4364–4374 *Nucleic Acids Research, 2006, Vol. 34, No. 16* *Published online 26 August 2006*
doi:10.1093/nar/gkl514

MUMMALS: multiple sequence alignment improved by using hidden Markov models with local structural information

Jimin Pei[1,*] and Nick V. Grishin[1,2]

[1]Howard Hughes Medical Institute and [2]Department of Biochemistry, University of Texas Southwestern Medical Center at Dallas, 5323 Harry Hines Boulevard, Dallas, TX 75390-9050, USA

Received May 10, 2006; Revised June 26, 2006; Accepted July 5, 2006

ABSTRACT

We have developed MUMMALS, a program to construct multiple protein sequence alignment using probabilistic consistency. MUMMALS improves alignment quality by using pairwise alignment hidden Markov models (HMMs) with multiple match states that describe local structural information without exploiting explicit structure predictions. Parameters for such models have been estimated from a large library of structure-based alignments. We show that (i) on remote homologs, MUMMALS achieves statistically best accuracy among several leading aligners, such as ProbCons, MAFFT and MUSCLE, albeit the average improvement is small, in the order of several percent; (ii) a large collection (>10 000) of automatically computed pairwise structure alignments of divergent protein domains is superior to smaller but carefully curated datasets for estimation of alignment parameters and performance tests; (iii) reference-independent evaluation of alignment quality using sequence alignment-dependent structure superpositions correlates well with reference-dependent evaluation that compares sequence-based alignments to structure-based reference alignments.

INTRODUCTION

Genome sequencing events have resulted in a rapid accumulation of protein sequences in public databases. As an essential tool in computational sequence analysis, sequence alignment is widely used in similarity searches, structure modeling, functional prediction and phylogenetic analysis (1–5). Construction of a multiple sequence alignment aims at arranging residues with inferred common evolutionary origin in the same position for a set of sequences (6). Valuable information regarding position-specific residue usage and conservation can be extracted numerically from a multiple sequence alignment for various applications.

A classical method for constructing a multiple alignment aligns sequences progressively, as exemplified by the program ClustalW (7). Guided by a tree or a dendrogram that reflects the similarities among sequences, a progressive method makes a series of pairwise alignments for neighboring sequences or pre-aligned sequence groups. In this way, similar sequences are aligned prior to divergent sequences. However, progressive methods do not correct errors made in each pairwise alignment step. While similar sequences can be aligned with acceptable quality, progressive methods using general amino acid substitution matrices have limited success in obtaining high-quality alignments for divergent sequences (8).

Two main techniques are utilized to correct or minimize mistakes made in the progressive alignment process. One is iterative refinement of the alignment after the progressive steps (9–11), e.g. by repeatedly dividing the aligned sequences into two groups and realigning the groups. The other technique, pioneered by the program T-COFFEE (12), makes a consistency measure among a set of pairwise sequence alignments before the progressive alignment steps. Such consistency-based scoring functions are superior to scoring functions based on general amino acid substitution matrices (12).

Improving pairwise sequence alignments is a key to provide high-quality starting materials for a consistency measure. The most common technique of constructing a protein pairwise alignment utilizes a substitution matrix of amino acids and a dynamic programming algorithm with gap penalties (13,14). Commonly used substitution matrices, such as the BLOSUM or PAM series matrices (15,16) are derived from large-scale analysis of relatively similar sequences. Other matrices have been derived from distant homologs based on structural alignments (17,18), and they are more suitable for aligning divergent sequences. Several studies have also shown that real or predicted local structural information can be used to improve pairwise alignment quality (19,20).

*To whom correspondence should be addressed. Tel: +1 214 645 5951; Fax: +1 214 645 5948; Email: jpei@chop.swmed.edu

Another approach to constructing a pairwise alignment relies on a hidden Markov model (HMM) (21,22). In a simple HMM for global pairwise alignment, aligned residue pairs are modeled by a hidden match state, while insertions and deletions are modeled by two hidden states that generate unmatched residues in either of the two sequences (22). In addition to being able to find an optimal alignment, a pairwise alignment HMM can be used to estimate the posterior probability of any residue in the first sequence being aligned to any residue in the second sequence (22). These residue match probabilities have been used in ProbCons (23), a multiple sequence alignment program based on probabilistic consistency.

In this study, we aim to improve alignment quality by using: (i) more complex pairwise alignment HMMs that incorporate local structural information and (ii) better estimation of HMM parameters from a large set of structural alignments of divergent domain pairs from SCOP database (24). We show that these two techniques improve both pairwise alignments and probabilistic consistency-based multiple sequence alignments in reference-dependent and reference-independent tests.

MATERIALS AND METHODS

Training and testing datasets of SCOP domain pairs

Protein domain sequences and structures were obtained from the ASTRAL database (25) based on SCOP (24) version 1.69. We used the dataset consisting of representatives at the 40% sequence identity threshold (SCOP40). Domains from SCOP classes 1 to 4 [all alpha, all beta, alpha and beta (a/b), alpha and beta (a + b)] were selected. For each domain pair from the same superfamily of SCOP40, we computed a structural alignment using the program DaliLite (26). Alignments with coverage (fraction of aligned region to the length of the shorter sequence) less than 0.6 were removed. The remaining alignments were divided into four datasets corresponding to four sequence identity bins: <10, 10–15, 15–20 and 20–40%. The method for calculating sequence identity that takes into account unaligned regions is described in Supplementary Data. To apply cross validation training and testing, we divided the domain pairs in each identity bin into four subsets at the SCOP fold level, so that no domains belonging to the same SCOP fold were shared by any two subsets. We estimated HMM parameters using alignments in three subsets and tested the programs on representative domain pairs of the remaining fourth subset. This procedure was performed four times corresponding to four ways of partitioning into training and testing datasets. To obtain representative alignments from a subset for testing, we randomly selected one domain pair from each SCOP fold in that subset. The number of domain pairs in each training dataset is larger than 10 000 (Supplementary Table S1).

HMMs of pairwise sequence alignment

Description of HMM structures. The standard pairwise alignment HMM has three hidden states emitting residues: a single match state 'M' emitting residue pairs, an 'X' state emitting residues in the first sequence and a 'Y' state emitting residues in the second sequence (Figure 1b) (22). This model structure is named HMM_1_1_0 (the format of HMM names is described below and in Table 1 and Supplementary Table S2).

The novelty of our complex HMMs is the introduction of more match states based on DaliLite structural alignments, which have aligned core blocks (structurally superimposable, shown in uppercase letters) and unaligned regions (structurally not superimposable, shown in lowercase letters)

(a) Structure-based alignment and hidden state paths

```
ss type:           ccceeeeeeeeeccccccchhhhhhhhhcc  cchhhhhhhhhhccccccccc
Sequence 1:    -----mDPFLVLLHSVSSSLsssELTELKYLCLc--aGRVGKRKLERVQatetqqs
Sequence 2:    mdakarNCLLQHREALEKDIa-aKTSYIMDHMIsdegGFLTISEEEKVRnep----
HMM_1_1_0:     YYYYYMMMMMMMMMMMMMMMXMMMMMMMMMMMMYYMMMMMMMMMMMMMMMXXXX
HMM_1_1_1:     YYYYYUMMMMMMMMMMMMMMUXUMMMMMMMMMMUYYUMMMMMMMMMMMMUUUXXXX
HMM_1_3_1:     YYYYYUCCSSSSSSSSSCCCUXUHHHHHHHHHCUYYUCHHHHHHHHHHCUUUXXXX
```

