

# New Local Potential Useful for Genome Annotation and 3D Modeling

John-Marc Chandonia and Fred E. Cohen\*

Department of Cellular and  
Molecular Pharmacology  
University of California  
Genentech Hall, 600 16th  
Street, Box 2240, San Francisco  
CA 94143-2240, USA

A new potential energy function representing the conformational preferences of sequentially local regions of a protein backbone is presented. This potential is derived from secondary structure probabilities such as those produced by neural network-based prediction methods. The potential is applied to the problem of remote homolog identification, in combination with a distance-dependent inter-residue potential and position-based scoring matrices. This fold recognition jury is implemented in a Java application called JThread. These methods are benchmarked on several test sets, including one released entirely after development and parameterization of JThread. In benchmark tests to identify known folds structurally similar to (but not identical with) the native structure of a sequence, JThread performs significantly better than PSI-BLAST, with 10% more structures identified correctly as the most likely structural match in a fold library, and 20% more structures correctly narrowed down to a set of five possible candidates. JThread also improves the average sequence alignment accuracy significantly, from 53% to 62% of residues aligned correctly. Reliable fold assignments and alignments are identified, making the method useful for genome annotation. JThread is applied to predicted open reading frames (ORFs) from the genomes of *Mycoplasma genitalium* and *Drosophila melanogaster*, identifying 20 new structural annotations in the former and 801 in the latter.

© 2003 Published by Elsevier Ltd.

**Keywords:** secondary structure; pseudopotential; threading; genome annotation

\*Corresponding author

## Introduction

Efforts in recent years have succeeded in elucidating the complete genome sequences of many organisms. A major challenge in the post-genomic era will be to determine the cellular functions of each protein and potential mutants, especially variations involved in disease. Determining the three-dimensional structure of a protein is a key step in acquiring a detailed understanding of enzymatic reaction catalysis and the interaction of proteins

with other molecules. However, predicting protein structure from its amino acid sequence remains one of the fundamental challenges of computational biology. For proteins with structures similar to one that has been determined experimentally, this problem is largely reduced to locating the similar fold and aligning it correctly with the new sequence.<sup>1</sup> For sequences with more than 25–30% identity with a protein of known structure, this can be accomplished by pairwise sequence alignment methods;<sup>2</sup> some of these tools, such as BLAST,<sup>3</sup> are still in widespread use today because of their speed. More remote homologs must be detected through sequence profile-based methods such as PSI-BLAST,<sup>4,5</sup> or by threading the sequence onto known folds using pseudopotential energy calculations.<sup>1,6</sup> The most accurate methods currently available are based on a combination of profile-based scoring and conformational energy evaluation.<sup>7–11</sup> Accurate prediction of novel folds is particularly important for structural genomics

Present address: J.-M. Chandonia, Berkeley Structural Genomics Center, Physical Biosciences Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA.

Abbreviations used: ORF, open reading frame; PSSM, position-specific scoring matrix; nr, non-redundant sequence database; MG, *Mycoplasma genitalium*.

E-mail address of the corresponding author:  
[cohen@cmpharm.ucsf.edu](mailto:cohen@cmpharm.ucsf.edu)

efforts,<sup>12</sup> as proteins reliably assigned to the current repertoire of folds are often eliminated as candidates for experimental structure determination by structural genomics groups.<sup>13</sup>

In order to model all proteins in newly sequenced genomes, it is necessary to recognize the structural templates associated with each gene sequence, and to produce accurate alignments of the sequences to their structural templates. Current state of the art modeling tools such as MODELLER<sup>14</sup> are critically dependent on accurate alignment to the template.<sup>15</sup> While fold recognition accuracy has improved gradually over time, alignment accuracy has not improved significantly until recently.<sup>8</sup> Computational speed is an important factor if we are to apply fold recognition methods to all the predicted gene sequences in large genomes. Conformational energy methods based on non-local interactions, such as potentials of mean force between amino acid residues,<sup>16</sup> are powerful, but computationally expensive.

Several aspects of protein structure, such as solvent exposure of amino acid residues and secondary structure, may be predicted directly from the primary sequence using tools such as neural networks.<sup>17–19</sup> This approach is usually computationally less expensive than threading, and the resulting predictions can be combined with existing threading methods. Integration of these predictions has been shown to improve the accuracy of remote homolog detection.<sup>9,20–22</sup> However, these predictions have not been demonstrated to improve alignment accuracy significantly. It is possible that this deficiency is caused by sub-optimal encoding of the structural predictions, or by inefficient combination of the prediction-based scoring terms with other metrics.

Direct comparison of the alignment accuracy of different methods is difficult, due to the lack of common benchmark data sets and even common measures of alignment accuracy. The latter measures generally fall into two categories: those based on the number of aligned residues in common with a reference alignment (i.e. Marchler-Bauer & Bryant<sup>23</sup>), and those based on correlation between contact maps of a model derived from threading and the correct structure (i.e. Panchenko *et al.*<sup>8</sup>). However, some improvement has clearly been shown in recent years in cases where one method has been compared directly to another on identical test sets. For example, the FUGUE method<sup>24</sup> was compared directly to CLUSTALW<sup>25</sup> on a set of 27 remote homologs (<20% sequence identity); average alignment accuracy improved from 32.6% to 51.1%. The COBLATH method<sup>10</sup> was compared to PSI-BLAST on a set of 307 structural pairs. Accuracy was assessed by deriving models from the alignments and counting the number with root-mean-square deviation of less than 8 Å from the correct structures; this number improved from 202 models derived from PSI-BLAST alignments to 223 derived from COBLATH alignments.<sup>10</sup> The 3D-PSSM<sup>9</sup> method extends stan-

dard sequence-based methods using evolutionary relationships identified manually in the SCOP database,<sup>26</sup> along with secondary structure predictions and a solvation potential. On a test set of 136 homologous pairs of proteins undetectable by PSI-BLAST, 3D-PSSM was able to reliably detect 18% of the relationships.<sup>9</sup>

Improvement in methods has been demonstrated through community participation in fold prediction servers, such as LiveBench<sup>27</sup> and EVA.<sup>28</sup> In particular, fold recognition methods based on the “meta-server” approach of combining structural models produced by many separate servers running a variety of algorithms has been shown to produce more accurate models than any of the individual servers.<sup>29,30</sup> It is expected that development of additional individual prediction methods will further enhance the accuracy of these meta-servers.<sup>29</sup>

Here, we present a new statistically derived potential, which represents the local conformational preferences of a protein backbone. This potential is combined with other scoring metrics, such as sequence profile-based matrices from PSI-BLAST and a distance-dependent inter-residue potential.<sup>16</sup> The combined method is tested on several benchmark data sets previously developed for comparison of threading methods.<sup>20,21</sup> Both fold recognition and alignment accuracy are demonstrated to improve significantly over current methods such as PSI-BLAST. We present results of our method on a recent set of LiveBench<sup>27</sup> targets, for comparison with other prediction methods and to benchmark accuracy on a set of structures that were all released after development and parameterization of our algorithm were completed. We also apply our method to open reading frames (ORFs) from the *Mycoplasma genitalium* and *Drosophila melanogaster* genomes, to identify new structural and functional assignments, and to determine additional proteins that may be modeled.

## Results

### Alignment accuracy

The Defay/Cohen benchmark set of proteins<sup>21</sup> contains 126 structural matches (see Materials and Methods). Correct alignments were generated by structural superposition, as described in Materials and Methods. Alignments were generated for each of the sequence/fold pairs using global dynamic programming with several different scoring functions. The Identity scoring method simply assigns a score of 1 for a match, and 0 for a non-match. BLOSUM62 is a 20 × 20 scoring matrix used by default with BLAST.<sup>3</sup> The position-specific scoring matrix (PSSM) generated by PSI-BLAST<sup>4</sup> using the sequence as a probe against the non-redundant sequence database (“nr”) was also used as a scoring method. Finally, several scoring functions based on secondary structure were

tested, individually and in combination with the PSI-BLAST PSSM. These are labeled P1–P3. P1 is a simple scoring function that assigns a score of 1 for a match of predicted secondary structure in the sequence with the known secondary structure in the fold, and scores 0 for a non-match; this is similar to the scoring system used by 3D-PSSM.<sup>9</sup> P2 is based on predicted secondary structure probabilities; a score from 0 to 1 is assigned on the basis of the predicted probability of the sequence, assuming the same secondary structure as the fold. P3 is the new local backbone potential, described in Materials and Methods and shown in Figure 3. The average accuracy for each scoring function is summarized in Table 1.

Several results are apparent from Table 1. First, there is an approximately 9% improvement in alignment accuracy when combining the new local potential (P3) with the PSI-BLAST PSSM, compared to using the PSSM alone. The results of differences in accuracy on individual sequences (which are not weighted by sequence length) form a distribution with a mean of 10.3(±1.5)%, and a standard deviation of 17.1%. Second, there are significant differences in accuracy depending on how secondary structure predictions are encoded. Consider exact matches to the structural alignment, ASNS0, and those scores that accommodate a tolerance of ±1 or ±4 residues, ASNS1 and ASNS4. By comparing the ASNS0 and ASNS4 columns, it is apparent that all secondary structure prediction-based potentials (P1–P3) were effective at producing an accurate rough alignment of secondary structure elements, while allowing small shifts of one to three residues. Purely sequence-based scoring methods such as BLOSUM62 and PSI-BLAST showed a smaller difference between ASNS0 and ASNS4. The new local potential (P3) performed better at the ASNS0 level than the other two prediction-based scoring methods, P1 and P2: the distribution of the differences in ASNS0 between

P3 and P2 for individual sequences has a mean of 5.4(±1.4)%; the equivalent distribution of differences in ASNS0 between P3 and P1 has a mean of 9.3(±1.7)%, so P3 is significantly more accurate than P2 or P1. This improvement may be due to separate parameterization of Gly, Pro, and Asn residues in the new potential; separate parameterization of these residues could lend P3 some of the advantageous properties of sequence-based scoring methods. The new local potential (P3) also performed better in combination with PSI-BLAST than the probability-based potential, P2. For both P2 and P3, nine possible weighting combinations with the PSSM were tested, ranging from 10% P2 (or P3) and 90% PSSM, to 90% P2 (or P3) and 10% PSSM. The optimal results for each, which occurred at 60% P2 and 70% P3, are reported here. Because the relative scales of both local potentials and the PSSM are arbitrary, no conclusion about the importance of secondary structure can be drawn from the higher weighting of the local potentials.

