Fold Recognition by Remote Homology Detection Using Hidden Markov Models

Kevin Karplus, Christian Barrett, Richard Hughey[*]

UCSC-CRL-98-06
2 February 1998

Computer Engineering, University of California, Santa Cruz, CA 95064, USA
Tel: (408) 459-4250, Fax: (408) 459-4829, email: karplus@cse.ucsc.edu

ABSTRACT

A new HMM-based method (target98) for finding remote homologs of protein sequences is evaluated using four test sets. Three of the test sets are fold-recognition tests, where the correct answers are determined by structural similarity. The fourth uses a curated database. The method is compared against WU-BLASTP and DOUBLE-BLAST, a two-step method similar to ISS, but using BLAST instead of FASTA. The HMM-based method had the fewest errors in all tests—dramatically so for the fold-recognition tests.

One key to the performance of the HMM method is a new score-normalization technique which compares the score to the score of the reversed sequence, rather than to a null model.

[*]This work was supported in part by NSF grant DBI-9408579, DOE grant DE-FG0395ER62112, and a grant from Digital Equipment Corporation.

0.1 Introduction

A critical issue confronting the genome sequencing projects today, and biology in general, is functional and structural characterization of new proteins. This characterization is often inferred by similarity to proteins of known structure or function. Mutation often makes these similarities difficult to detect in distantly related proteins. Finding these evolutionary connections is called *remote-homolog detection.* Methods that are able to detect more subtle similarities between sequences are thus able to assign putative structure and functional characterization to more new proteins.

The focus of this paper is to present results using hidden Markov model methods to detect remote homologies. We compare our results to those using more established methods. The results are presented in the context of four tests, three of which are fold-recognition tests. These three tests use a set of *target* sequences whose folds are to be determined, a *fold* database of sequences of known structure, and a definition of correct target-fold pairings. The fourth uses a curated database whose protein sequences were grouped according to family. For all the tests, we used only primary sequence information—the test was purely one of detecting remote-homologs, not of protein structure prediction or threading. These test sets are described in Section 0.2.

For the fold-recognition tests, our HMM-based methods did extremely well at all levels of acceptable error, finding many more remote homologs than the more traditional sequence-based methods. The results on the fourth test were mixed. At the minimum-error point our method had the fewest errors. If only a few false positives can be tolerated, WU-BLASTP did better. The detailed results are presented in Section 0.3, the methods themselves in Section 0.4, and the reasons for the difference in performance are discussed in Section 0.5.

0.2 Test Sets

The first three test sets described are databases of sequences of known structure that provide a measure of relatedness between the structures. In each case, we had a set of target sequences whose fold we wanted to determine by matching it against all the sequences in the fold database. We evaluated how method described in Section 0.4 discriminated between the homologous and nonhomologous sequences in the database for all of the target sequences.

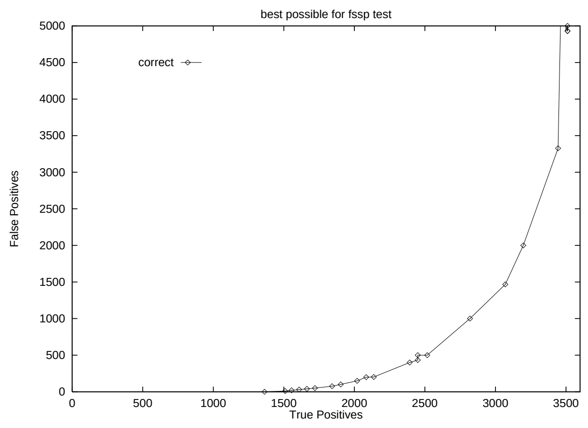
0.2.1 FSSP

The FSSP test set is based on July 1997 FSSP protein classification tree [9, 10]. Our fold database contains the sequences of all 1050 leaves of the FSSP tree, and our target list is a subset of 166 sequences chosen arbitrarily to cover all major subtrees. The use of the FSSP tree ensures that no two sequences in the database have more than 25% identical residues in the correct structural alignment.

The FSSP data set uses DALI structure comparison [8] to determine structural homology. A soft classification was made, in which DALI z-scores higher than 6 were considered to be homologs, z-scores lower than 2 were non-homologs, and z-scores between 2 and 6 were counted as partly homologous and partly nonhomologous. For the 174,134 pairs, the sum of the homology counts was 3510.85 (about 2%), though even a perfect classifier makes at least 1494.95 errors (Figure 0.1).

0.2.2 SCOP

We used two test sets [6, 16] derived from the Structural Classification of Proteins (SCOP) hierarchy [11]. For each test set, we used identical lists for both the the target list and the database of known folds. A pair was labeled as a homolog pair if both sequences were in the same SCOP superfamily, else it was labeled a nonhomolog pair. No two sequences in either test set had more than 40% sequence similarity.