(b) HMM_1_1_0 **(c)** HMM_1_1_1 **(d)** HMM_1_3_1

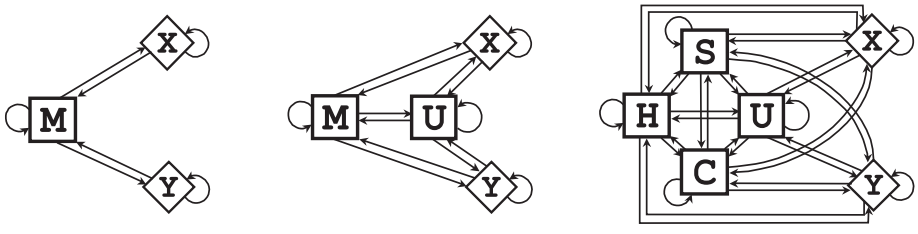

Figure 1. (**a**) An illustration of structure-based sequence alignment and hidden state paths. In Sequences 1 and 2, uppercase letters and lowercase letters represent aligned core blocks and unaligned regions, respectively. If two corresponding unaligned regions bounded by the same two core blocks are of different length, we split the shorter one into two pieces and introduce contiguous gaps in the middle. For both N- and C-terminal ends, the shorter unaligned region is pushed toward the core blocks. Secondary structure (ss) types (helix, 'h'; strand, 'e'; coil, 'c') are shown for Sequence 1. The hidden state paths for three models are shown below the amino acid sequences. (**b**) Model structure of HMM_1_1_0. Residue pairs in unaligned regions are modeled using the same match state ('M') as those in the aligned blocks. Insertions in the first sequence and second sequence are modeled using states 'X' and 'Y', respectively. (**c**) Model structure of HMM_1_1_1. Residue pairs in the unaligned regions are modeled using a different match state ('U') than the match state in the core blocks ('M'). (**d**) Model structure of HMM_1_3_1. Residue pairs in aligned core blocks are modeled using three match states ('H', 'S', 'C') according to three secondary structure types of the first sequence. In (b), (c) and (d), match states are shown as squares and insertion states are shown as diamonds. Begin state, end state, and transitions from or to them are present in these models, but are not shown.

(Figure 1a). Residue pairs in aligned core blocks can be modeled by a match state ('M'). If the two unaligned regions in between two core blocks have different and non-zero lengths, they can be modeled using a different 'match' state ('U', for unaligned) (Figure 1c) and 'X' or 'Y' state. This is done by evenly splitting the unaligned residues in the sequence with the shorter unaligned region into two segments, placing them beside the two adjacent core blocks and separating them by a single stretch of gap symbols. In this way, transitions between 'X' and 'Y' do not occur. The residue pair emission probabilities of these artificially forced matches in unaligned regions ('U') are expected to be different from those in the core blocks ('M'). This model (HMM_1_1_1) has two distinct match states: 'M' and 'U' (Figure 1a and c).

Furthermore, the residue matches in aligned core blocks can be modeled by several distinct match states based on the local structure information of the first sequence, for instance, secondary structure types (Figure 1d). Since residue usage and substitution patterns can vary for different local structural environments, the three match states according to secondary structure types have different sets of emission probabilities of 400 residue pairs. Secondary structure types were assigned using program DSSP (27) for SCOP domains. The model that incorporates three secondary structure types ('H': helix, 'S': strand and 'C': coil) of the first sequence and unaligned match state ('U') is named HMM_1_3_1 (Figure 1d). Similarly, multiple match states can be introduced based on several solvent accessibility categories, or a combination of solvent accessibility categories and secondary structure types. Solvent accessibilities of residue sidechains were calculated by program NACCESS (28). Three solvent accessibility categories were used based on our previous studies (29). The model that incorporates three solvent accessibility categories and unaligned match state is named HMM_3_1_1. Combination of three solvent accessibility categories and three secondary structure types results in nine classes of match states for the core blocks (model HMM_3_3_1).

Each HMM model is named in the format 'HMM_solv_ss_u', where 'solv' is the number of solvent accessibility categories, 'ss' is the number of secondary structure types, and 'u' is 1 if unaligned regions are modeled with an additional match state. Some statistics about the five HMMs, such as the number of hidden states, the number of parameters and time complexity, are listed in Supplementary Table S2.

Estimation of HMM parameters. We used a supervised learning method to estimate HMM parameters from DaliLite structural alignments, since hidden state paths are known for them (Figure 1a) (22). These parameters are: transition probabilities among hidden states [match state(s), 'X' state, 'Y' state, begin state and end state], emission probabilities of residue pairs for each match state, and residue emission probabilities for 'X' state and 'Y' state (22). Given a training dataset, we assign a weight to each sequence. If a sequence occurs N times in the dataset, its weight is $1/N^{1/2}$. For large SCOP superfamilies, one sequence is involved in a large number of pairwise alignments. Sequence weighting aims to reduce the bias introduced by large superfamilies and folds. The parameters were estimated from weighted counts of observed transitions and emissions in DaliLite alignments (22).

The supervised learning method was also used to estimate model parameters from pairwise alignments in BAliBASE2.0 (30), from which ProbCons derived its HMM parameters with a different approach (23). These alignments also have core blocks assigned (uppercase letters). However, local structural information is not available for every sequence. Therefore, from BAliBASE2.0 pairwise alignments, we derived parameters only for models HMM_1_1_0 and HMM_1_1_1 that do not have local structure-dependent match states.

Pairwise alignment with optimal match probabilities. We apply standard forward and backward algorithms (22) to align two sequences $x = (x_1, x_2, \ldots, x_m)$ and $y = (y_1, y_2, \ldots, y_n)$ (m and n are the lengths of x and y). Given a pairwise HMM, the forward algorithm applies a dynamic programming technique to calculate the probability of observing two subsequences $(x_1, \ldots, x_i)$ and $(y_1, \ldots, y_j)$ generated by an HMM, with the last position of the two subsequences being in a certain hidden state. The formula of forward probability is:

$$\mathrm{F}_K(i,j) = \mathrm{P}[x_1, \ldots, x_i, y_1, \ldots, y_j, \pi(i,j) = K] \quad \mathbf{1}$$

Here, $\pi(i, j)$ is the hidden state of the last position. K is one of the hidden states. For example, for model HMM_1_3_1, K is from the set {'H', 'S', 'C', 'U', 'X', 'Y'} (Figure 1d).

The backward algorithm calculates the probability of generating two subsequences $(x_{i+1}, \ldots, x_m)$ and $(y_{j+1}, \ldots, y_n)$, given the condition that the previous position of the two subsequences is of a certain type of hidden state. The formula of the backward probability is:

$$\mathrm{B}_K(i,j) = \mathrm{P}[x_{i+1}, \ldots, x_m, y_{j+1}, \ldots, y_n \mid \pi(i,j) = K] \quad \mathbf{2}$$

The following formula can be derived (22):

$$\mathrm{P}[\pi(i,j) = \mathrm{K} \mid x, y] = \mathrm{F}_K(i,j) * \mathrm{B}_K(i,j)/\mathrm{P}(x,y), \quad \mathbf{3}$$

where $\mathrm{P}[\pi(i, j) = \mathrm{K} \mid x, y]$ is the probability of aligned residue pair (x_i, y_j) being generated by a certain hidden match state K given the two full-length sequences x and y, and $\mathrm{P}(x, y)$ is the full probability of observing sequences x and y given the model. $\mathrm{P}(x, y)$ can be calculated by forward algorithm or backward algorithm (22).