### Estimation of alignment accuracy

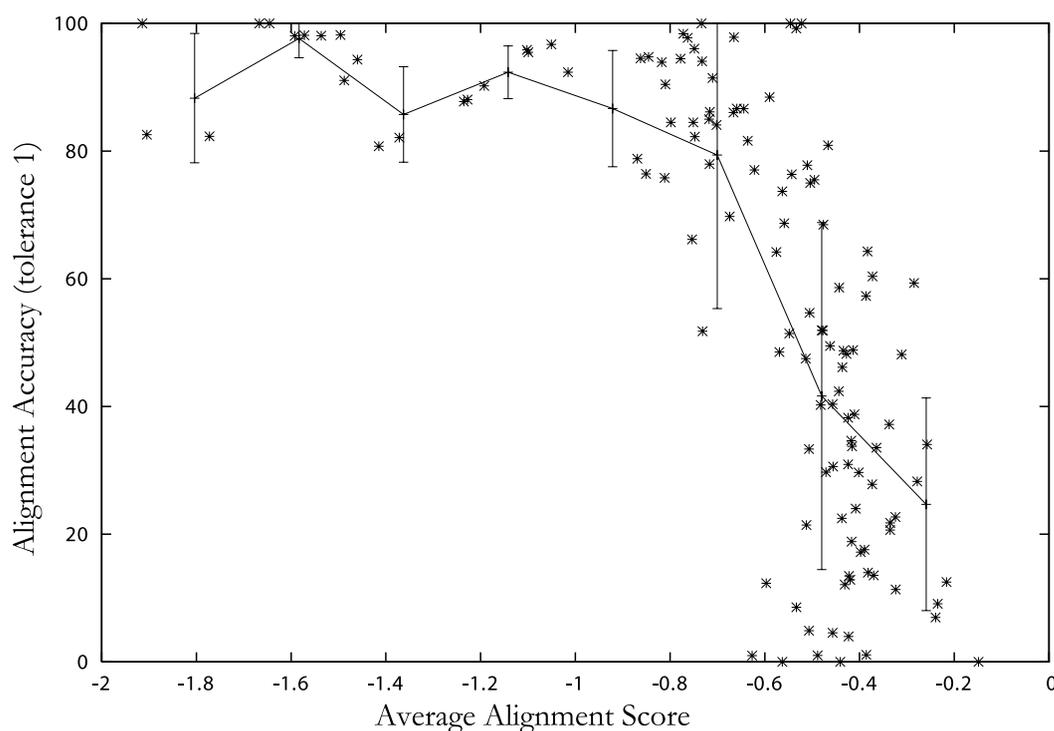
Although the accuracy of the combined scoring function is significantly better than for other methods tested (the mean improvement in accuracy over the next best method is 4.9(±1.6)%), there is considerable variation among individual proteins. Percentage accuracy values for each of the 126 structural matches form a distribution with a mean of 57.7% and a standard deviation of 32.9%, leading to great uncertainty in the value of any alignment for further modeling. The method performs significantly better on more homologous sequences. For the 56 structural matches with 12% or greater sequence identity in the structural alignment, accuracy forms a distribution with an average of 88.6% and a standard deviation of 8.9%. Unfortunately, sequence identity in the structural alignment cannot be measured *a priori*, and sequence identity in the calculated alignment does not correlate well with accuracy (data not shown).

One metric that does correlate well with accuracy, and can be measured in the calculated alignments, is average alignment score. This is the total score, including gap penalties, resulting from the dynamic programming calculation, divided by the number of aligned residues. The 49 matches with the best average scores also have significantly more accurate alignments; alignments in this subset are 88.6% accurate, on average, with a standard deviation of 9.9%. A plot of accuracy *versus* alignment score is shown in Figure 1. Alignment scores are sorted into eight bins of equal width, and the average and standard deviation in accuracy within each bin are plotted. This principle was used to derive a rough estimate of the accuracy of any alignment based on the average alignment score; details are given in Materials and Methods.

**Table 1.** Accuracy of different scoring methods on the Defay/Cohen test set

Scoring method	% Accuracy	ASNS0	ASNS1	ASNS4
Identity	37.7	25.8	35.7	55.2
BLOSUM62	48.1	36.4	45.3	63.4
P1—predicted 2ary (simple)	40.3	15.4	38.8	64.7
P2—pred. 2ary (probabilities)	45.3	20.3	43.8	69.4
P3—pred. 2ary ( $\phi/\psi$ )	45.9	25.4	43.8	66.9
PSI-BLAST	53.0	40.5	48.6	63.9
P2/PSI-BLAST combination	55.4	41.3	52.0	73.8
P3/PSI-BLAST combination	62.0	47.0	56.8	75.1

Scoring methods are described in the text. Data in the % Accurate column are calculated using a tolerance of ±1. ASNS $n$  is alignment sensitivity (see the text for a full definition) calculated using a tolerance of ± $n$ .



**Figure 1.** Alignment accuracy *versus* average alignment score. Alignment scores are sorted into eight bins of equal width, and the average and standard deviation in accuracy within each bin is plotted (error bars indicate one standard deviation). Alignment accuracy is calculated using a tolerance of  $\pm 1$ .

### Additional test sets

Alignments were performed on the Fischer/Eisenberg test set, using the same scoring methods that were tested on the Defay/Cohen set. Optimal gap penalties and relative weighting when combining the new local potential with the PSI-BLAST PSSM were not recalculated. Structural matches and sequence alignments were calculated in several different ways. For direct comparison to the Defay/Cohen test set results, structural matches and correct sequence alignments were calculated using MINAREA. This data set includes 128 structural matches, a result similar to the number of matches in the Defay/Cohen set. Results on these structural matches are shown in Table 2, columns 1–4. For comparison to other groups, the 68 structural matches (one per sequence) used in the original work<sup>20</sup> were tested. The results are shown in the last column of Table 2.

On the Fischer/Eisenberg test set, results are somewhat more accurate when tested on the Fischer/Eisenberg structural matches than on the matches identified by MINAREA. The Fischer/Eisenberg structural matches contain only the best match possible for each sequence; MINAREA identifies 128 possible structural matches, an average of almost two per sequence. The MINAREA set includes more remote homologs, for which results are less accurate. For most methods tested, the accuracy on the Fischer/Eisenberg data set is somewhat lower than on the Defay/Cohen data set. This is likely due to a larger number of multi-

domain proteins in the Fischer/Eisenberg set. Because all tests were performed using global alignments, rather than local, the method does not perform as well when on larger proteins with multiple domains. However, the results are qualitatively similar; the improvement resulting from integration of the new local potential with the PSI-BLAST PSSM is reduced from 9% on the Defay/Cohen set to 5–6% on the Fischer/Eisenberg set. Other methods have been tested on the Fischer/Eisenberg set. The ASNS0 of GenTHREADER alignments is reported for 44 structural matches correctly identified by the method; accuracy was

**Table 2.** Accuracy of different scoring methods on the Fischer/Eisenberg test set

Scoring method	% Accurate				Fischer % accurate
	ASNS0	ASNS1	ASNS4	ASNS0	
Identity	35.7	24.8	33.9	53.4	35.9
BLOSUM62	46.6	34.1	44.3	64.6	48.3
$\phi/\psi$	44.4	24.1	42.0	66.4	50.6
PSI-BLAST	53.0	39.1	49.9	66.7	57.5
$\phi/\psi$ /PSI-BLAST combination	58.0	41.8	53.2	69.7	63.4

The rightmost column shows accuracy calculated for Fischer/Eisenberg matches, and the other columns show accuracy calculated for MINAREA matches. Data in the % Accurate column are calculated using a tolerance of  $\pm 1$ . Scoring methods are described in the text.

calculated relative to reference alignments created using the structural superposition program SSAP.<sup>31</sup> The average ASNS0 of GenTHREADER on these matches, weighted by alignment length, is 44.7%.<sup>7</sup> For the same 44 matches, the average ASNS0 of the combined scoring function described above is 58.2%. The calculated ASNS0 on this subset is larger than for the entire set because GenTHREADER's accuracy is not reported for pairs that were not ranked first by its fold recognition algorithm; the other 24 pairs are presumably more difficult.