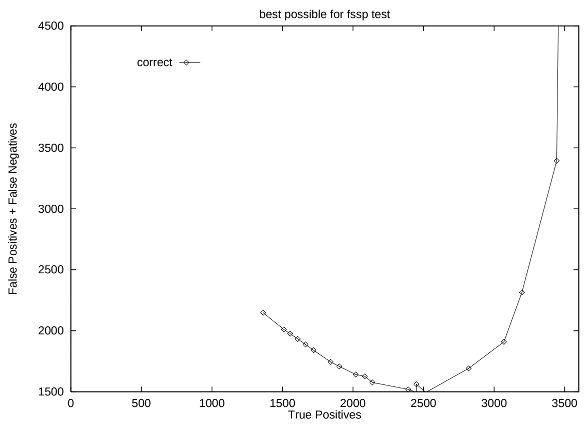


Figure 0.1: The best possible number of false positives (left) and the errors as a function of the number of true positives for the soft-thresholding done in the FSSP test.

The *whole-chain* test set was composed of 571 single-domain proteins. Of the 147,345 pairings, only 931(0.6%) are considered correct homologies. The *domain* test set contained the whole-chain test set, plus another 364 domains that were only parts of chains (935 sequences in total). Of the 436,645 possible non-self pairs, only 2605 were considered homologs (0.6%).

0.2.3 Pearson

The fourth test is Pearson's test for sequence-comparison tools [17]. It is a curated version of the PIR database (PIR1, release39) [4] that Pearson augmented with the addition of 237 sequences. In total, it comprises 12,216 sequences.[1] We report results here for the "e0" set of 67 target sequences chosen by Pearson. Of the 818,405 possible non-self pairs of the target sequence with database sequences, 3474 (0.4%) were considered correct. The Pearson test set is of a different character than the three fold-recognition tests. Since Pearson's families are generally of fairly close homologs, this is more a test of close-homolog classification.

0.3 Results

Common to each of the test sets was a list of target (query) sequences and a database that contained homologs for each of these target sequences. A perfect homolog search method would cleanly separate the homologs in the database from the non-homologs. For any given threshold, we can identify the true positives (homolog pairs scoring better than the threshold), the false positives (non-homolog pairs scoring better than the threshold), and the false negatives (homolog pairs scoring worse than the threshold). An *error* is either a false positive or a false negative.

To evaluate the performance of the homolog search methods described in Section 0.4 for each test set, all pairs of target sequence and database sequence were sorted from best score to worst score. By sweeping through this sorted list, we compare the methods in three fashions. First, to make comparisons based on one number, we compare the number of errors at each method's minimum error point. See Table 0.1. Next, we plot the number of non-homolog pairs found for number of homolog pairs found. That is, the false positives as a function of true positives. In all plots, we see a fairly gentle initial slope, with a few false positives, transitioning to a steep slope at some point. The minimum-error point is generally at this transition. Finally, we plot the total number of errors as a function of true positives. This provides a more detailed look at the tradeoff between precision (minimizing false positives) and recall (minimizing false negatives).

[1] There are 12,219 sequences in the database, but three of them are duplicates of others.

method	FSSP	SCOP whole chain	SCOP domain	Pearson
optimum, true positive	2449.45	931	2605	3474
optimum, false positive	433.55	0	0	0
optimum, errors	1494.95	0	0	0
WU-BLASTP, true positives	173.75	212	353	2948
WU-BLASTP, false positives	26.25	21	24	195
WU-BLASTP, errors	3363.35	740	2276	721
double-blast, true positives	279.30	288	533	3072
double-blast, false positives	50.00	22	71	352
double-blast, errors	3281.55	665	2143	754
target97, true positives	497.60	400	–	–
target97, false positives	123.40	41	–	–
target97, errors	3136.65	572	–	–
target98, true positives	459.68	397	880	3296
target98, false positives	81.33	21	68	406
target98, errors	3132.50	555	1793	584

Table 0.1: Table of minimum-error points for the different test sets and different methods. Each column is one of the four test sets. The target97 method was used only on the FSSP and whole-chain test sets.

Note that we use a single threshold for each method for all of the targets in a test set, not a separate threshold for each target as done previously for the Pearson test set [1]. Using seperate thresholds would provide much more impressive numbers, but the single-threshold is a more valuable test. We are not testing how well a particular library of models can be tuned, but how well a set of homologs can be found for a protein of unknown character. If we do not already know the classification, we cannot choose a classification-specific threshold, hence the insistence on a single threshold.

Because of changes in the SAM software package between creation of the target97 and target98 models, the target97 method cannot be fairly applied to new datasets. Only the FSSP and SCOP whole-chain datasets had the target97 alignments built before the changes in SAM, so the SCOP domains and Pearson test sets do not have target97 results.