The posterior probability (22,31) of residue i in sequence x being aligned to residue j in sequence y (match probability) can be calculated as:

$$\mathrm{P}(x_i \text{ aligned to } y_j) = \sum_{K_m} \mathrm{P}[\pi(i,j) = K_m \mid x, y] \quad \mathbf{4}$$

Here, K_m belongs to a set of match states. For model HMM_1_3_1, K_m is from the set {'H', 'S', 'C', 'U'}.

We then find the alignment path that maximizes the sum of match probabilities of residue pairs in the alignment path using dynamic programming with zero gap penalties. It has been shown that this technique can give better alignment quality than Viterbi algorithm (23).

Multiple sequence alignment procedure

We have developed a progressive multiple sequence alignment program implementing our HMMs (MUMMALS, standing for MUltiple alignment with Multiple MAtch state models of Local Structure). It applies a probabilistic consistency scoring function similar to the one in ProbCons (23). First, a tree is built-in a fast way based on a *k*-mer count method (10). An initial alignment is built progressively guided by the tree with a simple sum-of-pairs scoring function. A second tree is then built with a UPGMA method based on sequence identities calculated from the initial alignment. We then apply the probabilistic consistency strategy as described by Do *et al*. (23). For each sequence pair, we calculate the match probabilities of residue pairs using one of the HMMs. These probability matrices are subject to consistency measure, which involves multiplications of the matrices (23). Finally, MUMMALS progressively aligns the sequences guided by the second tree using the consistency-based scoring function.

To properly balance alignment speed and accuracy, we have also applied a two-stage alignment strategy similar to the one used in our program PCMA (32). In the first stage, highly similar sequences are progressively aligned in a fast way without consistency scoring. The scoring function in this stage is a weighted sum-of-pairs measure of BLOSUM62 scores (16). If two neighboring groups on a tree have an average sequence identity higher than a certain threshold, they are aligned in this fast way. The result of the first stage is a set of pre-aligned groups that are relatively divergent from each other. In the second alignment stage, one representative sequence is selected from each pre-aligned group, and these representative sequences are subject to the more time-consuming probabilistic consistency measure. Then the representatives are aligned progressively according to the consistency-based scoring function. Finally, the pre-aligned groups obtained in the first stage are merged according to the alignment of the representatives to obtain the final alignment of all sequences.

Performance assessment of multiple sequence alignment programs

To test multiple sequence alignment programs, we used the same representative SCOP pairs selected for pairwise alignment tests (Supplementary Table S1). For each domain, we added up to 24 homologs, which were obtained from PSI-BLAST searches (1) with five iterations and an *E*-value cut-off of 0.0001. We removed highly similar sequences (>97% identity) and sequence fragments (coverage less than half of the query) before randomly selecting up to 24 homologs. This procedure is similar to the one used to build the PREFAB database (10). Similar to pairwise alignment tests, the 4-fold cross validation approach is applied to the tests of MUMMALS.

Two large testing datasets compiled by other researchers were used as well. One is the SABmark database (version 1.65) (33), which is based on homologous SCOP domains, and the other is PREFAB database (version 4) (10), which is based on structural alignments and homologous sequences from database searches. Alignment quality scores (*Q*-scores) were calculated using the built-in programs in SABmark and PREFAB packages. The *Q*-score is the fraction of the number of correctly aligned residue pairs in the test alignment to the total number of aligned residue pairs in the reference alignment. Additional series of smaller datasets constructed by MAFFT authors (11) were 55 HOMSTRAD (34) structural alignments with various numbers of close homologs added ('HOM + X', 'X' is the number of close homologs added and is 0, 20, 50 or 100). To obtain more balanced datasets from 'HOM + 100', we used BLASTCLUST to cluster sequences in any 'HOM + 100' alignment at 50 or 60% identity level, and selected up to two homologs in each sequence cluster. The resulting datasets are called 'HOM + 100_bcl50' and 'HOM + 100_bcl60'. We also tested programs on 218 BAliBASE3.0 (35) alignments with full-length sequences. The column scores (the fraction of entirely correct columns) were reported in addition to *Q*-scores for BAliBASE3.0. Wilcoxon signed-ranks tests were performed to calculate statistical significance of comparisons between alignment programs, which include ProbCons (version 1.10) (23), MAFFT (version 5.667) (11) with several options, MUSCLE (version 3.52) (10) and ClustalW (version 1.83) (7).

In addition to *Q*-score, we utilized a reference-independent assessment of alignment quality from a structural modeling perspective. Given a test sequence alignment of two SCOP domains with known structures, corresponding Cα atoms in aligned residue pairs were used as equivalent points for structural superposition to minimize the root mean square distance (RMSD). The structural similarity scores were calculated based on this forced superposition of the aligned residue pairs. The following scores were used: DALI *Z*-score (36), GDT score (37), TM-score (38), 3D-score (39) and two LiveBench contact scores (39) (see Supplementary Data for the equations of 3D-score and LiveBench contact scores). An alignment with better quality should have a higher similarity of the two structures superimposed in a sequence-dependent way described above, and thus have a larger structural score. Two reference-independent sequence similarity scores were also calculated for aligned residue pairs in a test alignment: sequence identity and average BLOSUM62 scores (16). Additionally, each of the scores was rescaled to take into account a random model (the reversed alignment), self-comparisons and alignment coverage (see Supplementary Data for details). Such rescaling places each score between 0 (random match) and 1 (self-match) and makes different scoring schemes more comparable to each other.

Availability

The MUMMALS multiple alignment web server is at: http://prodata.swmed.edu/mummals/, with the source code of MUMMALS available for download.

RESULTS

Complex HMMs with local structural information improve pairwise alignments

Applying a cross validation approach, we tested our pairwise alignment HMMs on representative domain pairs from the ASTRAL SCOP40 dataset (Table 1 and Supplementary Table S3). Model HMM_1_1_0 represents the simplest

Table 1. Average *Q*-scores in pairwise alignment tests on representative SCOP40 domain pairs

Method/Model	Testing datasets SCOP 0–10% (355)	SCOP 10–15% (432)	SCOP 15–20% (420)	SCOP 2–40% (578)	SCOP All (1785)
HMM_1_1_0[a]	0.146	0.322[c]	0.568[c]	0.851[c]	0.516[c]
HMM_1_1_1[a]	0.146	0.328[c]	0.573[c]	0.855[c]	0.520[c]
HMM_3_1_1[a]	0.150[c]	0.327[c]	0.574[c]	0.858[c]	0.521[c]
HMM_1_3_1[a]	**0.152**[c]	0.334[c]	0.585[c]	0.858[c]	0.526[c]
HMM_3_3_1[a]	0.151[c]	**0.335**[c]	**0.586**[c]	**0.860**[c]	**0.527**[c]
HMM_1_1_0[b]	0.123	0.295	0.551	0.843	0.498
HMM_1_1_1[b]	0.132	0.31	0.572	0.851	0.511
ProbCons	0.116	**0.294**	**0.536**	**0.833**	**0.490**
MAFFT-fftnsi	0.087	0.256	0.496	0.809	0.457
MAFFT-einsi	0.081	0.248	0.491	0.809	0.453
MAFFT-linsi	0.116	0.262	0.495	0.794	0.460
MAFFT-ginsi	0.116	0.265	0.496	0.794	0.461
MUSCLE	**0.139**	0.293	0.507	0.817	0.482
ClustalW	0.136	0.27	0.482	0.809	0.467

Each HMM is named in the format 'HMM_solv_ss_u', where 'solv' is the number of solvent accessibility categories, 'ss' is the number of secondary structure types, and 'u' is 1 if unaligned regions are modeled with an additional match state. Average *Q*-scores of four testing datasets with different identity ranges are shown. *Q*-score is the number of correctly aligned residue pairs in the test alignment divided by the total number of aligned residue pairs in the reference alignment. The number of alignments in each testing dataset is shown in parentheses and the identity range in % is specified above the number of alignments. The best results of our models and the best results of other programs are in bold numbers.