Both the Defay/Cohen test set and the Fischer/Eisenberg test set were submitted to the 3D-PSSM server. Because 3D-PSSM is available only as a server and not as a downloadable program, the fold library could not be controlled. The current 3D-PSSM fold library contains proteins with at least 70% sequence identity with every protein in both test sets, with 100% identical sequences available for the majority of proteins in both sets. Interestingly, the 100% identical matches were not always the top hit returned by the server. Alignment accuracy could be directly compared only when the 3D-PSSM server returned a match to a fold that was identical with one of the proteins in the Defay/Cohen or Fischer/Eisenberg fold libraries. For the Defay/Cohen test set, 19 of 126 matches could be compared directly. Of these, all statistics were statistically indistinguishable, with 3D-PSSM 1.6% ahead on ASNS0 and the combined P3/PSI-BLAST scoring function 2.3% better on ASNS4. For the Fischer/Eisenberg test set, 28 of 128 matches could be compared. For these matches, 3D-PSSM performed significantly better than the combined P3/PSI-BLAST potential (average results weighted by alignment length: 58% *versus* 49% for ASNS0, and 85% *versus* 68% for ASNS4). However, these results are not expected to be indicative of performance on newly sequenced proteins, because very similar test sequences were included in the 3D-PSSM fold library and, presumably, the training set. For the P3/PSI-BLAST potential, similar sequences were excluded from training sets as described in Materials and Methods.

### Accuracy of structural models

Although direct measures of alignment accuracy are useful for comparing methods, it is informative to compare the quality of the implied structural models. Because alignment accuracy is a major factor influencing model quality,<sup>15</sup> accurate alignments are a necessary but not sufficient prerequisite for accurate models. Models were built for each sequence in the Defay/Cohen and Fischer/Eisenberg test sets from calculated alignments to the optimal fold library templates using MODELLER,<sup>14</sup> version 6v2, with default options (the "model" routine with one model). Models were compared to the correct structures using MaxSub.<sup>32</sup> To compensate for inaccuracies caused

by the modeling procedure rather than the alignments, we also built "optimal" models from the correct alignments (calculated from a structural superposition; see Materials and Methods) and calculated MaxSub scores for these models.

The results on the two data sets were very similar. The optimal models calculated from the correct alignments had average MaxSub scores of 0.49 in each set, out of a possible 1.0 for a perfect model. This difference reflects limitations in the automated modeling procedure and structural dissimilarities between the fold templates and the true structures. Rankings for other methods were similar to the rankings for alignment sensitivity. Average MaxSub scores for each method and data set are shown in Table 3. The best predicted models in each set were produced by a combination of the new secondary structure prediction-based potential and the PSI-BLAST PSSM.

### Fold recognition accuracy

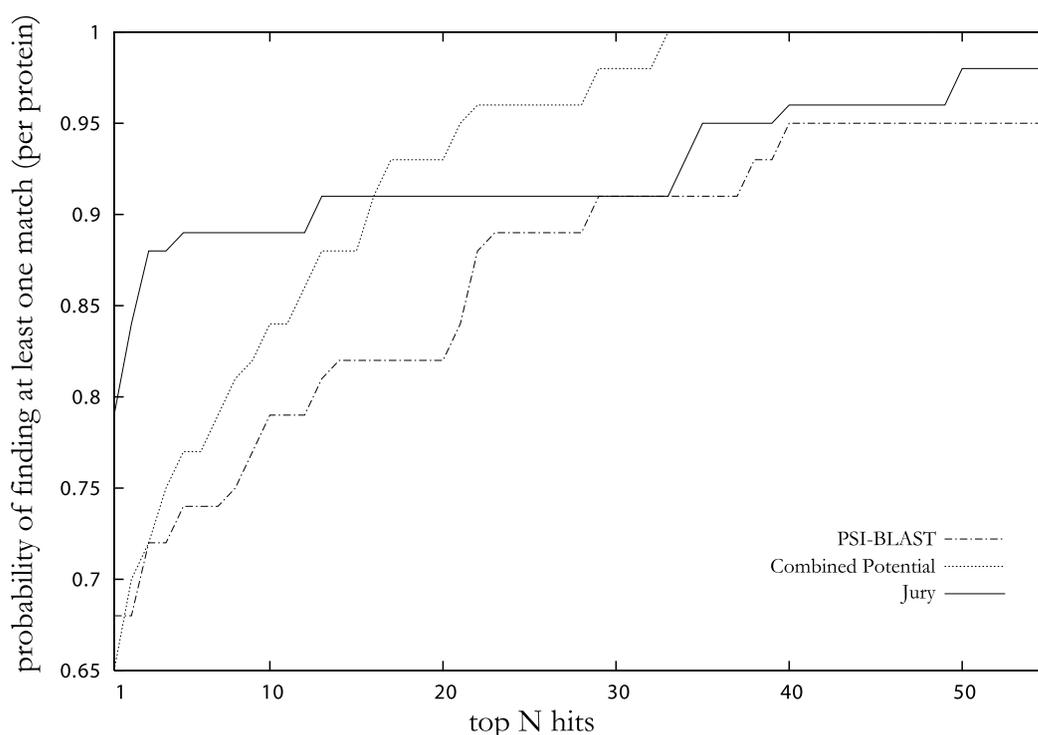
Although a simple combination of the new local potential with the PSI-BLAST PSSM improves alignment accuracy and some aspects of fold recognition accuracy, fold recognition accuracy is further enhanced using a jury method. This method is described in detail in Materials and Methods, and outlined in Figure 4.

The Defay/Cohen test set contains 58 sequences for which at least one structural match is present in the fold library. Using the "one-to-many" test of fold recognition accuracy, described in Materials and Methods, the probability of finding a match among the top *N* hits was calculated for several scoring methods. Results for the PSI-BLAST PSSM, a combination of the new local potential with the PSI-BLAST PSSM, and the fold recognition jury are compared in Figure 2.

While the PSI-BLAST PSSM correctly identifies a matching fold as the top hit for 67% of the test sequences, subsequent hits are less likely to identify correct matches. The chance of a correct fold occurring anywhere among the top five hits is 74%, and the chance of a correct fold occurring anywhere in the top 20 hits increases to only 82%. The combination of the new local potential and the PSSM is less accurate for the top hit (65% *versus* 67%), but more useful for finding a correct match

**Table 3.** MaxSub scores for models created from alignments using different scoring methods on the Defay/Cohen (D/C) and Fischer/Eisenberg (F/E) test sets

Scoring method	Average	
	D/C	F/E
Identity	0.21	0.19
BLOSUM62	0.27	0.25
$\phi/\psi$	0.22	0.20
PSI-BLAST	0.28	0.29
$\phi/\psi$ /PSI-BLAST combination	0.34	0.32
Correct alignments	0.49	0.49



**Figure 2.** Fold recognition accuracy. Using the “one-to-many” test of fold recognition accuracy, described in Materials and Methods, the probability of finding a match among the top  $N$  hits was calculated for several scoring methods.

among the top five hits (77% versus 74%) or top 20 hits (92% versus 82%). Potential users of the threading tool would presumably be most interested in the accuracy of the first hit, or first several hits, as further investigation of possible structural matches might be conducted manually or with the help of more specific and time-sensitive algorithms. Therefore, the fold recognition jury was tuned to obtain maximum accuracy among the top five hits. The resulting accuracy for the top hit was 79% (versus 67% for the PSI-BLAST PSSM), and an accuracy rate of 88% was obtained for the top three hits. However, little additional benefit is gained from examination of hits beyond the best three; the combination of the PSSM with the new local potential becomes more reliable when considering more than 15 possible candidates.

Fold recognition tests were performed also on the Fischer/Eisenberg benchmark set, using the set of 68 matches supplied by Fischer as the correct standard. As in other studies,<sup>7</sup> matches containing at least one common domain classified in the same homologous superfamily in the CATH<sup>33</sup> structural database were counted as correct, resulting in a total of 213 possible structural matches. The PSI-BLAST PSSM identifies a matching fold correctly as a top hit in 75% of the test sequences. The probability of a correct match increases to 84% among the top five hits, and to 87% among the top 20 hits. As with the Defay/Cohen benchmark set, the combination of the new local potential and the PSSM is less accurate for the top hit

(69% versus 75%), but more accurate when the top five hits (85% versus 84%) or the top 20 hits (93% versus 84%) are considered. The jury method is more accurate than either of the other methods, finding a match as the top hit for 76% of the sequences, 93% in the top five, and 96% in the top 20. The jury method compares favorably with other fold recognition methods tested on the same data set. GenTHREADER<sup>7</sup> finds a match as the top hit for 74% of the sequences, with 82% in the top five and 94% in the top 20. All sequences in the Fischer test set were submitted to the 3D-PSSM server.<sup>9</sup> Because the fold library could not be controlled and contained many of the test sequences, results with more than 25% sequence identity with the submitted sequence were ignored. Folds returned by the 3D-PSSM server were mapped to CATH superfamilies, allowing an overlap of up to ten residues at each end of the sequences. In cases where a single fold overlapped several CATH superfamilies, a match with any of them was counted as correct. 3D-PSSM found a match as the top hit for 65% of the sequences, with 88% in the top five, and 96% in the top 20. However, these results are not comparable directly to those reported for the jury method or GenTHREADER, since the 3D-PSSM library is larger than the Fischer fold library.

The complete jury method, including estimation of fold recognition accuracy as described in Materials and Methods, is implemented as a Java application called Jthread, which also performs

sequence alignment on potential structural matches, using the optimal alignment parameters described above. In addition to identifying a large percentage of correct structural matches, a fold recognition method is most useful for annotation if it produces a low rate of false positives. JThread was parameterized on the Defay/Cohen data set, so all annotations on that set with estimated accuracy >99% were indeed true positives. On the Fischer/Eisenberg data set, 58 structural matches (representing 32 of the 68 sequences) were annotated at confidence levels of >99%. Of these, three matches (all immunoglobulins) initially appeared to be false positives. However, two of the three structures have been classified as immunoglobulins in a more recent version (2.0) of CATH, and assigned the same CATH code as the potential matches predicted by JThread. A third protein (PDB code 1PFC) remains unclassified in CATH. Examination of the 1PFC structure and its headers indicates that it is also an immunoglobulin domain, as predicted by JThread.