0.3.1 FSSP

As mentioned in Section 0.2.1, structure pairs in this test set whose DALI z-score was between 2 and 6 were not judged to be completely homologous or non-homologous. They were considered fractionally homologous, with the fraction depending linearly upon the actual z-score. Thus, we have fractional values for true and false positives and errors.

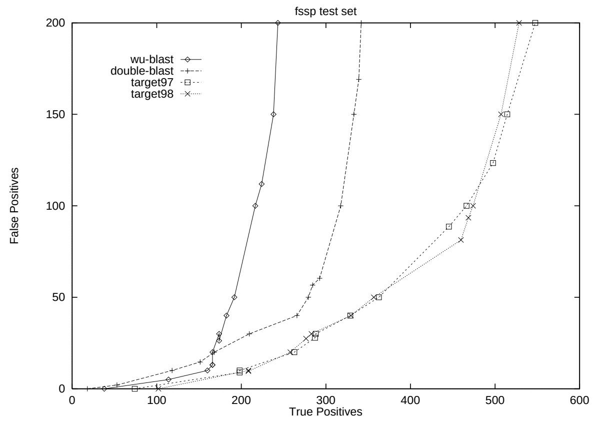
Figure 0.2 shows false positives and errors as functions of true positives. Both curves show the HMM-based methods doing much better than the BLAST-based methods. At the minimum-error point, target98 does slightly better than target97, but target97 does not degrade quite as fast if the threshold is too loosely set.

0.3.2 SCOP

0.3.2.1 Whole-chain test set

For the whole-chain SCOP dataset, the HMM-based methods perform best for all levels of false positives. If no false positives are allowed, WU-BLASTP gets 148 true positives, DOUBLE-BLAST gets 233, target97 gets 267, and target98 gets 256. The minimum-error points are even more dramatically separated with only 555 errors for target98 vs. 665 for DOUBLE-BLAST (see Table 0.1).

Figure 0.3 shows the tradeoff as the threshold is changed. Target98 does much better when few false positives are allowed, but when more than 100 are allowed, target97 finds more homologs.



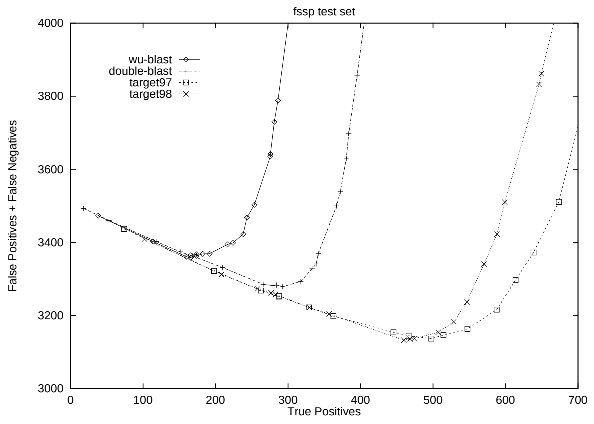
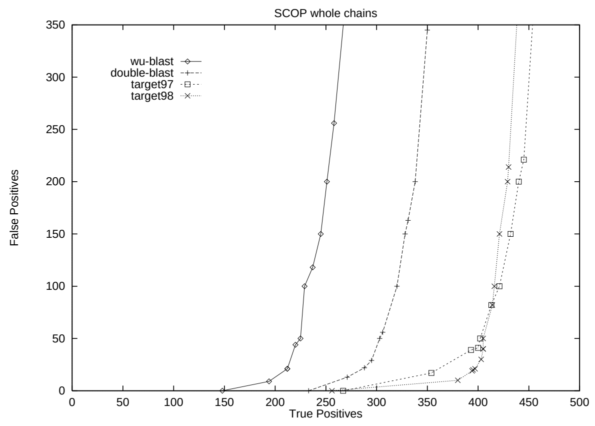


Figure 0.2: Comparison of four methods for the FSSP test set. For the theoretically best possible performance, see Figure 0.1. If the thresholds are set loose enough (large numbers of true and false positives) the target97 method slightly outperforms the target98 method.



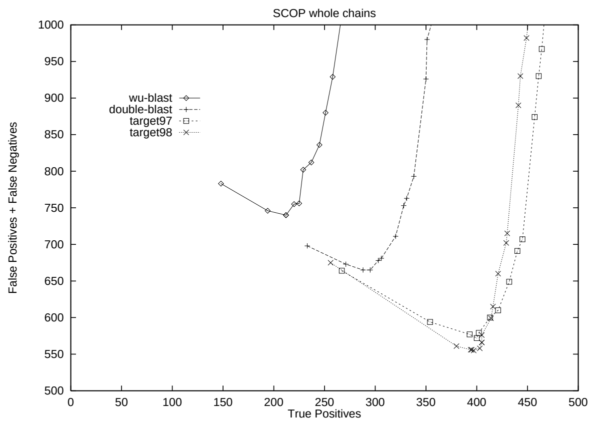


Figure 0.3: Results for four methods on the SCOP whole-chain test of 571 sequences. The maximum possible number of true positives is 931.