[a]Trained on DaliLite alignments of SCOP40 domain pairs with 20–40% identity.

[b]Trained on BAliBASE2.0 pairwise alignments.

[c]Our model is statistically better than the best of other programs according to Wilcoxon signed-rank test ($P < 0.015$).

model with a single match state (Figure 1a). It can serve as a control for more complex models, including HMM_1_1_1 that assigns residue pairs in unaligned regions to a different match state, HMM_1_3_1 that captures secondary structure information, HMM_3_1_1 that captures solvent accessibility information, and HMM_3_3_1 that has both types of local structural information (see Materials and Methods).

Several conclusions can be drawn from Table 1 and Supplementary Table S3. (i) For each testing dataset, the best performance is usually achieved with parameters estimated on the training dataset with the same identity range (Supplementary Table S3). For example, models trained on alignments with the highest identity range (20–40%) exhibit best performance on testing alignments in that range. Thus residue substitution and insertion/deletion statistics are different for sequence pairs with different degrees of similarity, in agreement with other reports (16,17). Although models trained on pairs with the lowest identity range can perform slightly better (~2%) on representatives in that range, they perform much worse on alignments with the highest identity range (nearly 10% decrease as compared to models trained on pairs with the highest identity range). (ii) More complex HMMs improve alignment quality. For example, HMM_1_3_1 performs better than HMM_1_1_1 in almost every case. HMM_1_3_1 performs almost equally well to HMM_3_3_1. (iii) Our models trained on pairs with 20–40% identity range perform better than several other aligners, such as ProbCons (23), MAFFT (11) with several option, MUSCLE (10) and ClustalW (7) on any testing dataset. Although the simplest model HMM_1_1_0 has the same model structure as the one used in ProbCons, it gives ~3–4% increase over ProbCons for every testing dataset. The best models HMM_1_3_1 and HMM_3_3_1 give 4–5% increase over ProbCons. (iv) When trained on BAliBASE2.0, HMM_1_1_0 and HMM_1_1_1 exhibit inferior performance compared to the same models trained on SCOP domain pairs with identity above 20%, suggesting that SCOP domain pairs with above 20% identity is a better training dataset than BAliBASE2.0. The two models (especially HMM_1_1_1) trained on BAliBASE2.0 nevertheless give slightly better performance than ProbCons, which was also trained on BAliBASE2.0 but with a different training method (unsupervised learning) (23).

To further compare the performance of our HMMs and the one used in ProbCons, we tested sequence pairs from BAliBASE2.0, on which the HMM in ProbCons was trained (Supplementary Table S4). BAliBASE2.0 is a manually curated database for testing multiple alignment programs. It contains core blocks (uppercase letters) with good quality and a large fraction of lowercase letter regions where high alignment quality is not guaranteed. HMM_1_1_0 and HMM_1_1_1 trained on DaliLite alignments of SCOP domain pairs with above 20% identity show better results than these two models trained on BAliBASE2.0 as well as ProbCons, suggesting that the structural alignments of relatively divergent SCOP domains form a better training set than BAliBASE2.0. HMM_1_3_1 trained on the same SCOP domain pairs gives the best BAliBASE2.0 results (Supplementary Table S4), confirming that a more complex model with multiple match states of secondary structures improves alignment quality.

MUMMALS and multiple sequence alignment tests

We implemented our HMM models in a multiple sequence alignment program based on probabilistic consistency. Our program is named MUMMALS, standing for MUltiple alignment with Multiple MAtch state models of Local Structure. MUMMALS was compared to several other programs, such as ProbCons (23), MAFFT (11), MUSCLE (10)

Table 2. Average alignment scores in tests of multiple sequence alignment programs

Methods/Models	Testing datasets								
	SCOP 0–10% (355)	SCOP 10–15% (432)	SCOP 15–20% (420)	SCOP 20–40% (578)	SCOP All (1785)	PREFAB (1682)	SABmark Sup (425)	SABmark Twi (209)	BAliBASE3.0 Q/col (218)
HMM_1_1_0	0.313	0.514[a]	0.727[a]	0.885	0.644[a]	0.723[a]	0.516[a]	0.193[a]	0.862/0.551
HMM_1_1_1	0.313	0.512[a]	0.728[a]	0.886	0.644[a]	0.724[a]	0.512[a]	0.186[a]	0.861/0.550
HMM_3_1_1	0.321[a]	0.514[a]	0.730[a]	0.888	0.647[a]	0.726[a]	0.516[a]	0.186[a]	0.862/0.554
HMM_1_3_1	0.327[a]	0.518[a]	0.732[a]	0.889[a]	0.650[a]	0.729[a]	0.519[a]	0.194[a]	0.863/0.554
HMM_3_3_1	**0.329**[a]	**0.520**[a]	**0.733**[a]	**0.889**[a]	**0.651**[a]	**0.731**[a]	**0.522**[a]	**0.196**[a]	**0.863/0.557**
ProbCons	0.291	0.486	0.702	0.879	0.625	0.716	0.485	0.166	0.862[c]/0.556[b]
MAFFT-fftnsi	0.283	0.472	0.673	0.865	0.608	0.7	0.45	0.147	0.829[c]/0.515[c]
MAFFT-einsi	0.293	0.498	0.71	0.882	0.631	0.72	0.502	0.175	0.866[c]/0.585[b]
MAFFT-linsi	0.301	**0.5**	0.707	0.883	0.633	**0.722**	**0.51**	**0.184**	**0.868[c]/0.586[b]**
MAFFT-ginsi	**0.308**	0.497	**0.714**	**0.888**	**0.637**	0.715	0.495	0.176	0.840[c]/0.526[c]
MUSCLE	0.262	0.453	0.662	0.866	0.597	0.68	0.433	0.136	0.816[c]/0.472[c]
ClustalW	0.21	0.357	0.566	0.798	0.519	0.617	0.39	0.127	0.749[c]/0.373[c]

The format of the HMM names ('HMM_solv_ss_u') is explained in Table1. Average *Q*-scores are shown for all the testing datasets. For the BAliBASE3.0 dataset, both the *Q*-score ('Q', first number) and column score ('col', second number, fraction of entirely correct columns) are shown. The first four testing datasets are representative SCOP40 domain pairs with added homologs. SABmark has 'superfamily' dataset (sup) and 'twilight zone' dataset (twi). The number of alignments in each testing dataset is shown in parentheses and the identity range in % is specified above the number of alignments for SCOP datasets. MUMMALS implementing different HMMs are the first five methods. All sequences pairs are subject to consistency measure in MUMMALS. The best scores of MUMMALS and the best scores of other programs are in bold.

[a]MUMMALS with this model is statistically better than the best of other programs according to Wilcoxon signed-rank test ($P < 0.015$).

[b]For BAliBASE3.0 test, the difference between MUMMALS with model HMM_1_3_1 or HMM_3_3_1 and this program is not statistically significant ($P > 0.05$) according to Wilcoxon signed-ranks test.

