Simultaneous Sequence Alignment and Tree Construction Using Hidden Markov Models

R.C. Edgar, K. Sjölander

Pacific Symposium on Biocomputing 8:180-191(2003)

SIMULTANEOUS SEQUENCE ALIGNMENT AND TREE CONSTRUCTION USING HIDDEN MARKOV MODELS

ROBERT C. EDGAR
195 Roque Moraes Drive
Mill Valley, California 94941, USA
bob@drive5.com

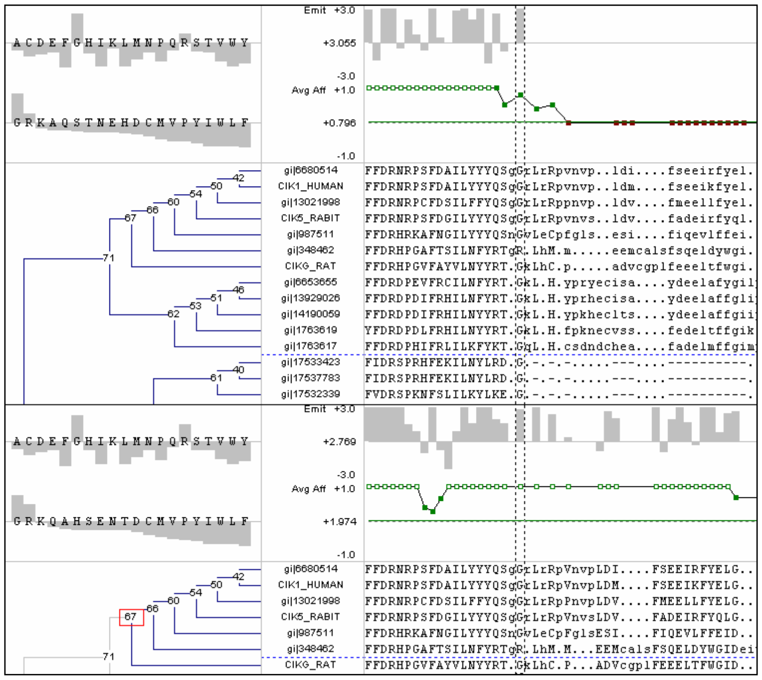
KIMMEN SJÖLANDER
Department of Bioengineering
University of California, Berkeley
California 94720, USA
kimmen@uclink.berkeley.edu

We present a new algorithm (SATCHMO) that simultaneously estimates a tree and generates a set of multiple sequence alignments given a set of protein sequences. Alignments are constructed for each node in the tree. These alignments predict the structurally conserved elements of the sequences in a subtree and are therefore of different lengths, and represent different amino acid preferences, at different nodes. Hidden Markov Models (HMMs) are also generated for each node and are used to determine branching order, to align sequences and to predict structurally alignable regions. In experiments on the BAliBASE benchmark alignment database, SATCHMO is shown to perform comparably to ClustalW and the UCSC SAM HMM software. Results using SATCHMO to identify protein domains are demonstrated on potassium channels, with implications for the mechanism by which tumor necrosis factor alpha affects potassium current.

1 Introduction

In the words of David Jones, "There are really only three things that govern the overall accuracy of comparative modeling: alignment quality, alignment quality, and…alignment quality" [1]. Comparative modeling is not the only application for which alignment quality is critical: multiple sequence alignments are used for profile construction, detection of critical residues, prediction of functional subfamilies, and a host of other tasks. Because of its central importance, the construction of multiple sequence alignments is a focus of the computational biology community. When sequences are similar to each other, virtually any alignment method will produce good results. However, evolutionary divergence in multi-gene families can result in family members with pairwise similarity so low as to be indistinguishable from chance. Even when sequence similarity is detectable, local changes in structure between members can be significant and represent a great challenge.