0.3.2.2 Domain test set

On the domains test, if the threshold is set to exclude all false positives, WU-BLASTP does best with 268 true positives, while DOUBLE-BLAST gets only 14, and target98 gets 101. The good performance of WU-BLASTP does not extend far, as target98 beats WU-BLASTP if only 10 false positives are allowed. This test set probably provides the most dramatic improvement of the HMM-based methods over the BLAST-based ones. This is particularly evident in Figure 0.4.

0.3.3 Pearson

The Pearson test set differs from the others in that the database sequences generally do not have known structure. The hand-classification of the sequences into families relies heavily on sequence similarity, resulting in families composed of generally close homologs.

The closeness of the members of the families can be seen in the excellent performance of WU-BLASTP on this dataset. With no false positives, WU-BLASTP gets 547 true positives, DOUBLE-BLAST gets 603 true positives, and target98 gets only 350. At 200 false positives (near WU-BLASTP's minimum-error point), WU-BLASTP gets 2952 true positives, DOUBLE-BLAST gets 2760, and target98 gets 2584. At 400 false

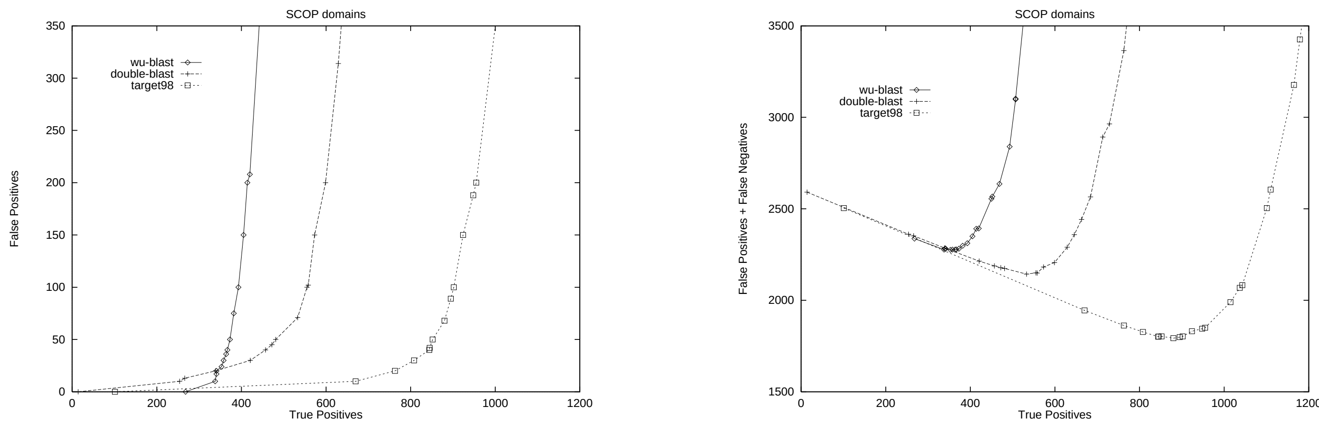


Figure 0.4: Results for three methods on the SCOP domains test of 935 sequences. The maximum possible number of true positives is 2605. The target97 method was not applied to this test set.

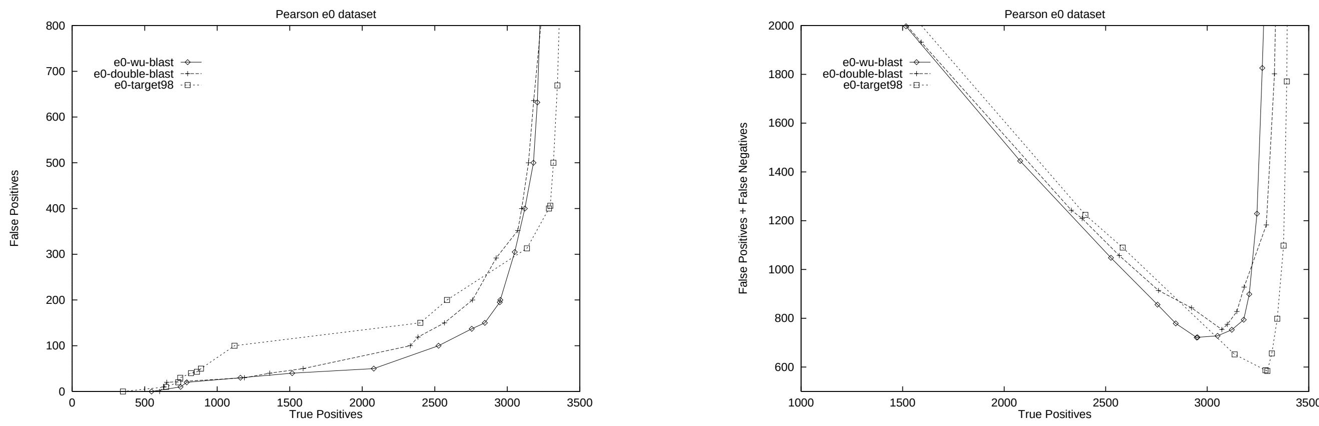


Figure 0.5: False positives versus true positives for the Pearson dataset. The maximum number of possible true positives is 3474. Note that WU-BLASTP does best for close homologs, and target98 does better for more remote ones.