[c]For BAliBASE3.0 test, MUMMALS with model HMM_1_3_1 or HMM_3_3_1 is statistically better than this program (*P*-value less than 0.01, except for *Q*-scores of ProbCons, for which $P = 0.017$).

and ClustalW (7). We assembled testing datasets of multiple sequences by adding homologs to pairwise alignments of SCOP domain pairs (see Materials and Methods). For three testing datasets with identity ranges below 20%, MUMMALS shows ~3–4% increase of average *Q*-score over ProbCons when the best-performing HMMs are selected (HMM_1_3_1 and HMM_3_3_1) (Table 2). For the dataset with identity range above 20%, the improvement is less prominent (~1%), although still statistically significant ($P < 0.005$). Among the other programs tested (ProbCons, several options of MAFFT, MUSCLE and ClustalW), MAFFT with options [lg]insi usually gives the best performance. Our program still outperforms any MAFFT option by ~2% for the three datasets with identity ranges below 20%.

The other two large datasets we have used for testing are PREFAB version 4 (PREFAB4) (10) and SABmark version 1.65 (33). PREFAB4 consists of 1682 alignments based on the consensus of two structural alignment programs SOFI (40) and CE (41), with up to 24 homologous sequences added from database searches. SABmark database has two alignment sets. The 'twilight zone' set contains SCOP (version 1.65) domain pairs with very low-to-low similarity and the 'superfamily' set contains SCOP domains with low to intermediate similarity. The reference alignments in SABmark were also derived from the consensus of SOFI and CE. The comparison of MUMMALS with the other programs on PREFAB4 and SABmark shows similar trends as seen in datasets assembled by us. MUMMALS with the best-performing models (HMM_1_3_1 and HMM_3_3_1) gives the highest average *Q*-scores, with ~2–4% increase over ProbCons and ~1% increase over the best of the MAFFT series (Table 2).

We also tested these programs on datasets of 55 HOMSTRAD structural alignments (11,34) with various numbers of homologs added (Supplementary Table S5). MUMMALS performs better than other programs for testing datasets 'HOM + 0', 'HOM + 20' and 'HOM + 50'. The results of MUMMALS on 'HOM + 100' are inferior to those on 'HOM + 50', which is in contrast to MAFFT that shows consistent increase of accuracy when more homologs are added. We reason that such a problem of MUMMALS is due to excessive number of close homologs in some sequence subgroups in 'HOM + 100' test cases, since our consistency measure does not take into account similarities among the testing sequences. To test this hypothesis, we removed closely related sequences in 'HOM + 100' alignment at 50 or 60% identity level (datasets 'HOM + 100_bcl50' and 'HOM + 100_bcl60'). The remaining sequences are more balanced in terms of similarities among them. We indeed obtained the best MUMMALS performance after removing closely related sequences (Supplementary Table S5). A similar degrading effect of closely related sequences is observed for ProbCons, which is also based on probabilistic consistency.

Recently, BAliBASE3.0 multiple alignment benchmark has been released (35). We tested the programs on 218 BAliBASE3.0 alignments with full-length sequences. These alignments represent difficult cases since N- or C-terminal non-homologous regions are included for some sequences. The average *Q*-scores and column scores (the fraction of entirely correctly columns) in manually defined core blocks are reported in Table 2. For *Q*-score measurement of alignment quality, the Wilcoxon signed rank tests show that MUMMALS with best-performing models (HMM_3_3_1 or HMM_1_3_1) is statistically better than other programs, albeit MUMMALS does not have the highest average *Q*-score. For column score measurement of alignment quality, MUMMALS with HMM_3_3_1 or HMM_1_3_1 has a lower average column score than MAFFT-[le]insi and ProbCons, but the difference between them is not statistically

significant. In fact MUMMALS with HMM_3_3_1 outperforms MAFFT-linsi (having the best average scores) in 106 cases while MAFFT-linsi performs better in 67 cases, measured by column score.

We found that the major cause of lower average column score of MUMMALS is due to a few alignments that MUMMALS gives considerably lower scores as compared to MAFFT-linsi (10 alignments with a score difference larger than 0.5). Manual inspections of these alignments revealed some interesting scenarios. In some cases, the low scores of MUMMALS are due to wrong alignment of a few divergent sequences, some of which have long extensions at the ends. Column score is very sensitive in such situations, since a low score is produced even if only one sequence is incorrectly aligned in many places. MAFFT alignments with local options ([le]insi) indeed show advantage in these situations (11). In several other cases, MUMMALS produces structurally meaningful alignments that are scored very low since they are not consistent with the reference alignments (Supplementary Figure S1). One example is testing case BB40037, for which the reference alignment aligns the 'fer2' domain [2Fe-2S iron sulfur cluster binding domain, named by Conserved Domain Database (42)] that is present in every sequence. Inspection of the sequences revealed two groups of sequences with different domain architectures in BB40037 (Supplementary Figure S1a). One group contains three domains in the order of 'fer2', 'FAD_binding_6' (oxidoreductase FAD-binding domain) and 'NAD_binding_1' (oxidoreductase NAD-binding domain). The other group contains the same three domains, but in the order of 'FAD_binding_6', 'NAD_binding_1' and 'fer2'. MUMMALS aligns 'FAD_binding_6' and 'NAD_binding_1' for the two groups and thus misaligned the 'fer2' domain. Since the reference-dependent evaluation is based on 'fer2' domain, MUMMALS alignment gets a column score of zero. Another evaluation problem is caused by domain duplication or repeats, as exemplified by test case BB40040. Most of BB40040 sequences have a single domain of carbonic anhydrase. However, the sequence 'CAH_DUNSA' has two such domains (Supplementary Figure S1b). The reference alignment aligns the first domain of 'CAH_DUNSA' to other sequences while MUMMALS aligns most of the second domain to other sequences, resulting in a low column score.

Secondary structure prediction and alignment quality

Model HMM_1_3_1 has hidden secondary structure-dependent match states that are used in alignment construction. However, secondary structure predictions are not explicitly generated and remain 'hidden' in the alignment process. A posterior decoding technique can be applied to this model to output explicit secondary structure predictions (see Supplementary Data for details). The prediction accuracy is ~60% on representative SCOP domains (Supplementary Table S6). This prediction accuracy is lower than many advanced methods, such as PSI-PRED (43), which use homologs from database searches to enhance the secondary structure signal of a single sequence. Our HMM-based prediction of secondary structure corresponds to a single sequence predictor, since position-specific information from multiple homologs is not used. The prediction accuracy of model HMM_1_3_1 is comparable to other methods that generate predictions based on a single sequence (44).

If secondary structure information is known for one of the aligned sequences, model HMM_1_3_1 can be modified to restrict the hidden state path to follow the experimental secondary structure (see Supplementary Data). Using real secondary structural information in MUMMALS with model HMM_1_3_1, we were able to obtain an average *Q*-score increase of 3% on SABmark datasets (Supplementary Table S8). Such a result suggests that one limitation of MUMMALS is the low accuracy of the implicit secondary structure prediction. Further improvement of alignment quality should be possible with more accurate secondary structure predictions that explore database homologs.