Methods for multiple sequence alignment can be broadly classified into two groups: Hidden Markov Model (HMM) methods [2-4], and those that optimize other scoring schemes, such as ClustalW [5]. Both approaches have essential limitations when applied to highly variable protein sequences. HMM methods tend to be successful at detecting and aligning critical motifs and conserved core structure of protein families, but may not correctly align regions between these conserved motifs. Other methods are often superior to HMMs at correctly aligning sequences within similar subgroups; however, subgroups with significant divergence may not be correctly aligned to the consensus structure, causing misalignment of family-defining conserved motifs. Here we present a new method designed to overcome these limitations by constructing a hierarchical tree and identifying conserved structural elements at each level in the tree, as illustrated in Figure 1.



A C D E F G H I K L M N P Q R S T V W Y
G R K A Q S T N E H D C M V P Y I W L F
Emit +3.0
+3.055
-3.0
Avg Aff +1.0
+0.796
-1.0
42
50
54
60
66
67
71
46
51
53
62
40
61
A C D E F G H I K L M N P Q R S T V W Y
G R K Q A H S E N T D C M V P Y I W L F
Emit +3.0
+2.769
-3.0
Avg Aff +1.0
+1.974
-1.0

```
gi|6680514   FFDRNRPSFDAILYYYQSgGrLrRpvnvp..ldi....fseeirfyel.
CIK1_HUMAN   FFDRNRPSFDAILYYYQSgGrLrRpvnvp..ldm....fseeikfyel.
gi|13021998  FFDRNRPCFDSILFFYQSgGrLrRppnvp..ldv....fmeellfyel.
CIK5_RABIT   FFDRNRPSFDGILYYYQSgGrLrRpvnvs..ldv....fadeirfyql.
gi|987511    FFDRHRKAFNGILYYYQSnGvLeCpfgls..esi....fiqevlffei.
gi|348462    FFDRHPGAFTSILNFYRTgR.LhM.m.....eemcalsfsqeldywgi.
CIKG_RAT     FFDRHPGVFAYVLNYYRT.GkLhC.p.....advcgplfeeeltfwgi.
gi|6653655   FFDRDPEVFRCILNFYRT.GkL.H.ypryecisa....ydeelafygil
gi|13929026  FFDRDPDIFRHILNFYRT.GkL.H.yprhecisa....ydeelaffgli
gi|14190059  FFDRDPDIFRHILNYYRT.GkL.H.ypkheclts....ydeelaffgii
gi|1763619   YFDRDPDLFRHILNYYRT.GkL.H.fpknecvss....fedeltffgik
gi|1763617   FFDRDPHIFRLILKFYKT.GqL.H.csdndchea....fadelmffgim
gi|17533423  FIDRSPRHFEKILNYLRD.G.-.-.-.....---...----------.
gi|17537783  FIDRSPRHFEKILNYLRD.G.-.-.-.....---...----------.
gi|17532339  FVDRSPKNFSLILKYLKE.G.-.-.-.....---...----------.
```

```
gi|6680514   FFDRNRPSFDAILYYYQSgGrLrRpVnvpLDI....FSEEIRFYELG..
CIK1_HUMAN   FFDRNRPSFDAILYYYQSgGrLrRpVnvpLDM....FSEEIKFYELG..
gi|13021998  FFDRNRPCFDSILFFYQSgGrLrRpPnvpLDV....FMEELLFYELG..
CIK5_RABIT   FFDRNRPSFDGILYYYQSgGrLrRpVnvsLDV....FADEIRFYQLG..
gi|987511    FFDRHRKAFNGILYYYQSnGvLeCpFglsESI....FIQEVLFFEID..
gi|348462    FFDRHPGAFTSILNFYRTgR.LhM.M...EEMcalsFSQELDYWGIDei
CIKG_RAT     FFDRHPGVFAYVLNYYRT.GkLhC.P...ADVcgplFEEELTFWGID..
```

Figure 1. SATCHMO graphical interface.