positives (near target98's minimum-error point), WU-BLASTP gets 3121 true positives, DOUBLE-BLAST gets 3099, and target98 gets 3287. Figure 0.5 shows this tradeoff in performance for the methods clearly.

Due to database size, we did not build 12,216 HMMs—one for each of the sequences in the database. Instead, we built just 67 HMMs, one for each target sequence. Thus, for this test set, only part of the complete target98 method described in Section 0.4.2 was used.

0.4 Remote Homology Search Methods

Four methods were evaluated in Section 0.3: two based on WU-BLASTP and two based on HMM methods using the SAM software [14, 12].

0.4.1 Blast-based methods

Two of the the methods used here are based on the BLAST search program [3], perhaps the most widely used bioinformatics tool today. This program is extremely fast and easy to use, so evaluating it is essential.

0.4.1.1 WU-Blast

The simplest approach to remote-homolog detection is to provide the target sequence to a version of BLAST, and collect the top hits in the database. The version of BLASTP that we used for these tests is `BLASTP 2.0a16MP-WashU`. In order to allow us to sweep the threshold over a wide range, we set the `E` parameter, (the expected number of false positives) to 10 for each search. We recorded the logarithm of the reported P-value as the score to threshold. The exact setting of E is probably unimportant, as the optimum threshold never corresponded to a P-value greater than 0.005.

0.4.1.2 Double-Blast

The DOUBLE-BLAST method was inspired by ISS [16]. No direct comparison with ISS is included here, but comparisons are being done on the two SCOP datasets, and DOUBLE-BLAST appears to be similar to ISS in the effectiveness of its searches.

Instead of trying to find the database sequences directly from the target sequence in one step, a two-step approach is used. First, a set of close homologs to the target sequence is found in a large database of sequences, then each homolog is used as a query to search the final database. WU-BLASTP is used in both steps for finding the set of close homologs and for doing the searches using them. The homologs are chosen from a non-redundant protein database (NRP [15]). The first search is done with an E-value of 0.00005, and the second search with an E-value of 0.2. The score reported is the maximum of the reported E-values for the hits. Each hit found in the first search is treated as a separate homolog, as attempts to combine the hits resulted in many more false positives, particularly on the SCOP domain test set, which has many non-homologous domains that are adjacent in the sequences they are derived from.

0.4.2 Hidden Markov Model Methods

The target97 and target98 methods are very similar—indeed the target98 method is only a minor improvement to the target97 method. Both methods attempt to find and multiply align a set of homologs to a given sequence, then create an HMM from that multiple alignment. For the fold-recognition tests, we created multiple alignments for all the sequences in the fold database (1050 for FSSP and 931 for SCOP, 1677 in all, taking the overlap into account).

There are two ways to score a target sequence against the fold database: we can build an HMM for the target sequence and score all the database sequences using the model, or we can score the target sequence against all the database models. For the fold-recognition tests, we did both forms of scoring and added the resulting scores together. For the Pearson test, since we were unwilling to build 12,216 multiple alignments, we used only the first method, scoring all the sequences against the 67 target HMMs.

We used the same method for building HMMs from multiple alignments and for scoring the HMMs for both target97 and target98—the only differences were in how the multiple alignments were built from single sequences. Section 0.4.2.1 describes how the HMMs were built and scored, Section 0.4.2.2 describes how sequence weighting is done, and Sections 0.4.2.3 and 0.4.2.4 describe how the multiple alignments were built.

0.4.2.1 Building and scoring HMMs

Both target97 and target98 use the same methods for building and scoring HMMs. First, a set of sequence weights is determined from the alignment (Section 0.4.2.2). Next, `modelfromalign` is used to build the model from the alignment and the sequence weights. Finally, `hmmscore` performs a local, all-paths scoring of the sequences, using a new reversed-sequence normalization feature. The rest of this section explains the new feature and the reasons it was introduced.

For many protein families, their multiple alignment shows columns of strict conservation of what are usually the more rarely seen residues (cysteine, for example). When scoring databases with an HMM built for these families, sequences that are compositionally biased towards these residues tend to receive inflated scores and become false positives. We noticed, for example, that metallothionein, with 20 cysteines out of 61 residues, scored extremely well on any sequence that had a cluster of closely spaced cysteines. Indeed the metallothioneins accounted about for half of the first 40 false positives for early versions of target97 in the SCOP whole-chain test.