### LiveBench results

JThread was used to predict folds for a recent set of LiveBench<sup>27</sup> targets, to benchmark accuracy on a set of structures that were all released after development and parameterization of our algorithm were completed. LiveBench Set 6 includes 98 sequence targets, and is pre-filtered to exclude “easy” targets for which a similar PDB sequence can be detected using BLAST. All targets, as well as all proteins in the JThread fold library, have recently been classified in SCOP version 1.63, which allows accuracy to be benchmarked on the basis of manual annotation by an expert. The structure 1IYA was superseded in the PDB by 1J3G, which was substituted for purposes of this analysis. JThread predicted 36 matches, covering 12 sequences, with >99% confidence. According to SCOP, all 36 predicted folds were classified in the same homologous superfamily as the corresponding target protein. Predictions made at lower confidence were examined. Of the top matches for 98 targets, 24 (24%) were in the correct superfamily, and three more (3%) were in the correct fold but different superfamilies, possibly indicating detection of analogous folds. When the top five predictions for each target were examined, 28 targets (29%) had at least one match in the correct superfamily, and ten (10%) more had matches in the correct fold but different superfamilies. Within the top ten predictions, 33 (34%) were predicted in the correct superfamily, and 11 (11%) more were predicted in the correct fold. Within the top 20 predictions, 39 (40%) were predicted in the correct superfamily, and 15 (15%) more were predicted in the correct fold. As these statistics were compiled on a set of proteins assembled after the development and parameterization of JThread, they give an unbiased sampling of the accuracy of the algorithm in making non-trivial predictions for newly sequenced

proteins. Unfortunately, due to time and memory requirements of JThread, it is currently impractical to provide a server that could participate in ongoing LiveBench<sup>27</sup> or EVA<sup>28</sup> evaluations.

### *Mycoplasma genitalium* genome

*Mycoplasma genitalium* (MG) is the smallest bacterial genome, with 480 predicted ORFs.<sup>34</sup> It has therefore been used to test several recently developed, fully automated methods for structural annotation.<sup>19,35,36</sup> We applied the pipeline method described in Materials and Methods to this genome to identify structural annotations for 270 (56.2%) of the ORFs. However, as the first methods in the pipeline are the local alignment algorithms BLAST and PSI-BLAST, a significant number of annotations covered only part of the sequence of the corresponding ORF. For example, the ORF MG104 is 725 amino acid residues long, but only a single domain of 72 amino acid residues could be annotated as having significant structural similarity to a known RNA-binding domain (PDB code 1SRO). Nevertheless, 213 of the 270 annotations (78.8%, or 44.3% of the ORFs) accounted for at least 50% of the sequence of the corresponding ORF. In total, the structural annotations account for 78,265 of the 174,959 residues (44.7%) in the MG genome. If short insertions (ten residues or fewer) are included in these statistics, the numbers increase to 217 annotations (80.3% or 45.2% of the ORFs) covering at least 50% of the sequence of the ORF, and 80,315 residues annotated (45.9%). All annotations for MG are summarized on our web site†.

Of the 270 annotations, 112 (41.4%, or 23.3% of the ORFs) were obtained using BLAST. An additional 138 annotations (51.1% of the annotations, or 28.7% of the ORFs) were obtained using PSI-BLAST. The remaining 20 annotations (7.4% of the annotations, or 4.1% of the ORFs) were obtained using JThread. Although the number of additional annotations that were found using JThread (but not PSI-BLAST) was relatively small, it included some annotations that were missed by automatic application of BLAST and PSI-BLAST. For example, three predicted ribosomal proteins (MG155, MG161, and MG174) are over 60% identical with the sequences of the matching PDB structures. However, these matches were not found by BLAST or PSI-BLAST, due to the low-complexity filter used in these algorithms. Although a different choice of BLAST parameters (eliminating the filter) might have alleviated this problem, this would likely have increased the potential for false positives. These cases illustrate the difficulty of setting up a fully automatic annotation system, and the importance of applying a pipeline procedure including several different methods to the annotation problem. We will discuss two additional examples in more detail:

† <http://www.cmpfarm.ucsf.edu/~jmc/mg/>

### ORF MG111: phosphoglucose isomerase

Phosphoglucose isomerase is a key enzyme in the glycolytic pathway, and therefore likely to be found in even the smallest bacterial genomes. MG111 could not be annotated as related to a protein of known structure by either the BLAST or PSI-BLAST algorithms. However, it was predicted to have structural similarity to a structure of phosphoglucose isomerase (PDB code 1BOZ) by JThread.

To assign functional as well as structural similarity, it is important to verify conservation of functionally important residues. Residues that are conserved in phosphoglucose isomerase enzymes from 42 different species were obtained from the PROSITE database.<sup>37</sup> A sequence alignment to the enzyme structure and an estimation of the alignment accuracy were calculated as described above. The resulting alignment is estimated to be over 88% accurate, and reveals complete conservation of all 22 conserved residues from the PROSITE motif. Therefore, the annotation of MG111 as phosphoglucose isomerase is fairly certain, and the sequence alignment could be used to produce a low-resolution model of the structure with homology modeling tools such as MODELLER.<sup>14</sup> As corroborating evidence that MG111 is phosphoglucose isomerase, the same structural classification is also made by the 3D-PSSM algorithm.<sup>9</sup> This prediction could be confirmed easily through biochemical analysis.

### ORF MG265: an enzyme with unknown function

MG265 is a conserved hypothetical protein with unknown function. MG265 could not be annotated as related to a protein of known structure by either the BLAST or PSI-BLAST algorithms. However, it was predicted to have structural similarity to a domain from L-2-haloacid dehydrogenase (PDB code 1QQ5, chain A) by JThread. This annotation implies that MG265 forms a multi-domain structure including a Rossmann fold.

In this case, an accurate sequence alignment could not be calculated (estimated alignment accuracy of only 27%), so a quantitative measure of the conservation of functionally important residues could not be determined. Furthermore, a reliable model cannot be constructed without a more accurate alignment. Structures containing a Rossmann fold are frequently enzymes that use the Rossmann fold domain to bind the substrate or a co-factor.<sup>38</sup> However, in this case, the specific type of enzyme cannot be determined without additional experimental work.

### *Drosophila melanogaster* genome

The fruit fly (*Drosophila melanogaster*) genome<sup>39</sup> contains 13,608 predicted ORFs, comparable in size to the 35,000–40,000 genes predicted for the human genome.<sup>40</sup> It is therefore a good benchmark

for annotation methods applicable to large eukaryotic genomes. We applied the pipeline method, resulting in structural annotations for 6717 (49.4%) of the ORFs. Although the fraction of annotated genes is similar to that for MG, the fly has a greater proportion of genes for which the annotation covers only part of the sequence of the corresponding ORF. Only 2938 of the 6717 annotations (43.7% or 21.6% of the ORFs) accounted for at least 50% of the ORF sequence, compared to 78.8% of the MG annotations. In total, the structural annotations account for 1,430,851 of the 6,600,557 residues (21.7%) in the fly genome. These numbers may be smaller compared to the MG because the fly contains more long, multi-domain proteins (average ORF length is 485 residues in the fly, *versus* 364 in MG), and no effort was made to annotate additional regions of an ORF once one region had been structurally annotated. It is possible that because the MG genome is more compact, a larger percentage of these proteins are conserved in multiple species, and thus have a greater change of homology to a protein that has been studied and structurally characterized. A recent survey of genomic ORFans (proteins with no detectable sequence similarity to proteins in other genomes) found no remaining ORFans in MG, but as many as 33% in larger bacterial genomes.<sup>41</sup> All annotations for *D. melanogaster* are summarized on our web site†.

Of the 6717 annotations, 2719 (40.5%, or 20.0% of the ORFs) were obtained using BLAST. An additional 2999 annotations (44.6% of the annotations, or 22.0% of the ORFs) were obtained using PSI-BLAST. The remaining 801 annotations (11.9% of the annotations, or 5.9% of the ORFs) were obtained using JThread. These numbers suggest that the lower annotation rate in the fly relative to MG is due to a greater number of remote homologs or ORFans, rather than simply being the result of longer proteins. BLAST and PSI-BLAST are local alignment algorithms, and are capable of identifying a single domain in a multi-domain protein. However, the annotation rates for these algorithms were both lower in the fly than in MG. JThread, which uses a global alignment algorithm, would be expected to miss some multi-domain proteins, because the fold library contains only single domains. However, the annotation rate for JThread was over 40% higher in the fly, indicating a relative abundance of remote homologs.