Here we show results using Satchmo to align two groups of sequences which share a common domain: voltage-gated potassium channels (above) and TNF-alpha induced protein B12 homologs (below). The lower-left pane in each view shows the tree, clicking on a node displays the alignment at that node. Edge lengths in this view of the tree are chosen for convenient display and are uninformative. In the upper view the root node is selected, showing a region that is aligned across all sequences. Below, an internal node is selected to show the same region aligned across K^+ channels only. The upper-right pane shows the affinity at each position (histogram) and smoothed affinity (see Section 2.5). The upper-left pane shows the affinity by residue type at the selected position, sorted alphabetically and by value. The lower-right pane shows the sequence alignment. Upper-case letters are aligned, lower-case letters are unaligned. See Section 3.5 for further discussion.

2 The SATCHMO algorithm

2.1 Algorithm

We call our algorithm SATCHMO, for Simultaneous Alignment and Tree Construction using Hidden Markov mOdels. SATCHMO is an agglomerative algorithm that uses HMMs to model classes produced in each iteration, to choose which two subtrees to join, and to generate an alignment of those two subtrees.

Input: A set of unaligned protein sequences.

Step 1: Create a node for each input sequence and construct an HMM from the sequence (Section 2.7). This step results in a set of trees, each consisting of a single node containing an HMM and a sequence.

Step 2: Measure the similarity S_{ij} of all pairs of trees (Section 2.4) and identify a pair *ab* with highest similarity. Merge this pair by adding edges from *a* and *b* to a new node *g*, forming a new binary tree rooted at *g*. Predict the conserved structural elements among the sequences rooted at *g* (Section 2.5). Create a multiple sequence alignment Aln_g and a profile HMM_g based on this prediction (Section 2.7), assign HMM_g and Aln_g to *g*.

Repeat Step 2 until: (a) all sequences are assigned to one tree, (b) the highest similarity between trees is below a user-defined threshold, or (c) no conserved elements are predicted.

$$\begin{tikzcd}
 & & & \mathbf{I}_1 \arrow[loop above] \arrow[dr] & \mathbf{I}_2 \arrow[loop above] & \cdots & \mathbf{I}_{m-1} \arrow[loop above] \arrow[dr] & & & & \\
\mathbf{S} \arrow[r] & \mathbf{N} \arrow[r] \arrow[loop above] & \mathbf{B} \arrow[r] \arrow[dr] & \mathbf{M}_1 \arrow[r] \arrow[u] \arrow[dr] & \mathbf{M}_2 \arrow[u] & \cdots & \mathbf{M}_{m-1} \arrow[r] \arrow[u] \arrow[dr] & \mathbf{M}_m \arrow[r] & \mathbf{E} \arrow[r] & \mathbf{C} \arrow[r] \arrow[loop above] & \mathbf{T} \\
 & & & \mathbf{D} \arrow[r] & \mathbf{D} & \cdots & \mathbf{D}_{m-1} \arrow[r] & \mathbf{D}_m \arrow[ur] & & &
\end{tikzcd}$$

Figure 2. HMM architecture.

Our model architecture follows the HMMer Plan 7 model, as described in Section 2.2.

2.2 HMM architecture

We chose to make our HMMs compatible with the Plan 7 architecture defined by version 2.2 of the HMMer package [6], as shown in Figure 2.

The *k*th node in the model ($k = 1 \ldots m$) has match, delete and insert states (M_k, D_k and I_k). The insert state in the last node (I_m) is disabled; terminal insert states N and C emit according to the null model, allowing N- and C-terminal extensions respectively; they emit on N→N and C→C transitions only. Note that D→I and I→D transitions are forbidden. This configuration is designed to encourage alignments that are global to the model and local to a sequence.

2.3 Aligning an alignment to an HMM

A key element of the SATCHMO algorithm is the alignment of two multiple sequence alignments to each other by aligning one (the "target") to a profile HMM constructed from the other (the "template"). This is done in such a way that columns of both alignments appear intact in the combined alignment; sequences in the target are not permitted to align individually to the profile HMM. The motivation for this constraint is the assumption that closely related sequences are already accurately aligned; hence any change that would be induced by re-aligning to a more distant profile is more likely an error than an improvement. This is implemented by allowing emitter (match and insert) states to generate multiple residues. If there are *n* sequences in the alignment, an emitter state is required to emit *n* times, producing one column. Weighting (Section 2.6) is accounted for by including transition probabilities *N* times (see Equation 6) and by calculating the emission probability for a column as $\prod_i p_i^{n_i}$, where p_i is the residue probability calculated from the template counts using Equation 7, and n_i are the weighted counts of the target column.