We found and implemented a workaround that eliminated this source of false positives—instead of comparing the log-probability of a sequence for an HMM with its log-probability in a null model (the NLL-minus-null score we have previously suggested [5]), we looked at the difference of the log-probability of the sequence and the log-probability of the reverse of the sequence. Since the reversed sequence has the same length and composition as the sequence, these two sources of error are effectively eliminated.

0.4.2.2 Weighting sequences for HMMs

The target97 and target98 methods both use sequence weighting for building models from alignments, both internally and when the final alignments are used to create the models for scoring a set of sequences, but not for retraining with `buildmodel`.

Target98 always uses the same weighting method. The relative weights are set with Henikoffs' position-based sequence weights [7], but the absolute weight is set to get a specific level of entropy averaged over all columns after a Dirichlet mixture regularizer [18] is applied to the weighted counts. The entropy is specified by the number of bits saved relative to the entropy of the background distribution. This relative entropy measure has been used previously to characterize substitution matrices [2], and the popular BLOSUM62 matrix corresponds to saving about 0.7 bits per column The savings for the varh50 method is not fixed but varies from 2.5 bits for alignments with only 20 match columns down to about 0.36 bits per column for alignments with over 600 match columns. More precisely, the savings requested is $50/\min(n, 140(1-e^{-0.008n}))$, where n is the length of the alignment.

The large savings requested for very short alignments is generally not available with any reasonable weights, and the relatively poor performance of the target98 method on short peptides (noticeable when analyzing the top false positives for the SCOP domains test set) may be due to this weighting problem.

The target97 method uses two other weighting schemes internally in addition to varh50: w0.5 and w1.0. Both attempt to set the relative weights and the absolute weight at the same time, using the encoding cost of each sequence raised to some power to set the relative weight. Constructing the target97 alignments uncovered the problems with this weighting method, and so its use has been discontinued.

0.4.2.3 Target97

Target97 starts with a sequence (or a seed alignment) and searches the non-redundant protein database using WU-BLASTP with a very loose threshold (approximately E=5000) to get a set of possible homologs. It then selects from this set to build a series of multiple alignments, starting with fairly close homologs and gradually increasing the remoteness it will consider.

The initial WU-BLASTP cull of NRP is necessary for two reasons: we don't expect an HMM built from a single sequence to do well at finding homologs, and an HMM database search of all of NRP is too slow.

Aside from WU-BLASTP in the initial stage, a prerelease of the SAM suite version 2.0 [14, 12] was used throughout—but not with default parameters. Of particular note are the use of `buildmodel` (SAM's expectation-maximzation parameter estimation module) and `multdomain` (which searches for multiple matches to an HMM in a sequence).

All `buildmodel` procedures used an initial model derived from a previous stage of the process—at no time was the default random initial model used. A Dirichlet mixture was used as a Bayesian prior to get estimates of amino acid probabilities, rather than the default pseudocount regularizer. Additionally, the initial_noise and anneal_noise parameters were set to 0.5 and 0.01, respectively, to reduce the effect of noise.

The multdomain program does a local alignment of the model and sequences, reporting high-scoring subsequence-submodel matches as long as they were at least 10% of the length of the model.

After getting a relatively small set of possible homologs, the next step used modelfromalign with a Dirichlet mixture prior to estimate an initial model (m0) from the target sequence. A transition regularizer that readily allowed a sequence to make a gapped alignment to the model was used so that sequences could more readily align on the conserved regions. A multdomain search placed all sequences that scored better (more negative) than -20 nats into the training set for the next round of HMM estimation.

Using m0 as an initial model and the same "cheap-gap" transition regularizer, a new model (m1) was estimated using buildmodel and the new set of training sequences. The effect of the transition regularizer was strengthened by a factor of 10 for this procedure to ensure that gaps remained cheap. A second multdomain search of the potential homolog set was performed with this model, placing any sequences that scored better than a now more relaxed cutoff of -15 nats into the training set for the next round of model estimation.

This next round of estimation for m2 was similar to that for m1, except that the m1 model was retrained using the larger training set and a different transition regularizer. The previous "cheap-gap" regularizer effectively allowed the training sequences to align in columns such that residue similarity was maximized. This often results in alignments with many gaps and short runs of matched residues, which are usually not meaningful. To correct this, a "long-match" regularizer was introduced to constrain the alignments into long stretches that left the conserved blocks intact, but used fewer insertions and deletions in the variable regions.

