G. E., Holbrook, S. R., and Brenner, S. E. (2005) A generalized affine gap model significantly improves protein sequence alignment accuracy. *Proteins* **58,** 329–338.
245. Thompson, J. D., Muller, A., Waterhouse, A., Procter, J., Barton, G. J., Plewniak, F., et al. (2006) MACSIMS: multiple alignment of complete sequences information management system. *BMC Bioinform.* **7,** 318.

246. Thompson, J. D., Holbrook, S. R., Katoh, K., Koehl, P., Moras, D., Westhof, E., et al. (2005) MAO: a multiple alignment ontology for nucleic acid and protein sequences. *Nucleic Acids Res*. **33,** 4164–4171.
247. Gotoh, O. (1999) Multiple sequence alignment: algorithms and applications. *Adv. Biophys*. **36,** 159–206.
248. Phillips, A., Janies, D., and Wheeler, W. (2000) Multiple sequence alignment in phylogenetic analysis. *Mol. Phylogenet. Evol*. **16,** 317–330.
249. Lambert, C., Campenhout, J. M. V., DeBolle, X., and Depiereux, E. (2003) Review of common sequence alignment methods: clues to enhance reliability. *Curr. Genom*. **4,** 131–146.
250. Wallace, I. M., Blackshields, G., and Higgins, D. G. (2005) Multiple sequence alignments. *Curr. Opin. Struct. Biol*. **15,** 261–266.
251. Edgar, R. C. and Batzoglou, S. (2006) Multiple sequence alignment. *Curr. Opin. Struct. Biol*. **16,** 368–373.
252. Morrison, D. A. (2006) Multiple sequence alignment for phylogenetic purposes. *Aust. Syst. Bot*. **19,** 479–539.
253. Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001) *Introduction to Algorithms*. MIT Press, Cambridge, MA.
254. Eppstein, D. (2000) Fast hierarchical clustering and other applications of dynamic closest pairs. *J. Exp. Algorithmics* **5**, 1–23.
255. Elias, I. and Lagergren, J. (2005) Fast neighbor joining. ICALP.
256. Waterman, M. S., Eggert, M., and Lander, E. (1992) Parametric sequence comparisons. *Proc. Natl. Acad. Sci. USA* **89,** 6090–6093.
257. Waterman, M. S. (1994) Parametric and ensemble sequence alignment algorithms. *Bull. Math. Biol*. **56,** 743–767.
258. Gusfield, D., Balasubramanian, K., and Naor, D. (1994) Parametric optimization of sequence alignment. *Algorithmica* **12,** 312–326.