Columns in the input alignment that have both gaps and residues ("mixed columns") require special consideration. We allow emitter states to emit a gap (or, equivalently, not to emit) with a given probability. This allows the model to generate mixed columns without requiring sequences to take different paths. Gaps from emitter states (as opposed to gaps produced by delete states) are emitted with a background probability, so in log-odds terms contribute zero to the column score after subtraction of the null model. Columns in the input alignment which are predicted not to be alignable are assigned a zero probability of being emitted by a match state, thus forcing them into insert states.

We use the Viterbi algorithm [7] to determine the most probable path through the model. The implementation does not require an inner loop over sequences; it suffices to pre-calculate the total number of residues of each type found in each column of the alignment being scored against the HMM.

2.4 Relationship score and similarity measure

We define a relationship score r_{ij} between two trees *i* and *j* from the log-odds score per sequence of the alignment of *i* (Aln_i) to the profile HMM of tree *j* (HMM_j):

$$r_{ij} = (1/N) \log_2(P(\mathrm{Aln}_i \mid \mathrm{HMM}_j) / P(\mathrm{Aln}_i \mid \mathrm{Null})) / \mathrm{length}(\mathrm{HMM}_j). \qquad (1)$$

Here, N is the estimated number of independent observations (Equation 6). Dividing by N gives the score per sequence, dividing by the model length corrects a bias towards longer models. We also define a symmetrical similarity measure,

$$S_{ij} = (r_{ij} + r_{ji})/2. \tag{2}$$

A large positive value of S_{ij} indicates a close relationship between the sequences in i and j; zero indicates a degree of relationship indistinguishable from chance.

2.5 Prediction of structurally alignable regions

The structurally alignable regions between two trees i and j are estimated as follows. One tree is chosen to be the target and one the template by calculating r_{ij} and r_{ji} using Equation 1. If $r_{ij} \geq r_{ji}$, then j is the template, otherwise j is the target. The target is then aligned to the template as described in Section 2.3, and affinities of template nodes to target columns are calculated. The affinity $A(k)$ of the kth model node is defined to be the log-odds score of the target column $c(k)$ generated by its match state:

$$A(k) = (1/N)\, \log_2(P(c(k) \mid \mathrm{HMM}) \,/\, P(c(k) \mid \mathrm{Null})). \tag{3}$$

If the most probable path passed through its delete state, the node is assigned an affinity value of zero. Affinity is a noisy signal; we therefore define a smoothed affinity $a_w(k)$ over a window of length w, an odd integer ≥ 1:

$$a_w(k) = |\mathrm{W}(k,w)|^{-1} \sum_{q \,\in\, \mathrm{W}(k,w)} A(q). \tag{4}$$

Here, $\mathrm{W}(k,w)$ is the set of nodes found in a window of length w centered on the kth node. $|\mathrm{W}(k,w)|$ is the number of nodes in that set, $|\mathrm{W}(k,w)|=w$, $w \leq k \leq (m-w)$. Window slots beyond an end of the model are empty. For example, if w=5, then the window at the first node k=1 has only three nodes, i.e. $|\mathrm{W}(1,5)|=3$.

The kth node is predicted to be an alignable position if and only if the following condition is met:

$$a_w(k) \geq Z. \tag{5}$$

Z is described as the minimum smoothed affinity threshold; it is a parameter of the algorithm.

2.6 Sequence weighting and amino acid probability estimation

Following standard practice, we employ relative weights to compensate for correlation among the sequences, using the scheme described by Gerstein *et al*. [8].

With Bayesian methods such as ours, the number of observations may swamp the contribution of priors to posterior amino acid estimates. We estimate the number of independent observations to be:

$$N = n^{1-C}. \tag{6}$$

Here, *n* is the number of sequences in the alignment and *C* is the average over all columns of the fractional occurrence of the most common residue in that column. Hence *C* ranges from 1 (all positions identical) to 1/20 (all amino acids having equal frequency) and *N* ranges from 1 to approximately *n*. When calculating measured counts in model construction, sequence weights are scaled to *N* rather than to *n*. We introduce a maximum permitted value *U* for *N* as a second line of defense against under-weighting the priors; *U* is a parameter of the algorithm.