Using the alignment of the training sequences for m2, a weight was assigned to each sequence (see Section 0.4.2.2) so that the average entropy of the match states was 1.0 bit less than the entropy of the background frequency distribution. From these weighted sequences a generalized model (m3) was estimated and used to search the potential homolog set once again, now using a cutoff of -9 nats to reflect the fact that we are now searching for more remote homologs. The sequences scoring better than -9 nats were used as the training set for m4.

A new model (m4) was estimated with a transition regularizer derived from structural alignments in the FSSP database. Since this database is composed of structural alignments that sometimes have quite low sequence identity, we expect it to provide a moderately accurate view of what gaps to expect in alignments among distantly related sequences.

Using m4 and its training sequences, a multiple alignment was once again created and its individual sequences weighted. This time, though, the weights were set so that the average entropy of the match states was 0.5 bits less than the entropy of the background frequency distribution, giving greater generalization than the m3 model. One last application of buildmodel using these sequences and their weights resulted in m5, which was used to search the potential homolog set one last time. All sequences that scored better than -9 nats were aligned using m5 for the final multiple alignment.

Since this process is involved and requires substantial computing time, it is only done once for any sequence and the final alignment kept as an entry in a library. Further use of the alignment is described in Section 0.4.2.1.

Because of changes in the way hmmscore scores local alignments in the latest release of SAM [19], local alignment scores have changed by approximately 6–7 nats since the target97 script was written. This change in scoring method, although generally an improvement, means that it is no longer possible to duplicate the target97 alignments exactly. Because the change in hmmscore came before all the multiple alignments had been built, only two of the test sets could be tested with the target97 method.

0.4.2.4 Target98

Target98 was created partly in response to the change in `hmmscore` and partly as a result of analyzing the worst errors of the target97 method on the SCOP whole-chain data set.

The initial step in the target97 method used WU-BLASTP to select the potential homologs from the non-redundant database. Target98 modified this search to produce two sets of homologs: one of very close homologs ($E \approx 0.00003$) and one of possible homologs ($E \approx 500$).

The target98 method then uses 4 iterations of a selection, training, and alignment procedure. For each iteration it needs an initial alignment, a set of sequences to search, a threshold value, and a transition regularizer.

On the first iteration the single sequence (or seed alignment) passed to the method is used as the initial alignment and the close homologs found by WU-BLASTP are used as the search set. The threshold is set very strictly (-40 nats), so that only really good matches to the sequence are considered. The transition regularizer is one that was set up to try to match the gap costs used by WU-BLASTP. Requiring both WU-BLASTP and the initial HMM to score a sequence well ensures that only very close homologs are included at this stage of the process, avoiding some of the early contamination of the multiple alignments that lowered the performance of target97.

On subsequent iterations the input alignment is the output from the previous iteration and the search set is the larger set of possible homologs found by WU-BLASTP. The thresholds are gradually loosened (-30 nats, -24 nats, -16 nats). The final threhshold, though it looks much tighter than the target97 threshold (-9 nats), is actually quite comparable, given the change in the behaviour of `hmmscore`.

For the second and third iteration, the "long-match" transition regularizer is used, and for the final iteration the transition regularizer trained on FSSP structural alignments is used.

The selection, training, and alignment procedure consists of several calls to SAM programs. First, a model is built from the initial alignment using the same varh50 script that is used in the final scoring of the target97 and target98 methods (see Section 0.4.2.1). Sequences are selected using `hmmscore` if they score better than the provided threshold and if they scored (threshold+7) nats better than the reverse of the sequence.

Only the relevant parts of the selected sequences are kept for training, by using `multdomain` to select from the sequences. To ensure that the initial input to the whole process is not lost, it is added to the training set at this point, and any duplicate sequences in the training set are eliminated.

Next, `buildmodel` is called to retrain the initial model on the new training set. The anneal_length parameter is set to 5 to cut of the noise sooner, allowing the convergence test to work properly in `buildmodel`. The retrained model is used (via multdomain) to select and align sequences from the set selected by the first `hmmscore` run of the iteration. This alignment is what is returned from the iteration.

0.5 Discussion

Although the percentage of correct homologs is smaller in the SCOP-based tests than the FSSP-based tests, the sequences considered as matching are somewhat more similar, so the test is somewhat easier to do well on. Because of the very narrow families in the Pearson test set, finding most of the homologs is extremely easy.

At the minimum-error points, the best method was always target98. If we compute the error rate as the number of errors divided by the number of possible true positives, we get an 89.2% error rate for the FSSP test set, 59.6% for the SCOP whole-chain test set, 68.8% for the SCOP domains, and 16.8% for the Pearson test set. We can see that even the best current remote-homology methods find only a small fraction of the evolutionary relationships available, and that the popular Pearson test set is really only suitable for testing close-homology methods, not for evaluating remote-homology detection.