### Common superfamilies annotated in *M. genitalium* and *D. melanogaster*

All structural annotations were identified by superfamily from the SCOP database (version 1.53). SCOP is a manually curated database that aims to identify structural and evolutionary relationships between proteins of known structure.<sup>26</sup>

† <http://www.cmpharm.ucsf.edu/~jmc/fly/>

The superfamilies most represented in structural annotations of *Drosophila* and MG are summarized in Table 4. The ten most common superfamilies from each species are shown. Three superfamilies occur in the top five rankings of each species. The first, P-loop-containing NTP hydrolases, is a very diverse family including kinases, G proteins, motor proteins, and the ATP-binding subunits of some transporters. This superfamily occurs 41 times in MG and 331 times in *Drosophila*. The second superfamily, immunoglobulin-like domains, occurs frequently in proteins attached to cell surfaces; it occurs six times in MG and 258 times in *Drosophila*. Finally, the colicin superfamily is a small family limited to a coiled-coil motif found in several related toxins produced by *Escherichia coli*. Despite the application of the low-complexity filter (SEG) in combination with PSI-BLAST (see Materials and Methods), many proteins from both *Drosophila* and MG were annotated by PSI-BLAST as having similarity to this domain. In a similar study using PSI-BLAST to annotate MG genes, proteins with coiled-coil regions were identified by using a separate procedure specialized for detection of these regions.<sup>35</sup> In the Müller *et al.* study, the number of coiled-coil proteins in the MG genome was estimated as four or five, compared to the 16 found in MG by PSI-BLAST in this study

(325 such regions were found in *Drosophila*). Therefore, the rate of false positives in this superfamily is expected to be significant in both MG and *Drosophila* annotations.

It is interesting to observe common superfamilies in MG that have not diverged significantly in *Drosophila*. Four superfamilies occur among the ten most common in MG, but have fewer than 20 members annotated in *Drosophila*. These include the anticodon-binding domains of class I and II tRNA synthetases (each of which has six annotated members in MG, and ten or 11 annotated members in *Drosophila*). Other superfamilies are a domain of SRP/SRP receptor G proteins (five members in MG and seven in *Drosophila*) and ribosomal fragments (six members in MG and 15 in *Drosophila*). The relative lack of specialization in these families may indicate that the functions, while important, are optimally performed by a small number of proteins.

### New predictions made by JThread

Of the 801 *Drosophila* annotations made by JThread, 692 (86%) are predicted to be structurally similar to proteins from SCOP families in which no member is found by BLAST or PSI-BLAST. Of the JThread annotations, 547 (68%) are novel at the SCOP superfamily level, and 223 (28%) are novel at the SCOP fold level. These predictions cluster into 34 newly annotated folds, 58 new superfamilies, and 86 new families. JThread annotations showed greater structural diversity than predictions produced by BLAST or PSI-BLAST. Although JThread produced 12% of the *Drosophila* annotations, 17% (86/515) of the SCOP families were annotated only by JThread.

Examination of these new annotations reveals some relative strengths and weaknesses of the JThread and BLAST/PSI-BLAST algorithms. As is the case with JThread annotations of MG, one of the newly annotated groups of folds includes structures similar to phosphoglucose isomerase (PGI). One of these, CG8251, is 69% identical in sequence with a structurally characterized PGI from rabbit (PDB code 1DQR). Two other genes, CG1345 and CG12449, are 38% identical in sequence with a structurally similar enzyme, the isomerase domain of glucosamine-6-phosphate synthase (GLMS). All three of these contain PROSITE motifs, suggesting conservation of function. An additional 18 annotated sequences range from 10% to 24% identity with a known structure, but functional conclusions cannot be drawn because the expected accuracy of the sequence alignment was too low (20–27%) to allow further modeling. Another similarity to the MG annotations was the discovery of two genes, CG3661 and CG14148, with structural similarity to ribosomal protein L14; sequence identity with the known structure ranges from 33 to 39%. In both the PGI and ribosomal protein L14 families, the high degree of sequence identity of annotated genes with known structures

**Table 4.** Common superfamilies annotated in *Mycoplasma genitalium* (MG) and *Drosophila melanogaster* (Fly)

Superfamily description	MG		Fly	
	Rank	Frequency	Rank	Frequency
P-loop containing NTP hydrolases	1	41	2	331
ConA-like lectins/glucanases	2	27	12	123
Colicin	3	16	3	325
Immunoglobulin	4	6	5	258
Nucleic acid-binding proteins	4	6	58	23
Translation factors	4	6	58	23
Ribosome and ribosomal fragments	4	6	82	15
Anticodon-binding domain of class I aa-tRNA synthetases	4	6	100	11
Anticodon-binding domain of class II AaRS	4	6	108	10
FAD/NAD(P)-binding domain	10	5	32	45
Domain of SRP/SRP receptor G proteins	10	5	140	7
Zn finger, C2H2	–	–	1	336
Heme-dependent peroxidases	–	–	4	268
Trypsin-like serine proteases	–	–	6	244
Interferon-induced GPB1, C-terminal Domain	13	4	7	199
Protein kinase-like	33	1	8	195
L domain-like	–	–	9	191
Transducin, $\gamma$ chain	–	–	10	177

suggests that sequence-based search methods such as BLAST should be able to annotate the genes, but were unable to for unknown reasons.

### Proteasome predictions

An interesting example of a fold for which BLAST and PSI-BLAST found no hits in *Drosophila*, but for which JThread found numerous examples, is the fold family containing proteasome  $\alpha$  and  $\beta$  subunits. In eukaryotic cells, most proteins are degraded *via* the ubiquitin-proteasome pathway.<sup>42</sup> The core of this pathway is a barrel-shaped proteolytic core complex, the 20 S proteasome. This particle is composed of 28 subunits, two copies each of seven  $\alpha$  subunits and seven  $\beta$  subunits. Two rings of  $\beta$  subunits are flanked by two rings of  $\alpha$  subunits, forming the barrel structure. Catalytic degradation of proteins is performed by three of the  $\beta$  subunits. Although much of the regulation of proteasome catalysis occurs in a 19 S particle that is attached at each end of the barrel, regulation by selective expression of subunit isoforms is known to occur. In mammals, an immune response stimulates expression of three additional active  $\beta$  subunits, each of which replaces a specific  $\beta$  subunit from the original particle. This "immuno-proteasome" is implicated in processing of antigens for presentation by MHC class I molecules. In *Drosophila*, testes-specific isoforms of proteasomes have been cloned; however, nothing is known of their functional role.<sup>42</sup> JThread identifies 36 proteins from the SCOP fold family, which includes proteasomes and similar hydrolases, including 25 for which the expected alignment accuracy with a known structure is 87% or better. In some of these gene products, the catalytic residues are conserved; others are likely to be inactive isoforms. The large number of genes suggests that selective expression of active and inactive isoforms of proteasome subunits may play a role in the regulation of protein degradation in *Drosophila*. Further modeling of the proteins and experimental characterization of the expression patterns of these genes may shed further light on this hypothesis.

### Discussion and Conclusions

As the number of completely sequenced genomes increases, there is a growing need for computational tools to aid in understanding the cellular functions of the gene products. Determining the three-dimensional structure of each protein is a key step in acquiring a detailed understanding of enzymatic reaction catalysis and the interaction of proteins with small molecule ligands and other proteins. Because computational modeling tools require an accurate alignment of a new sequence to a template protein with known structure, it is important to develop tools that can calculate these alignments accurately. We have shown that a combination of existing sequence-based potentials

with a new local potential based on secondary structure predictions creates a significant improvement in alignment accuracy over current methods. In addition, use of the JThread algorithm in a genomic annotation pipeline reveals a significant number (5–10%) of additional annotations, many of which could be used to produce structural models.

The pipeline method of genomic annotation reveals several strengths and weaknesses of JThread and other current methods. First, the size of newly sequenced eukaryotic genomes demonstrates the need for fast algorithms. JThread relies on homologous sequences identified by tools such as PSI-BLAST. Thus, the speed of the algorithm is limited by the time required to apply PSI-BLAST to a single protein. This currently averages about ten minutes on a Pentium III class computer, or about 100 days to test every protein in a typical eukaryotic genome. Although multiple computers can perform this computation in parallel, this demonstrates that algorithms even a single order of magnitude slower than PSI-BLAST could easily become computationally prohibitive. Although computational power is increasing, the number of sequences and genomes to process may be increasing at an even faster rate. Second, the pipeline revealed several proteins that were identified only by JThread, but that were similar enough to proteins of known structure that sequence-based methods would have been expected to identify them. This problem reveals the difficulty of choosing a single set of parameters in a fully automated genome annotation method. Finally, the large number of coiled-coil and other non-globular proteins identified by PSI-BLAST and JThread emphasizes the need for filtering of these proteins early in an annotation pipeline. Specialized computational tools may be needed to identify these proteins, which can create difficulty for algorithms tuned to perform on water-soluble, globular proteins.

Predictions made by JThread should be of special interest to biologists who have focused their interest on modeling a particular protein of unknown structure. Compared to other current methods, JThread has a greater probability of placing a true structural match high in a ranked list of possible fold candidates. Even in cases where a detailed annotation cannot be made by any method, thorough examination and modeling of several candidates, combined with expert knowledge of a protein of interest, may lead to a structural model.

Additional improvements to JThread are expected in several areas. First, accuracy of the algorithm could be increased through the use of additional non-local potentials such as more accurate inter-residue potentials or a potential that explicitly evaluates the burial of hydrophobic side-chains. Second, algorithms for secondary structure and solvent exposure prediction should continue to increase in accuracy as the number of known sequences increases. In addition, structural genomics initiatives should produce a more uniform

sampling of the universe of possible protein folds than is currently available in the PDB. This should result in more cases where an impossible fold recognition target becomes merely difficult. Finally, use of a local alignment algorithm and additional attempts to annotate small sections of a protein sequence that were not annotated during the initial evaluation should greatly increase the coverage of annotated sequence space in existing genomes.

Annotations of *D. melanogaster* and *M. genitalium* are available on our web site†.