Estimated residue probabilities p_i, i=1...20 are calculated at each position in the alignment by combining the weighted observed residue counts n_i with a Dirichlet mixture density [9], as follows (for simplicity, we refer to the *j*th component by its hyperparameters α_j):

$$p_i = \sum_j P(\alpha_j \mid counts)\ (n_i + \alpha_{ji})\ /\ (N + |\alpha_j|). \tag{7}$$

2.7 Constructing an HMM that models the consensus structure of two alignments

When two nodes are joined to make a new tree, their alignments are aligned to each other (Section 2.3) and alignable regions are estimated (Section 2.5). A profile HMM that models the consensus structure of the combined alignment is constructed by creating a node for each position that meets the minimum smoothed affinity condition (Equation 5); i.e., for each position that is predicted to be alignable across both alignments. The nodes in the new model thus correspond to a subset of the nodes in the template profile HMM. This guarantees that model lengths are monotonically non-increasing (i.e. can get shorter but not longer) along a path from a leaf towards its root, reflecting the expectation that increasingly diverged proteins have progressively shorter mutually alignable regions. In the special case of constructing an HMM from a single sequence in Step 1 of the algorithm (see Section 2.1), a node is created for each residue.

Match state emission probability distributions are estimated by combining weighted residue counts with a Dirichlet mixture prior (Equation 7). In the preliminary implementation described here we use the HMMer default priors for transition and insert state emission probabilities; we do not attempt to estimate these parameters from observations (see Section 4 for further discussion).

2.8 Complexity

Given n sequences of length L, the space complexity of SATCHMO is dominated by the dynamic programming matrix used by the Viterbi algorithm, which is $O(L^2)$. The total time complexity is $O(L^2n^2 + Ln^3)$. On representative current hardware (a PC with one 2.5 GHz Pentium 4 processor), our implementation of SATCHMO is typically able to align 100 sequences of length 100 in under two minutes.

3 Experimental results

3.1 Reference alignments

We used version 1 of the BAliBASE benchmark alignment database [10]. BAliBASE is divided into five reference sets. Ref1 contains alignments of a small number (< 6) of equidistant sequences, meaning that the percent identity between two sequences is within a specified range. These Ref1 alignments contain sequences of similar length, with no large insertions or extensions. Alignments in Ref2 add up to three distantly related sequences (< 25% identical) from Ref1 with a family of at least 15 closely related sequences. Ref3 contains alignments of up to four subgroups, with < 25% identity between sequences from different groups. Ref4 contains alignments with long N/C-terminal extensions of up to 400 residues. Ref5 has long insertions of up to 100 residues. Ref1, 2 and 3 are divided into groups with short, medium and long sequences. Ref1 is further subdivided by percent identity.

3.2 Alignment quality scoring

BAliBASE provides a module (*BaliScore*) that defines two scores. SP is the ratio of the number of correctly aligned pairs of positions in the test (predicted) alignment to the number of aligned pairs in the reference (structurally informed) alignment. TC is the ratio of the number of correctly aligned columns in the test alignment to the number of aligned columns in the reference alignment. Both SP and TC range from 1.0 for perfect agreement to 0.0 for no agreement. The designers of BAliBASE recommend SP as the best quality score for Refs1, 2 and 3, TC as the best score for Refs4 and 5 [11]. We wrote our own module to compute SP and TC as the published *BaliScore* software module produces incorrect results on some inputs: specifically we found that *BaliScore* would report scores that were less than the correct value for alignments with gapped positions. Using the published *BaliScore* gave very similar relative rankings of the tested methods to our own scoring module, but reduced median scores.