Evaluation of programs with reference-independent scores

All previously described tests are based on comparison with reference alignments that are treated as 'gold standards'. It is unclear how the quality of reference 'gold' alignments influences the ranking of alignment programs (45). Several examples from BAliBASE3.0 show some problems with reference-dependent evaluation. Since the structures of all domains in our testing datasets are known, we can evaluate the quality of a test alignment using reference-independent scores that reflect the similarity between two structures after they are superimposed according to aligned residues in the test alignment. The structural scores we have used are: DALI *Z*-score, TM-score, GDT-TS score, 3D-score and two LiveBench contact scores. From Table 3, it is clear that MUMMALS with the best-performing models (HMM_1_3_1 and HMM_3_3_1) produces higher average reference-independent structural scores than other programs for divergent domain pairs (identity below 20%), suggesting that our HMM models produce potentially more useful alignments in terms of structure modeling. Interestingly, MUMMALS does not produce the highest reference-independent sequence-based similarity scores (sequence identity and BLOSUM62 scores of aligned residue pairs). This result suggests that minimization of these sequence similarity measures alone does not lead to better alignment quality for divergent sequences. Our experiments with reference-independent evaluation shows good correlation with reference-dependent results described above, thus rendering a 'gold standard' reference unnecessary for alignment evaluation.

Pairwise comparisons of multiple sequence aligners

MUMMALS shows improvement over MAFFT and ProbCons by only a few percent in terms of average alignment quality scores on various testing datasets. Although this improvement is statistically significant and does not depend on the testing dataset (assembled by us, or by other researchers) or alignment quality measure (reference-dependent *Q*-score, or reference-independent DALI *Z*-score, TM-score, GDT-TS, etc), it is not clear what these average scores actually represent. To gain more understanding, we directly compared alignment scores of different programs for individual domain pairs. Table 4 shows the number of alignments where one program performs better than another program by 10% or more (measured by *Q*-score or

Table 3. Assessment of multiple sequence alignment programs using reference-independent sequence and structural similarity scores on 1207 representative SCOP40 domain pairs with identity <20%

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

The first five methods are MUMMALS implementing different HMMs. The format of the HMM names ('HMM_solv_ss_u') is explained in Table 1. The best scores of MUMMALS and the best scores of other programs (ProbCons, MAFFT with different options, MUSCLE, ClustalW) are in bold.
[a]MUMMALS with this model is statistically better than the best of other programs according to Wilcoxon signed-rank test ($P < 0.01$).

normalized TM-score) for 1207 divergent SCOP40 pairs with added homologs (sequence identity <20%). The numbers for such differences among MUMMALS family of programs with five different HMMs are relatively small, suggesting that MUMMALS with different HMMs generates alignments not that different from each other. The differences between MUMMALS and other programs are significantly larger. For example, MUMMALS with HMM_3_3_1 has a large *Q*-score increase (≥10%) over ProbCons on 201 alignments, while ProbCons is better by 10% or more on 62 alignments. For MUMMALS with HMM_3_3_1 and MAFFT-ginsi, these numbers are 199 and 117, respectively. Our comparisons thus suggest that different programs can explore somewhat different regions in alignment space, and the program giving lower average performance may be capable of generating better alignments in many cases. However, selection of a better alignment in the absence of structural comparison is still a difficult task (46).

DISCUSSION

Comparison of multiple sequence alignment programs

ProbCons (23) and MAFFT (11) are two of the most accurate multiple sequence alignment programs that explore only sequence information. They use different strategies to improve the accuracy of progressive alignment. ProbCons mainly relies on the consistency-based scoring function and MAFFT utilizes iterative refinement. MAFFT's scoring function is a weighted sum-of-pairs score of BLOSUM62, a general amino acid substitution matrix. Recent versions of MAFFT enhance performance by exploring aligned core regions with a simple consistency measure (11). However, the time-consuming ProbCons-type consistency operations on sequence triplets are not implemented in MAFFT. Our results show that MUMMALS based on probabilistic consistency can perform better than MAFFT, implying that consistency-based scoring function is superior to weighted sum-of-pairs measures of general substitution matrices.

Among existing multiple aligners, MUMMALS is methodologically closest to ProbCons. However, we implement HMMs that are different from the one used in ProbCons in three aspects. First, we introduce more complex model structures by increasing the number of match state types. While HMM_1_1_0, used mainly as a control, has the same model structure as the one in ProbCons, the other four models (HMM_1_1_1, HMM_1_3_1, HMM_3_1_1 and HMM_3_3_1) incorporate more match states, which take into account residue substitution differences in unaligned regions and aligned core blocks, as well as local structural differences in aligned core blocks. Second, we have estimated model parameters from DaliLite structural alignments of divergent SCOP domains (<40% identity), while the training data of ProbCons are from BAliBASE2.0. Third, we applied a supervised learning approach to estimating model parameters from structural alignments that have known hidden state paths. ProbCons applied an unsupervised learning approach (expectation maximization) to derive parameters without using reference alignments. Trained and tested on DaliLite structural alignments of SCOP40 domain pairs and on BAliBASE2.0 alignments, our models give better alignment quality than ProbCons and a number of other programs. We show that more complex models improve alignment quality and SCOP40 superfamily pairs with identity range of 20–40% provide a better training dataset than BAliBASE2.0 alignments.

Although we do not explicitly perform structure prediction in the course of alignment construction, our HMMs with multiple match states have built-in information of residue substitution characteristics in distinct local structure environments. Such information is learned from structural alignments in the training process and is encoded in the models. Since these match states are hidden states, our HMMs construct alignments using only sequence information. In this aspect, our method is different from some other methods that require direct use of real or predicted local structural information (19,47).

Time complexity of HMMs and MUMMALS

The trade-off for improvement of alignment quality with more complex model structures is more time for computation

Table 4. Number of large *Q*-score or TM-score differences (no less than 0.1) among the multiple sequence alignment programs on 1207 representative SCOP40 domain pairs with identity <20%

	HMM_1_1_0	HMM_1_1_1	HMM_3_1_1	HMM_1_3_1	HMM_3_3_1	ProbCons	MAFFT-fftnsi	MAFFT-einsi	MAFFT-linsi	MAFFT-ginsi	MUSCLE	ClustalW
HMM_1_1_0	—	13/8	41/18	29/15	**45/24**	50/157	105/191	108/163	119/127	85/102	92/208	73/399
HMM_1_1_1	14/14	—	35/16	18/6	**38/24**	53/156	100/189	103/171	109/124	90/110	91/209	76/397
HMM_3_1_1	28/45	22/40	—	16/24	**4/5**	39/164	86/211	101/183	109/135	81/124	82/225	63/418
HMM_1_3_1	19/29	9/22	26/20	—	**25/19**	46/155	90/202	107/169	110/124	89/116	92/214	70/408
HMM_3_3_1	**29/57**	**22/54**	**6/17**	**20/36**	—	**38/169**	**85/207**	**97/185**	**111/138**	**82/128**	**82/227**	**59/431**
ProbCons	187/76	186/80	203/67	189/73	**201/62**	—	149/133	152/98	172/80	162/76	162/169	110/336
MAFFT-fftnsi	334/123	334/118	354/113	344/115	**352/115**	242/154	—	123/98	137/71	153/76	115/143	88/342
MAFFT-einsi	234/156	232/152	260/147	247/152	**258/138**	152/186	116/225	—	80/37	133/96	134/193	93/369
MAFFT-linsi	206/138	204/140	237/131	219/143	**239/128**	133/188	87/207	62/82	—	85/98	93/196	60/387
MAFFT-ginsi	160/124	162/124	192/119	176/126	**199/117**	113/184	109/253	129/158	111/132	—	90/185	67/395
MUSCLE	370/94	374/94	390/83	384/90	**401/83**	302/138	218/136	309/134	295/110	327/106	—	75/288
ClustalW	627/67	628/66	645/60	645/65	**649/55**	559/103	498/96	582/91	585/70	600/76	449/100	—

The first five methods are MUMMALS implementing different HMMs. The format of the HMM names ('HMM_solv_ss_u') is explained in Table 1. Each none-diagonal cell has two numbers separated by a slash. The first number is the number of cases where the alignment quality score of the program listed to the left (in a row) is inferior to that of the program listed above (in a column) by 0.1 or more. The second number is the number of cases where the score of the 'row' program is better than that of the 'column' program by 0.1 or more. The alignment quality scores in the lower triangle and upper triangle are *Q*-scores and weighted and normalized TM-scores, respectively. Comparisons of MUMMALS with the best model (HMM_3_3_1) with other programs are highlighted in bold.