## Materials and Methods

### Data sets

JThread was developed and tested on a library of 58 sequences and 305 folds used in a previous threading study.<sup>21</sup> Structures of all proteins were determined by X-ray crystallography to at least 2.5 Å resolution. Structural matches were determined by the structural superposition program MINAREA,<sup>43</sup> using the same cutoff as in the previous study (a ratio score of 0.2 or lower) for assigning structural similarity. In order to mimic blind structure prediction challenges, the true structures of the 58 test sequences and their homologs (more than 25% sequence identity) were not considered. This procedure identified 126 structural matches, an average of 2.2 per sequence. “Correct” sequence alignments were determined from the structural superposition, using MINAREA. This procedure uses dynamic programming between the template and target, based on the C<sup>α</sup>–C<sup>α</sup> distances in the structural alignment, and does not require gap penalties. The resulting alignment is filtered to remove aligned residues with an inter-C<sup>α</sup> distance greater than 6 Å, and aligned segments shorter than two residues.

Further testing was done using a library of 68 sequences and 301 folds introduced by Fischer & Eisenberg,<sup>20</sup> and commonly used to benchmark threading studies.<sup>7,10,24</sup>

Testing was performed on a set of 98 targets downloaded from the LiveBench<sup>27</sup> server. Performance on LiveBench Set 6, which was completed most recently, was evaluated.

Genomic threading was carried out using all predicted ORFs from *M. genitalium* and *D. melanogaster*, downloaded from‡. The *M. genitalium* genome contained 480 sequences, and the *D. melanogaster* genome 13,308. These sequences were threaded against a fold library derived from the ASTRAL database of protein domains,<sup>44,45</sup> version 1.50. The 30% identity subset was used, from which proteins with incomplete structural information were discarded. The resulting fold library contains 2123 folds from a diverse set of protein families.

Inter-residue pair potentials were used in the fold recognition jury (described below). These were calculated using the method of Sippl,<sup>16</sup> on the same non-redundant database of 681 proteins used to train and test the Pred2ary program.<sup>46</sup> Potentials between C<sup>β</sup> atoms (C<sup>α</sup> for glycine residues) were used.

### Multiple sequence gathering

Multiple homologs for each protein used in the study were obtained using PSI-BLAST<sup>4</sup> version 2.0.7 and the nr database of non-redundant sequences from NCBI (downloaded 11/19/1999). All default options (0.001 *e*-value cutoff for inclusion of a sequence in the matrix calculations, filtering turned on) were used, except that the maximum number of rounds was set to 10. In cases where the PSSM used by PSI-BLAST was required for alignment calculations, this matrix was obtained using the checkpoint feature of PSI-BLAST.

### Secondary structure prediction

Secondary structure predictions for all proteins threaded in the study were obtained using the Pred2ary program.<sup>46</sup> For each residue of every sequence, Pred2ary predicts the probability of helix, strand, and coil. These are normalized to sum to 1.0, and correspond well to the actual probabilities when compared for large data sets. For soluble, globular proteins, the largest of the three probabilities corresponds to the correct secondary structure with an average accuracy of over 75%; either the first or second alternative is correct at 94% of the positions.<sup>46</sup> The “large” jury size was used for all predictions. Both the Defay/Cohen and Fischer/Eisenberg benchmark sequence sets contain proteins similar (more than 25% identical) to proteins in the training sets used to train some of the neural network jurors. Because these networks would produce more accurate predictions than could be expected in a truly blind test, they were eliminated from the large jury during prediction of the secondary structure of the proteins in question. During genomic threading trials, sequences similar to any protein of known structure (including those used previously in the Pred2ary training sets) were pre-filtered and annotated using BLAST or PSI-BLAST (as described below).

### Alignment method

All alignments were done using global dynamic programming<sup>47</sup> with an affine gap penalty.<sup>48</sup> Unaligned ends for both proteins being aligned were treated as gaps and penalized accordingly. Penalties for gap opening and extension were optimized individually for every scoring method or combination of scoring methods, using the non-linear optimization method described by Hooke & Jeeves.<sup>49</sup> This method is a heuristic search tool, and therefore not guaranteed to find the global optimum. However, it is very useful for optimization problems in which the objective function (in this case, alignment accuracy) is difficult to calculate directly from the parameters. In order to decrease the number of tunable parameters in the method and minimize the possibility of over-adaptation to a particular data set, the parameters were optimized using the Defay/Cohen data set, and not re-calculated for different data sets.

### Measurements of accuracy

Alignment accuracy for each method was calculated by comparing the calculated sequence alignments to the alignments generated by MINAREA from the structural superposition. Percentage accuracy was determined by dividing the number of correctly aligned residues in the calculated sequence alignment by the number of

† <http://www.cmpharm.ucsf.edu/~jmc/genomes/>

‡ <http://www.ebi.org>

residues aligned (to any residue, but not a gap) in both the structural and calculated alignments. Alignment sensitivity (ASNS) was also calculated, by dividing the number of correctly aligned residues by the number of residues aligned in the structural alignment. Because the former measure is more sensitive than the latter to the number of residues aligned in the dynamic programming calculation, ASNS was used as the measurement of accuracy when developing parameters for each method.

Several different levels of stringency were considered when measuring alignment accuracy. At the most stringent, or zero tolerance level, aligned positions had to be identical in the calculated and structural alignments. At the  $\pm 1$  tolerance level, a shift of one residue in the calculated alignment was considered to be correct. Automated structural alignment algorithms often differ at the zero tolerance level, but agree at the  $\pm 1$  threshold.<sup>21</sup> A tolerance level of  $\pm 4$  was tested also, corresponding to alignment differences of one helical turn. ASNS measured at a non-zero tolerance level is denoted with the level used; for example, ASNS1 indicates a  $\pm 1$  tolerance level.

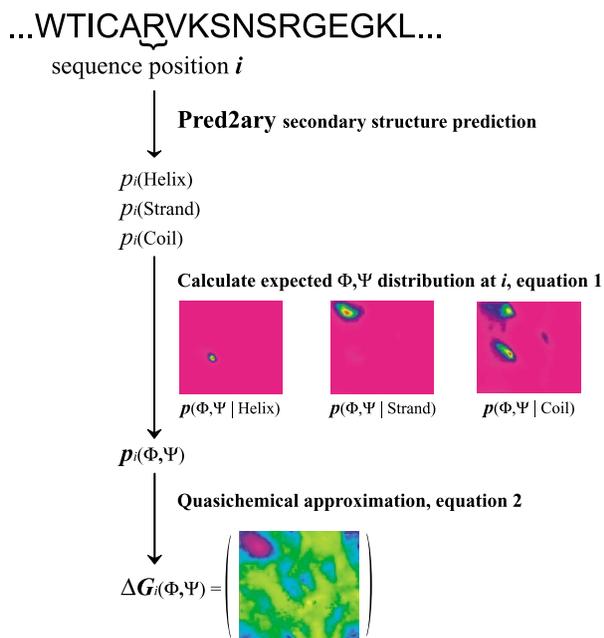
Fold recognition accuracy was measured by comparing all folds in a library to a given sequence, and ranking them according to a score calculated by a single method or jury of methods (described below). We use the one-to-many measure of successful fold recognition,<sup>35</sup> a sequence is considered to be recognized correctly if any structural match is ranked as the top hit, regardless of the rankings of other possible structural matches for that sequence. Different methods are compared according to the percentage of sequences for which a structural match is ranked as the top hit. Because for many “real life” fold recognition problems it is feasible to build and examine several alternative models, we also calculated the percentage of sequences for which a structural match was found anywhere in the top  $N$  hits. For optimizing the parameters of our method, it was desirable to calculate a measure of accuracy that was very sensitive to small changes in fold ranking. We therefore calculated the average rank of structural matches, and the reciprocal weighted average rank. The reciprocal weighted rank is useful in parameter optimization, because it places more emphasis on improvements in the relative rankings of structural matches that are already ranked fairly highly, while lowering the importance of structural matches that are ranked far down the list. However, we consider both these calculated measurements to be of less practical interest to users of the method than those discussed previously.

### Design of local backbone potential

An overview of the local backbone potential is shown in Figure 3. The Pred2ary program<sup>46</sup> predicts the probability of helix, strand or coil occurring at each position in a sequence. At sequence position  $i$ , these probabilities are denoted  $p_i(\text{Helix})$ ,  $p_i(\text{Strand})$ , and  $p_i(\text{Coil})$ , respectively. The predicted secondary structure probabilities were used to calculate the expected distribution of backbone dihedral angles for each residue in the sequence. The expected distribution of dihedral angles is computed for each residue using equation (1):

$$\mathbf{p}_i(\phi, \psi) = p_i(\text{Helix}) \times \mathbf{p}(\phi, \psi | \text{Helix}) + p_i(\text{Strand}) \times \mathbf{p}(\phi, \psi | \text{Strand}) + p_i(\text{Coil}) \times \mathbf{p}(\phi, \psi | \text{Coil}) \quad (1)$$

The distributions  $\mathbf{p}(\phi, \psi | \text{secondary str})$  are constant, and taken from a large, non-redundant set of known



**Figure 3.** Calculation of local backbone potential. The Pred2ary program<sup>46</sup> predicts the probability of helix, strand or coil occurring at each position in a sequence. These are used to calculate the expected distribution of backbone dihedral angles for each residue in the sequence, using equation (1). The expected distribution is translated into a pseudopotential using equation (2).

structures.<sup>46</sup> Because of their unusual dihedral angle preferences, the distributions for Gly, Pro, and Asn residues are calculated separately; other residue types are grouped into a single category. The expected  $\phi, \psi$  distribution is unique to every residue, although it will be identical between residues with the same secondary structural probabilities and type. This distribution is then transformed into an energy potential using the quasichemical approximation,<sup>50</sup> as shown in equation (2):

$$\Delta G_i(\phi, \psi) = -kT \times \ln \left( \frac{\mathbf{p}_i(\phi, \psi)}{\mathbf{p}_{\text{ref}}(\phi, \psi)} \right) \quad (2)$$

The reference distribution  $\mathbf{p}_{\text{ref}}(\phi, \psi)$  is a frequency distribution computed over all residues in the database, regardless of secondary structure. Because this potential is scaled arbitrarily relative to other potentials, the  $kT$  factor is ignored.