We felt that while the SP and TC scores are useful, they have limitations as measures of alignment quality. Neither measure penalizes columns in the test alignment that are not structurally alignable, i.e. over-alignment. They thus fail to distinguish between algorithms that predict alignable regions, such as SATCHMO and SAM, from those that do not, such as ClustalW. Moreover, the "correct" alignment is sometimes ambiguous as experts may disagree on the identification of homologous positions; however SP and TC give no credit for positions with small shifts.

As a complementary measure of alignment quality, we used the Cline shift score (CSS) [12]. CSS penalizes both over- and under-alignment and gives positive, though reduced, scores for positions with small shifts. CSS is controlled by a parameter ε which determines how long a shift must be to contribute a negative value to the score. We follow Cline's recommendation and set $\varepsilon = 0.2$, which gives a negative score to shifts of more than five positions (approximately one turn in a helix). Cline defines the score on a pair-wise alignment; we take the average over all pairs of sequences present in both the test and reference alignments. CSS ranges from 1.0 in the case of perfect agreement between the test and reference alignments to $-\varepsilon$ as a lower limit.

3.3 Algorithm parameters

In addition to Dirichlet mixture priors, SATCHMO has the following parameters: U, the maximum number of independent observations, Z, the minimum smoothed affinity, and w, the window length for affinity smoothing. Two additional parameters g_1 and g_2 are defined to isolate tuning of gap-related transitions: g_1, multiplies the M$\rightarrow$D (gap-open) probability, g_2 multiplies the D$\rightarrow$D (gap-extend) probability. The transition distributions from M and D are of course re-normalized after the multiplications have been applied.

For the BAliBASE reference alignments, we found that the SP, TC and CSS scores were optimized by setting $Z = -\infty$, causing all candidate positions to be aligned irrespective of w. This parameter setting is not optimal when aligning sequences that may share only a single domain, or are otherwise more variable structurally than those in BAliBASE (as in Figure1). For such inputs, we find setting Z close to 0, and using a window size w of 5 or 7, to be optimal. There was some sensitivity to g_1 and g_2: we found that $g_1 = g_2 = 0.75$ gave a small but significant improvement over the HMMer defaults induced by $g_1 = g_2 = 1$. Very little variation was found with U, which is not surprising given that BAliBASE alignments have small numbers of sequences; we therefore set $U = \infty$. We performed some experiments where we separated the reference alignments into training sets used for parameter optimization and test sets. The parameters found to be optimal for these smaller training sets were identical to those found to be optimal for all of BAliBASE.

Table 1. Alignment quality scores for BAliBASE reference sets.
For each program and each scoring method, the median score is shown for each BAliBASE reference set. In the case of Refs 1, 2 and 3, the median score in each sub-category is also shown. Finally the median score over all BAliBASE alignments is given for each program. We show here SP results for Refs 1, 2 and 3, and TC for Refs 4 and 5, following the suggestion of the BAliBASE authors and for consistency with their published analyses.

		SP / TC			**CSS**	
	ClustalW	**SAM**	**Satchmo**	**ClustalW**	**SAM**	**Satchmo**
Ref1 short <25% identity	0.72	0.40	0.40	0.39	0.26	0.42
Ref1 medium <25% identity	0.68	0.61	0.73	0.52	0.22	0.51
Ref1 long <25% identity	0.64	0.60	0.59	0.47	0.10	0.33
Ref1short 20-40% identity	0.92	0.95	0.93	0.70	0.59	0.73
Ref1 medium 20-40% identity	0.96	0.97	0.96	0.78	0.56	0.80
Ref1 long 20-40% identity	0.96	0.96	0.95	0.77	0.52	0.75
Ref1 short >35% identity	0.99	0.99	0.98	0.90	0.94	0.89
Ref1medium >35% identity	0.98	0.99	0.99	0.93	0.90	0.92
Ref1long >35% identity	0.99	0.99	0.99	0.89	0.92	0.89
All Ref1	*0.94*	*0.97*	*0.94*	*0.77*	*0.59*	*0.77*
Ref2 short	0.88	0.00	0.76	0.68	0.81	0.67
Ref2 medium	0.86	0.89	0.78	0.71	0.85	0.73
Ref2 long	0.88	0.90	0.80	0.54	0.83	0.51
All Ref2	*0.88*	*0.77*	*0.78*	*0.66*	*0.82*	*0.69*
Ref3 short	0.72	0.00	0.79	0.65	0.58	0.66
Ref3 medium	0.74	0.76	0.66	0.75	0.62	0.77
Ref3 long	0.91	0.90	0.88	0.64	0.77	0.66
All Ref3	*0.84*	*0.76*	*0.60*	*0.71*	*0.71*	*0.74*
Ref4	0.52	0.32	0.74	0.24	0.00	0.25
Ref5	0.58	0.75	0.71	0.23	0.47	0.23
All BAliBASE	**0.88**	**0.90**	**0.88**	**0.70**	**0.67**	**0.69**