The biggest innovation in this paper is the reversed-sequence scoring. Not only does this method correct for length and composition biases, but some other, subtler effects are also cancelled—for example the periodic hydrophobicity patterns of ampipathic helices or beta strands also appear in the reversed sequence, as does the lower frequency surface-core hydrophobicity pattern. Because of these subtle effects, the reversed sequence is a much more realistic decoy than a scrambled sequence. For example, in the scoring we did for the CASP2 contest [13], we had to eliminate by hand some coiled-coil models that scored any helical protein well—the reversed-sequence decoy eliminates these problems as well.

Thresholds need to be about 7 nats looser when using reversed sequences rather than null models, to get comparable numbers of sequences found. We do not have a good Bayesian interpretation of the reversed sequence scoring, nor do we have any theoretical justification for the fairly consistent 7 nat shift in the scores.

Part of the target98 method has been installed on the Web as

`http://www.cse.ucsc.edu/research/compbio/build-target98.html`

Acknowledgements

We thank Nguyet Manh, who ran many of the early tests of the target97 method on the FSSP test set, and Cyrus Chothia, who provided us with the selected sequences that make up the SCOP test sets.

References

[1] P. Agarwal and D. J. States, "Comparative accuracy of methods for protein-sequence similarity search," *Bioinformatics*, vol. 14, no. 1, pp. 40–47, 1998.

[2] S. F. Altschul, "Amino acid substitution matrices from an information theoretic perspective," *JMB*, vol. 219, pp. 555–565, 1991.

[3] S. F. Altshul, W. Gish, W. Miller, M. E. W., and L. D. J., "Basic local alignment search tool," *JMB*, vol. 215, pp. 403–410, 1990.

[4] W. Barker, D. George, and L. Hunt, "Protein sequence database," *Methods Enzymol.*, vol. 183, pp. 31–49, 1990.

[5] C. Barrett, R. Hughey, and K. Karplus, "Scoring hidden Markov models," *CABIOS*, vol. 13, no. 2, pp. 191–199, 1997.

[6] S. E. Brenner, *Molecular propinquity: evolutionary and structural relationships of proteins.* PhD thesis, University of Cambridge, Cambridge, England, 1996.

[7] S. Henikoff and J. G. Henikoff, "Position-based sequence weights," *JMB*, vol. 243, pp. 574–578, Nov. 1994.

[8] L. Holm and C. Sander, "Protein structure comparison by alignment of distance matrices," *JMB*, vol. 233, pp. 123–138, 5 Sept 1993.

[9] L. Holm and C. Sander, "The FSSP database: Fold classification based on structure-structure alignment of proteins," *NAR*, vol. 24, pp. 206–209, 1 Jan 1996.

[10] L. Holm and C. Sander, "Dali/FSSP classification of three-dimensional protein folds," *NAR*, vol. 25, pp. 231–234, 1 Jan 1997.

[11] T. Hubbard, A. Murzin, S. Brenner, and C. Chothia, "SCOP: a structural classification of proteins database," *NAR*, vol. 25, pp. 236–9, Jan. 1997.

[12] R. Hughey and A. Krogh, "Hidden Markov models for sequence analysis: Extension and analysis of the basic method," *CABIOS*, vol. 12, no. 2, pp. 95–107, 1996. Information on obtaining SAM is available at `http://www.cse.ucsc.edu/research/compbio/sam.html`.

[13] K. Karplus, Kimmen Sjölander, C. Barrett, M. Cline, D. Haussler, R. Hughey, L. Holm, and C. Sander, "Predicting protein structure using hidden Markov models," *Proteins: Structure, Function, and Genetics*, vol. Suppl. 1, pp. 134–139, 1997.

[14] A. Krogh, M. Brown, I. S. Mian, K. Sjölander, and D. Haussler, "Hidden Markov models in computational biology: Applications to protein modeling," *JMB*, vol. 235, pp. 1501–1531, Feb. 1994.

[15] NRP (Non-Redundant Protein) Database. Distributed on the Internet via anonymous FTP from ftp.ncifcrf.gov, under the auspices of the National Cancer Institute's Frederick Biomedical Supercomputing Center., 1998.

[16] J. Park, S. Teichmann, T. Hubbard, and C. Chothia, "Intermediate sequences increase the detection of homology between sequences," *JMB*, vol. 273, pp. 349–354, 1997.

[17] W. Pearson, "Comparison of methods for searching protein sequence databases," *Protein Science*, vol. 4, pp. 1145–1160, 1995.

[18] K. Sjölander, K. Karplus, M. P. Brown, R. Hughey, A. Krogh, I. S. Mian, and D. Haussler, "Dirichlet mixtures: A method for improving detection of weak but significant protein sequence homology," *CABIOS*, vol. 12, pp. 327–345, Aug. 1996.

[19] C. Tarnas and R. Hughey, "Reduced space hidden Markov model training," *Bioinformatics*, vol. 14, no. 5, pp. 401–406, 1998. **USE TARNAS98**.