### Combination of potentials

Scoring methods that can be used directly in dynamic programming algorithms, such as residue identity-based scoring matrices, and the local potential, were simply added to each other in the dynamic programming matrix. When multiple methods were combined, each component was weighted with a single, normalized coefficient. These coefficients were optimized manually.

Several scoring methods that rely on inter-residue pair potentials (e.g. see Sippl<sup>16</sup>) cannot be applied directly in the dynamic programming algorithm. Iterative double dynamic programming methods have been used in these cases for individual threading calculations. However, this approach would slow the algorithm sufficiently to make genome-wide threading unfeasible. Scores for

these methods were calculated only for alignments that were created using dynamic programming with a different scoring method or combination of methods.

### Estimation of alignment accuracy

For some scoring methods, the average score (the total score from the dynamic programming method, divided by the number of aligned residues) was observed to correlate well with the accuracy of the alignment. Average scores were translated into estimates of the alignment accuracy (such as estimated ASNS1) using a method similar to that used for translating raw neural network output values into estimated secondary structure probabilities in previous studies.<sup>46</sup> Scores from a set of known sequence/fold pairs (the Defay/Cohen set) were used to parameterize the method. The scores were sorted into 100 ranges of equal width, encompassing the entire set of observed values. Because some ranges included only a sparse amount of data (less than ten observed values), these ranges were expanded symmetrically in each direction until they included at least ten data points. The average and standard deviation of alignment accuracy measures (such as ASNS0, ASNS1, etc.) in each range were then measured, creating a look-up table that translates observed scores into estimates of alignment accuracy. Estimates of the accuracy of new alignments are looked-up from the table according to the calculated average score; scores that are outside the range in the table are assigned estimated accuracy values corresponding to the nearest score in the table.

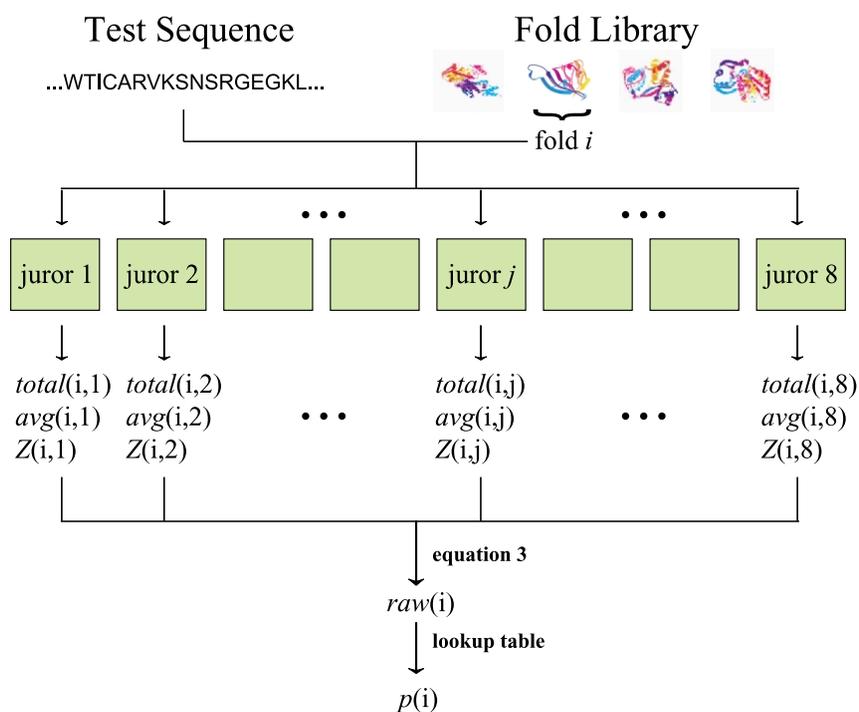
### Jury method for fold recognition

Optimal fold recognition accuracy was obtained using a jury of multiple scoring methods, shown in Figure 4. Each of eight jurors used a single scoring method or combination of methods:

*Juror 1.* The scoring method used was the PSI-BLAST PSSM obtained by using the protein sequence to be recognized (the test sequence) as a probe against the nr database, as described above in the section on Multiple sequence gathering. Gap penalties that had been optimized for alignment accuracy tests were used.

*Juror 2.* The scoring method used was a combination of the local backbone potential and an averaged PSI-BLAST PSSM. In alignment accuracy tests described above, a combination of 70% local backbone potential and 30% PSI-BLAST PSSM was found to be optimal, so these weights and the associated optimal gap penalties were used for this juror. The only difference between the scoring scheme used here and that used in the alignment tests is that the PSI-BLAST PSSM contribution was replaced with an average of PSI-BLAST PSSM scores over multiple homologs of the fold sequence (gathered using the multiple sequence gathering method described above). Gaps in the latter alignment did not contribute to the average score at each position. Because the number of detected homologs was often large, this scoring method was relatively slow compared to the other jurors.

*Juror 3.* This Juror is similar to Juror 2, except that the alignments were computed using a combination of the local backbone potential and an unaveraged PSI-BLAST PSSM (exactly as in alignment tests). The resulting alignment was then scored using the same scoring method as



**Figure 4.** Fold recognition jury. Eight jurors use different scoring methods to evaluate the compatibility of a test sequence with each fold in a fold library. The total score, average score, and Z-score for each sequence/fold/juror combination are calculated as described in the text. These are combined into a single raw score for the sequence/fold combination using equation (3). The raw score is then translated into an estimated probability of a sequence adopting a given fold, as described in the text.

that used by Juror 2. This resulted in an improvement in computational speed relative to Juror 2.

*Juror 4.* Alignments were computed as for Juror 3, and then scored using inter-residue pair potentials calculated using the method of Sippl.<sup>16</sup> The “frozen approximation”<sup>51</sup> was used: the score for placing a residue from the query sequence at a given position in a template fold was calculated relative to native residues from the fold. No gap penalties were used in scoring the alignments.

*Juror 5.* Juror 5 was the same as Juror 4, except the same gap penalties used in calculating the alignments were used in scoring the alignments.

*Juror 6.* Juror 6 was the same as Juror 4, except that the frozen approximation was not used. The score for placing a residue from the query sequence at a given position in a template fold was calculated relative to other residues from the query sequence that had been aligned to different positions in the fold. Residues that had been aligned to gaps did not contribute to the score.

*Juror 7.* Juror 7 was the same as Juror 6, except the same gap penalties used in calculating the alignments were used in scoring them.

*Juror 8.* Alignments were computed as for Juror 3, and then scored using a combination of the potential used for scoring Juror 2 and the potential used for scoring Juror 7. The scoring potential for Juror 8 contained equally weighted contributions from the two scoring potentials used by Jurors 2 and 7.

### Combination of jury scores

As shown in Figure 4, each Juror scores a given sequence against all folds in a fold library using its scoring method as described above. For every combination of sequence, fold, and Juror, the total score, average score (total score divided by the number of aligned residues), and Z-score are calculated. The Z-score is calculated from the total score, relative to the ensemble of scores produced by the same sequence and Juror, against every fold in the fold library. The Z-score measures the number of standard deviations the total score is below the mean score of the ensemble. For every sequence and fold pair, the total, average, and Z-scores contributed by each Juror are combined linearly to produce a single raw score for the pair, as shown in equation (3):

$$\begin{aligned} \text{raw}(\text{seq, fold}) &= \sum_{\text{jurors}} [w_i \times \text{total}(\text{seq, fold, juror}) \\ &+ w_{i+1} \times \text{average}(\text{seq, fold, juror}) \\ &+ w_{i+2} \times Z(\text{seq, fold, juror})] \end{aligned} \quad (3)$$

A linear, weighted, combination was used instead of a more sophisticated method (such as a neural network) in order to limit the number of tunable parameters of the method and to facilitate interpretation of the results. In addition, many of the weights were set to zero in order to limit the complexity of the method. Our optimal jury of eight methods used only 17 non-zero parameters, out of a possible 24 ( $8 \times 3$ ).

The jury weights were optimized in several steps, using the Defay/Cohen data set. Initial weights were solved *via* least-squares, with the desired raw score for each sequence/fold combination set to the MINAREA ratio score from the structural superposition. As a lower

MINAREA ratio score represents a closer structural match, sorting the list of folds according to the raw scores from the initial round of optimization should provide some initial separation between the structural matches and non-matches. However, to achieve greater accuracy at separating matches from non-matches, direct optimization towards this goal was necessary. All 24 weights were further optimized using the method of Hooke & Jeeves.<sup>49</sup> The reciprocal weighted average ranking of the structural matches was used as the objective function. This function was useful because it is sensitive to small changes in ranking, and because improvements in ranking among low-rated sequence/fold pairs are given more weight than improvements in ranking among other pairs.

In order to limit the complexity of the method, an iterative procedure was used to eliminate some of the weights (setting them to zero). Starting from the initial set of  $N$  converged weights, each was set in turn to zero, and the remaining  $N - 1$  weights were re-optimized using the same objective function. The set of  $N - 1$  weights resulting in the lowest value of the objective function was retained as the initial set of weights from which to begin the next round of elimination. The best set of weights from each round was further optimized towards an objective function that we thought would be more relevant to the protein modeling community. For each sequence, folds are sorted according to the raw scores, and the total number of structural matches among the top five hits was calculated; this value was maximized as an objective function. We found that seven of the initial 24 weights could be eliminated without reducing this objective. After that, further elimination of weights resulted in a decrease in the metric. Therefore, the set of 17 weights that produced the maximum number of structural matches among the top five hits was used by the fold recognition jury. The final 17 weights are available from the authors upon request.