3.4 Comparison with ClustalW and SAM

We chose to compare the performance of SATCHMO with two tools, ClustalW [5] and the UCSC SAM "tuneup" algorithm based on their SAM-T99 clustering and alignment method [13]. These tools are representative of the non-probabilistic and

HMM categories of alignment methods respectively. ClustalW has been shown to have excellent performance against BAliBASE, ranking behind only PRRP [14] in a comparison of ten algorithms [11]. We used ClustalW version 1.81 with default parameters. Following the procedure recommended Karplus and Hu for evaluating tuneup performance on BAliBASE alignments [13], we assigned zero scores to the 18 reference sets where tuneup failed to produce a complete alignment of the test sequences owing to rejection of one or more sequences deemed to be too distantly related.

Median scores for the three programs on BAliBASE reference sets are shown in Table 1. In our preliminary implementation we do not handle undetermined residues; the following reference alignments that include letters X or B were therefore excluded: 1bbt3_ref1, 1havA_ref1, 1ppn_ref1, 5ptp_ref1, 9rnt_ref1 2trx_ref2, 1ajsA_ref2, 2myr_ref2, 4enl_ref2, 1ajsA_ref3, 2myr_ref3, 4enl_ref3, 1lkl_ref4 and 2abk_ref4.

3.5 Domain identification

Our preliminary experiments with SATCHMO suggest that it is effective at identifying protein domains. In Figure 1, we show the tree constructed by SATCHMO for two sets of proteins: TNF-alpha-induced protein B12 and homologs, and voltage-gated potassium channels. The surprising homology between these two groups was discovered by one of us (Sjölander) while scoring the NR database with an HMM constructed for voltage-gated potassium channels [15], where these B12 proteins received weak but significant scores. SATCHMO assigns these two groups to separate subtrees, and identifies a common domain. Our analysis shows this region to be the tetramerization, or T1, domain of potassium channels, for which a solved structure exists. This allows us to predict the fold of TNF-alpha-induced protein B12 and homologs. Even more intriguingly, tumor necrosis factor alpha is known to affect potassium current, but the precise mechanism is unknown [16, 17]. Since TNF-alpha induces the B12 protein and its homologs, and these B12 proteins share homology with the tetramerization domain, we predict that at least one of the mechanisms by which TNF-alpha affects potassium current is by inducing the B12 proteins which tetramerize with potassium channels.

4 Discussion

The test results suggest that SATCHMO performs competitively with state-of-the-art algorithms while providing more insight into common structural elements and variations of those elements among the input sequences. We find this encouraging, especially given that we regard the current implementation of SATCHMO as preliminary and expect that significant improvements are possible. We plan to investigate several areas. We will pursue a more rigorous treatment of gaps and better estimation of transition probabilities and insert state emission probabilities. We will also enable local-local alignment. As presented here, SATCHMO is a progressive algorithm, meaning that once a pair of sequences has been aligned, this part of the alignment is fixed. We may find that improvements are possible by keeping regions of high confidence fixed and re-aligning marginal regions using iterative methods when more sequences have been added. We also plan to investigate different smoothing heuristics and alternative methods for predicting alignable regions.

We also plan to investigate the performance of SATCHMO on input that is more representative of typical applications than the small number of carefully screened sequences found in the BAliBASE alignments. Adding homologs of the BAliBASE reference sequences may enable us to improve upon the scores reported hereby providing intermediates that interpolate between, and therefore guide the alignment of, those sequences.