(Supplementary Table S2). HMM_3_3_1 (modeling both secondary structure and solvent accessibility) has the largest number of hidden states and is the most time-consuming model, running about three times slower than HMM_1_3_1 (modeling only secondary structure). However, the performance of these two models is fairly similar in pairwise or multiple sequence alignment tests (Tables 1–3). HMM_3_1_1 (modeling only solvent accessibility) has the same time complexity as HMM_1_3_1, but usually shows slightly inferior performance to HMM_1_3_1. These observations suggest that HMM_1_3_1 (modeling only secondary structure) is most efficient in balancing alignment accuracy and speed.

For MUMMALS, the rate-limiting steps are the computation of match probabilities using forward and backward algorithms (time order is $N^2 * L^2 * K^2$, N: number of sequences, L: average length, K: number of hidden states), and the computation of the consistency-based scoring function, which requires operations on all sequence triplets (time order is $N^3 * L^2$). If probabilistic consistency measure is applied to every sequence pair, MUMMALS is much slower than MAFFT, MUSCLE and ClustalW (Supplementary Table S9). For example, the median CPU time of MUMMALS with model HMM_1_3_1 on 1785 SCOP40 pairs with added homologs is 174s per alignment, as compared to 2.5s for MAFFT with 'ginsi' option (on Redhat Enterprise Linux 3, AMD Opteron 2.0 GHz). Applying a two-stage alignment strategy similar to the one used in the program PCMA (32), we were able to make MUMMALS almost an order of magnitude faster while still maintaining the same level of alignment accuracy on SCOP testing datasets (Supplementary Table S9). Since highly similar sequences can be aligned accurately with a general scoring function (8), there is no need for all of them to be subject to the time-consuming consistency measure. Although adding homologous sequences can in general improve alignment quality, our results on HOMSTRAD datasets suggest that proper balance of the similarity among the added homologs is also critical for MUMMALS as well as ProbCons.

Methods for alignment quality evaluation

Reference-dependent alignment evaluation relies on comparison of a test alignment to a reference alignment that is assumed to be correct. A commonly used alignment quality score is the fraction of correctly aligned residue pairs (*Q*-score). More complex scores have been designed to take into account the relative shifts of residues (48,49). Even though structure-based alignments can serve as high-quality references, there are several well-known drawbacks. First, structural alignments generated by automatic programs could still contain errors, especially for pairs with relatively low structural similarity, i.e. careful expert-driven multilateral research may generate alignments more meaningful than those obtained by programs. Second, defining the optimal structural alignment in certain regions is difficult for structurally divergent pairs, i.e. it is simply not possible to identify a single 'correct' alignment, which may not even exist. Third, for multi-domain proteins, especially those with repeats or duplications, multiple ways of aligning structurally similar parts exist, resulting in a variety of reference

alignments that are correct from structure modeling perspective. However, structure superposition programs usually provide a single best alignment. Some problems of reference-dependent evaluations are illustrated by examples from BAliBASE3.0. For SCOP40 testing datasets, we found about 20 domain pairs for which all sequence-based programs failed to align a single position correctly according to DaliLite reference alignment. Manual inspection of these instances suggests that most of these domains involve structural repeats. Many alignments produced by the tested programs did produce structurally meaningful results, but they had zero scores because they matched different repeats than those aligned by DaliLite.

Reference-independent evaluation can avoid these drawbacks. Although reference-independent evaluation is routinely used for assessing structure prediction models (50,51), it has not been frequently used in assessment of multiple sequence alignment programs (10,45). Our results show that reference-independent evaluation using various structural similarity scores produces similar, if not identical, rankings of the programs compared to reference-dependent evaluations using DaliLite alignments (Table 3). On the one hand, this result suggests that a large collection of reference alignments generated by structure alignment program DaliLite are on average well-suited for assessment of alignment quality. On the other hand, it is probably unnecessary to have a set of 'gold standard' reference alignments. Reference-independent evaluation through scoring superpositions generated according sequence-based alignments is an easier and maybe more fair method to assess alignment quality and to compare different aligners. Therefore, particularly when the size of a testing dataset is large, improving reference alignment quality either by manual inspection and adjustment (30), or by using the consensus of structural aligners (10,33) is probably not necessary. With reference-independent evaluation techniques, we even suggest bypassing the step of generating reference alignments.

Multiple sequence alignment: still an unsolved problem

Despite several percentage points' improvement in alignments provided by MUMMALS, alignment quality for divergent sequences (sequence identity <15%) is still not high, since only ~50% or even less residue pairs on average are aligned correctly according to reference alignments (Table 2). Although the best programs (MUMMALS, MAFFT and ProbCons) have limited differences in terms of average alignment quality score, they can produce quite different alignments for some sequence pairs (Table 4). It can be a good practice to use several aligners to generate alignments for divergent sequences. Manual examination of these alignments can be helpful in making expert judgments. Exploration of the consensus among programs is fruitful as well and is frequently used in meta-servers for structure prediction (52–54). Results of the meta-aligner M-COFFEE (46) as well as ours (data not shown, MUMMALS package contains a meta-program similar to M-COFFEE) suggest that combining different multiple sequence aligners can give significant but limited improvement of alignment quality.

One advantage of MUMMALS is the use of HMMs with local structure information learned from structural alignments. Our model HMM_1_3_1 gives low secondary structure prediction accuracy. Providing real secondary structure information to this model can further improve alignment quality by a few percent. These results suggest that application of more accurate secondary structure predictions can lead to better alignment quality. Exploration of database sequence homologs and available 3D structures was also shown to be helpful (11,55,56). Further enhancement of multiple alignment quality could be achieved by effective integration of existing alignment techniques and various evolutionary and structural information resources.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

The authors would like to thank Bong-Hyun Kim for the program to calculate reference-independent sequence and structural scores and Dr Kazutaka Katoh for providing the HOMSTRAD datasets. The authors would also like to thank Lisa Kinch, James Wrabl and anonymous referees for helpful comments. Funding to pay the Open Access publication charges for this article was provided by Howard Hughes Medical Institute. This work was supported in part by NIH grant GM67165 to NVG.

Conflict of interest statement. None declared.

REFERENCES

1. Altschul,S.F., Madden,T.L., Schaffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*., **25**, 3389–3402.
2. Eddy,S.R. (1998) Profile hidden Markov models. *Bioinformatics*, **14**, 755–763.
3. Lichtarge,O., Bourne,H.R. and Cohen,F.E. (1996) An evolutionary trace method defines binding surfaces common to protein families. *J. Mol. Biol*., **257**, 342–358.
4. Jones,S. and Thornton,J.M. (2004) Searching for functional sites in protein structures. *Curr. Opin. Chem. Biol*., **8**, 3–7.
5. Wallace,I.M., Blackshields,G. and Higgins,D.G. (2005) Multiple sequence alignments. *Curr. Opin. Struct. Biol*., **15**, 261–266.
6. Edgar,R.C. and Batzoglou,S. (2006) Multiple sequence alignment. *Curr. Opin. Struct. Biol*., **16**, 368–373.
7. 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.
8. Thompson,J.D., Plewniak,F. and Poch,O. (1999) A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Res*., **27**, 2682–2690.
9. 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.
10. Edgar,R.C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res*., **32**, 1792–1797.
11. 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.
12. 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.