We will further explore the use of SATCHMO in phylogenetic tree construction through experiments based on functionally characterized proteins. Other work in this area suggests that this approach can produce trees and alignments of high quality [18]. We will evaluate the trees constructed using experimental information on molecular function, binding pocket positions, and other biological data.

Finally, preliminary results (as shown in Figure 1) suggest that SATCHMO is effective at identifying domain-level similarities between proteins, and this aspect of its functionality will be examined further.

Acknowledgements

The authors thank Sean Eddy for making the HMMer package and its source code readily available, Melissa Cline for helpful discussions, and Wayne Christopher for providing a subroutine for efficient calculation of the log-gamma function.

References

1. Jones, D.T., *Progress in protein structure prediction.* Curr Opin Struct Biol, 1997. **7**(3): p. 377-87.
2. Krogh, A., M. Brown, I.S. Mian, K. Sjolander, and D. Haussler, *Hidden Markov models in computational biology. Applications to protein modeling.* J Mol Biol, 1994. **235**(5): p. 1501-31.

3. Eddy, S.R., *Hidden Markov models.* Curr Opin Struct Biol, 1996. **6**(3): p. 361-5.
4. Karplus, K., K. Sjolander, C. Barrett, M. Cline, D. Haussler, R. Hughey, L. Holm, and C. Sander, *Predicting protein structure using hidden Markov models.* Proteins, 1997. **Suppl 1**: p. 134-9.
5. Thompson, J.D., D.G. Higgins, and T.J. Gibson, *CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice.* Nucleic Acids Res, 1994. **22**(22): p. 4673-80.
6. Eddy, S.R.; *HMMER: Profile hidden Markov models for biological sequence analysis*. http://hmmer.wustl.edu/
7. Forney, G.D., *The Viterbi Algorithm.* Proc. IEEE,, 1973(61): p. 268-278.
8. Gerstein, M., E.L. Sonnhammer, and C. Chothia, *Volume changes in protein evolution.* J Mol Biol, 1994. **236**(4): p. 1067-78.
9. Sjolander, K., K. Karplus, M. Brown, R. Hughey, A. Krogh, I.S. Mian, and D. Haussler, *Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology.* Comput Appl Biosci, 1996. **12**(4): p. 327-45.
10. Thompson, J.D., F. Plewniak, and O. Poch, *BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs.* Bioinformatics, 1999. **15**(1): p. 87-8.
11. Thompson, J.D., F. Plewniak, and O. Poch, *A comprehensive comparison of multiple sequence alignment programs.* Nucleic Acids Res, 1999. **27**(13): p. 2682-90.
12. Cline, M., R. Hughey, and K. Karplus, *Predicting reliable regions in protein sequence alignments.* Bioinformatics, 2002. **18**(2): p. 306-14.
13. Karplus, K. and B. Hu, *Evaluation of protein multiple alignments by SAM-T99 using the BAliBASE multiple alignment test set.* Bioinformatics, 2001. **17**(8): p. 713-20.
14. Gotoh, O., *A weighting system and algorithm for aligning many phylogenetically related sequences.* Comput Appl Biosci, 1995. **11**(5): p. 543-51.
15. Sjolander, K. *Automated domain identification in proteins using HMMs*. Presented at Genome Sequencing and Analysis Conference, Miami, FL; September, 1999
16. Soliven, B., S. Szuchet, and D.J. Nelson, *Tumor necrosis factor inhibits K+ current expression in cultured oligodendrocytes.* J Membr Biol, 1991. **124**(2): p. 127-37.
17. McLarnon, J.G., M. Michikawa, and S.U. Kim, *Effects of tumor necrosis factor on inward potassium current and cell morphology in cultured human oligodendrocytes.* Glia, 1993. **9**(2): p. 120-6.
18. Mitchison, G.J., *A probabilistic treatment of phylogeny and sequence alignment.* J Mol Evol, 1999. **49**(1): p. 11-22.