13. 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.
14. Smith,T.F. and Waterman,M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol.*, **147**, 195–197.
15. Dayhoff,M.O., Schwartz,R.M. and Orcutt,B.C. (1978) A model of Evolutionary Change in Proteins. In Dayhoff,M.O. (ed.), Atlas of Protein Sequences and Structures. *National Biomedical Research Foundation*. Washington, DC., Vol. 5, pp. 345–352.
16. Henikoff,S. and Henikoff,J.G. (1992) Amino acid substitution matrices from protein blocks. *Proc. Natl Acad. Sci. USA*, **89**, 10915–10919.
17. Prlic,A., Domingues,F.S. and Sippl,M.J. (2000) Structure-derived substitution matrices for alignment of distantly related sequences [In Process Citation]. *Protein Eng.*, **13**, 545–550.
18. Blake,J.D. and Cohen,F.E. (2001) Pairwise sequence alignment below the twilight zone. *J. Mol. Biol.*, **307**, 721–735.
19. 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.
20. Huang,Y.M. and Bystroff,C. (2006) Improved pairwise alignments of proteins in the Twilight Zone using local structure predictions. *Bioinformatics*, **22**, 413–422.
21. Eddy,S.R. (1996) Hidden Markov models. *Curr. Opin. Struct. Biol.*, **6**, 361–365.
22. Durbin,R., Eddy,S.R., Krogh,A. and Mitchison,G. (1998) Pairwise alignment using HMMs. In *Biological Sequence Analysis*. Cambridge University Press, pp. 80–99.
23. Do,C.B., Mahabhashyam,M.S., Brudno,M. and Batzoglou,S. (2005) ProbCons: probabilistic consistency-based multiple sequence alignment. *Genome Res.*, **15**, 330–340.
24. Murzin,A.G., Brenner,S.E., Hubbard,T. and Chothia,C. (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, **247**, 536–540.
25. Chandonia,J.M., Hon,G., Walker,N.S., Lo Conte,L., Koehl,P., Levitt,M. and Brenner,S.E. (2004) The ASTRAL Compendium in 2004. *Nucleic Acids Res.*, **32**, D189–D192.
26. Holm,L. and Sander,C. (1996) Mapping the protein universe. *Science*, **273**, 595–603.
27. Kabsch,W. and Sander,C. (1985) Identical pentapeptides with different backbones. *Nature*, **317**, 207.
28. Hubbard,S.J., Campbell,S.F. and Thornton,J.M. (1991) Molecular recognition. Conformational analysis of limited proteolytic sites and serine proteinase protein inhibitors. *J. Mol. Biol.*, **220**, 507–530.
29. Pei,J. and Grishin,N.V. (2004) Combining evolutionary and structural information for local protein structure prediction. *Proteins*, **56**, 782–794.
30. Bahr,A., Thompson,J.D., Thierry,J.C. and Poch,O. (2001) BAliBASE (Benchmark Alignment dataBASE): enhancements for repeats, transmembrane sequences and circular permutations. *Nucleic Acids Res.*, **29**, 323–326.
31. Miyazawa,S. (1995) A reliable sequence alignment method based on probabilities of residue correspondences. *Protein Eng.*, **8**, 999–1009.
32. Pei,J., Sadreyev,R. and Grishin,N.V. (2003) PCMA: fast and accurate multiple sequence alignment based on profile consistency. *Bioinformatics*, **19**, 427–428.
33. 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.
34. de Bakker,P.I., Bateman,A., Burke,D.F., Miguel,R.N., Mizuguchi,K., Shi,J., Shirai,H. and Blundell,T.L. (2001) HOMSTRAD: adding sequence information to structure-based alignments of homologous protein families. *Bioinformatics*, **17**, 748–749.
35. 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.
36. Holm,L. and Sander,C. (1998) Dictionary of recurrent domains in protein structures. *Proteins*, **33**, 88–96.
37. Zemla,A., Venclovas,C., Moult,J. and Fidelis,K. (1999) Processing and analysis of CASP3 protein structure predictions. *Proteins*, **3**, 22–29.
38. Zhang,Y. and Skolnick,J. (2004) Scoring function for automated assessment of protein structure template quality. *Proteins*, **57**, 702–710.
39. Rychlewski,L., Fischer,D. and Elofsson,A. (2003) LiveBench-6: large-scale automated evaluation of protein structure prediction servers. *Proteins*, **53**, 542–547.
40. Boutonnet,N.S., Rooman,M.J., Ochagavia,M.E., Richelle,J. and Wodak,S.J. (1995) Optimal protein structure alignments by multiple linkage clustering: application to distantly related proteins. *Protein Eng.*, **8**, 647–662.
41. Shindyalov,I.N. and Bourne,P.E. (1998) Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng.*, **11**, 739–747.
42. Marchler-Bauer,A., Anderson,J.B., Cherukuri,P.F., DeWeese-Scott,C., Geer,L.Y., Gwadz,M., He,S., Hurwitz,D.I., Jackson,J.D., Ke,Z. *et al.* (2005) CDD: a Conserved Domain Database for protein classification. *Nucleic Acids Res.*, **33**, D192–D196.
43. Jones,D.T. (1999) Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.*, **292**, 195–202.
44. Rost,B. (2001) Review: protein secondary structure prediction continues to rise. *J. Struct. Biol.*, **134**, 204–218.
45. 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**, i215–i221.
46. 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.
47. Zhou,H. and Zhou,Y. (2005) SPEM: improving multiple sequence alignment with sequence profiles and predicted secondary structures. *Bioinformatics*, **21**, 3615–3621.
48. Cline,M., Hughey,R. and Karplus,K. (2002) Predicting reliable regions in protein sequence alignments. *Bioinformatics*, **18**, 306–314.
49. 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.
50. Venclovas,C. and Margelevicius,M. (2005) Comparative modeling in CASP6 using consensus approach to template selection, sequence-structure alignment, and structure assessment. *Proteins*, **61**, 99–105.
51. Kinch,L.N., Wrabl,J.O., Krishna,S.S., Majumdar,I., Sadreyev,R.I., Qi,Y., Pei,J., Cheng,H. and Grishin,N.V. (2003) CASP5 assessment of fold recognition target predictions. *Proteins*, **53**, 395–409.
52. Ginalski,K., Elofsson,A., Fischer,D. and Rychlewski,L. (2003) 3D-Jury: a simple approach to improve protein structure predictions. *Bioinformatics*, **19**, 1015–1018.
53. Wallner,B. and Elofsson,A. (2005) Pcons5: combining consensus, structural evaluation and fold recognition scores. *Bioinformatics*, **21**, 4248–4254.
54. Chivian,D., Kim,D.E., Malmstrom,L., Schonbrun,J., Rohl,C.A. and Baker,D. (2005) Prediction of CASP6 structures using automated Robetta protocols. *Proteins*, **61**, 157–166.
55. 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.
56. Simossis,V.A. and Heringa,J. (2004) Integrating protein secondary structure prediction and multiple sequence alignment. *Curr. Protein Pept. Sci.*, **5**, 249–266.