Raw scores produced by the jury are translated into estimated probabilities that a sequence/fold pair is a structural match using a method similar to that used to produce estimates of the alignment accuracy. A look-up table translating raw scores into structural match probabilities was created using the raw scores from the Defay/Cohen data set. Raw scores were grouped into 100 bins ranging from the minimum to maximum value of the score. To correct for sparse data, the width of each bin was allowed to expand until each contained at least ten data points. The probability of finding a structural match in each bin was then measured. Raw scores for new sequence/fold pairs are translated into approximate probabilities of being a structural match using the table.

### Genomic threading

ORFs from genomic data are annotated using a pipeline of programs. First, the BLAST program<sup>3</sup> is used to search for hits against pdbaa, the BLAST database of sequences from the current release of the Protein Data Bank<sup>52</sup> of solved structures. This is a very fast procedure, requiring several seconds of CPU time per ORF on a modern workstation (800 MHz Intel Pentium-III). Sequences for which BLAST produces a hit with an  $e$ -value of less than  $10^{-4}$  are annotated and excluded from further processing. This should correspond to an error rate of about 1 in 10,000 annotations.

The second round of searching uses the more sensitive tool, PSI-BLAST.<sup>4</sup> The PSI-BLAST program is run twice per ORF. In the first run, the nr database of non-redundant sequences is used in order to create a PSSM and gather multiple sequences. All default options (0.001 *e*-value cutoff for inclusion of a sequence in the matrix calculations, filtering turned on) were used, except that the maximum number of rounds was set to ten. In the second run, the PSSM from the first run is used to perform a search in the pdbaa database, with only a single round of searching. Any hits with *e*-values of less than  $10^{-4}$  are collected as annotations. Although this *e*-value implies an error rate of 1 in 10,000, a study of the true error rate of PSI-BLAST<sup>35</sup> found the error rate corresponding to *e*-values of  $10^{-4}$  to be higher, of the order of 1 in 100. The processing time for this second round of searching is more significant, requiring approximately ten minutes, on average per ORF on a standard desktop machine.

The set of aligned sequences gathered using PSI-BLAST in the second round is used as input to Pred2ary<sup>46</sup> to predict the secondary structure of the unknown sequence. Along with the PSSM obtained in the second round, this is sufficient data to apply the jury threading procedure using the fold library obtained from the ASTRAL database (described above). For folds with non-zero estimated probabilities of being a structural match, a sequence alignment is also calculated. The resulting probabilities and alignments are stored in a database for later retrieval and analysis. All possible structural matches with an estimated probability of greater than 99% are annotated.

---

## Acknowledgements

This work is supported by grants from the NIH (GM39900, F32-HG00200-03, and 1-P50-GM62412) and by the US Department of Energy under contract DE-AC03-76SF00098.

## References

- Bowie, J. U., Luthy, R. & Eisenberg, D. (1991). A method to identify protein sequences that fold into a known three-dimensional structure. *Science*, **253**, 164–170.
- Sander, C. & Schneider, R. (1991). Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins: Struct. Funct. Genet.* **9**, 56–68.
- Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. (1990). Basic local alignment search tool. *J. Mol. Biol.* **215**, 403–410.
- Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucl. Acids Res.* **25**, 3389–3402.
- Schaffer, A. A., Aravind, L., Madden, T. L., Shavirin, S., Spouge, J. L., Wolf, Y. I. *et al.* (2001). Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. *Nucl. Acids Res.* **29**, 2994–3005.
- Jones, D. T., Taylor, W. R. & Thornton, J. M. (1992). A new approach to protein fold recognition. *Nature*, **358**, 86–89.
- Jones, D. T. (1999). GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences. *J. Mol. Biol.* **287**, 797–815.
- Panchenko, A. R., Marchler-Bauer, A. & Bryant, S. H. (2000). Combination of threading potentials and sequence profiles improves fold recognition. *J. Mol. Biol.* **296**, 1319–1331.
- Kelley, L. A., MacCallum, R. M. & Sternberg, M. J. (2000). Enhanced genome annotation using structural profiles in the program 3D-PSSM. *J. Mol. Biol.* **299**, 499–520.
- Shan, Y., Wang, G. & Zhou, H. X. (2001). Fold recognition and accurate query-template alignment by a combination of PSI-BLAST and threading. *Proteins: Struct. Funct. Genet.* **42**, 23–37.
- Yona, G. & Levitt, M. (2002). Within the twilight zone: a sensitive profile-profile comparison tool based on information theory. *J. Mol. Biol.* **315**, 1257–1275.
- Teichmann, S. A., Chothia, C. & Gerstein, M. (1999). Advances in structural genomics. *Curr. Opin. Struct. Biol.* **9**, 390–399.
- Brenner, S. E. (2000). Target selection for structural genomics. *Nature Struct. Biol.* **7**, 967–969.
- Sali, A. & Blundell, T. L. (1993). Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.* **234**, 779–815.
- Sánchez, R. & Sali, A. (1997). Advances in comparative protein-structure modelling. *Curr. Opin. Struct. Biol.* **7**, 206–214.
- Sippl, M. (1990). Calculation of conformational ensembles from potentials of mean force. *J. Mol. Biol.* **213**, 859–883.
- Rost, B. & Sander, C. (1993). Prediction of protein secondary structure at better than 70% accuracy. *J. Mol. Biol.* **232**, 584–599.
- Chandonia, J. M. & Karplus, M. (1999). New methods for accurate prediction of protein secondary structure. *Proteins: Struct. Funct. Genet.* **35**, 293–306.
- Jones, D. T. (1999). GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences. *J. Mol. Biol.* **287**, 797–815.
- Fischer, D. & Eisenberg, D. (1996). Protein fold recognition using sequence-derived predictions. *Protein Sci.* **5**, 947–955.
- Defay, T. R. & Cohen, F. E. (1996). Multiple sequence information for threading algorithms. *J. Mol. Biol.* **262**, 314–323.
- Rost, B., Schneider, R. & Sander, C. (1997). Protein fold recognition by prediction-based threading. *J. Mol. Biol.* **270**, 471–480.
- Marchler-Bauer, A. & Bryant, S. H. (1997). Measures of threading specificity and accuracy. *Proteins: Struct. Funct. Genet.* **74**–82.
- Shi, J., Blundell, T. L. & Mizuguchi, K. (2001). FUGUE: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. *J. Mol. Biol.* **310**, 243–257.
- Thompson, J. D., Higgins, D. G. & 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. *Nucl. Acids Res.* **22**, 4673–4680.
- Murzin, A. G., Brenner, S. E., Hubbard, T. & Chothia, C. (1995). SCOP: a structural classification of proteins

- database for the investigation of sequences and structures. *J. Mol. Biol.* **247**, 536–540.
27. Bujnicki, J. M., Elofsson, A., Fischer, D. & Rychlewski, L. (2001). LiveBench-2: large-scale automated evaluation of protein structure prediction servers. *Proteins: Struct. Funct. Genet.* 184–191.
  28. Koh, I. Y., Eyrich, V. A., Marti-Renom, M. A., Przybylski, D., Madhusudhan, M. S., Eswar, N. *et al.* (2003). EVA: evaluation of protein structure prediction servers. *Nucl. Acids Res.* **31**, 3311–3315.
  29. Lundstrom, J., Rychlewski, L., Bujnicki, J. & Elofsson, A. (2001). Pcons: a neural-network-based consensus predictor that improves fold recognition. *Protein Sci.* **10**, 2354–2362.
  30. Ginalska, K., Elofsson, A., Fischer, D. & Rychlewski, L. (2003). 3D-Jury: a simple approach to improve protein structure predictions. *Bioinformatics*, **19**, 1015–1018.
  31. Orengo, C. A. & Taylor, W. R. (1996). SSAP: sequential structure alignment program for protein structure comparison. *Methods Enzymol.* **266**, 617–635.
  32. Siew, N., Elofsson, A., Rychlewski, L. & Fischer, D. (2000). MaxSub: an automated measure for the assessment of protein structure prediction quality. *Bioinformatics*, **16**, 776–785.
  33. Orengo, C. A., Michie, A. D., Jones, S., Jones, D. T., Swindells, M. B. & Thornton, J. M. (1997). CATH—a hierarchic classification of protein domain structures. *Structure*, **5**, 1093–1108.
  34. Fraser, C. M., Gocayne, J. D., White, O., Adams, M. D., Clayton, R. A., Fleischmann, R. D. *et al.* (1995). The minimal gene complement of *Mycoplasma genitalium*. *Science*, **270**, 397–403.
  35. Müller, A., MacCallum, R. M. & Sternberg, M. J. E. (1999). Benchmarking PSI-BLAST in genome annotation. *J. Mol. Biol.* **293**, 1257–1271.
  36. Kelley, L. A., MacCallum, R. M. & Sternberg, M. J. E. (2000). Enhanced genome annotation using structural profiles in the program 3D-PSSM. *J. Mol. Biol.* **299**, 499–520.
  37. Bucher, P. & Bairoch, A. (1994). A generalized profile syntax for biomolecular sequence motifs and its function in automatic sequence interpretation. In *Proc. Intl Conf. Intelligent Systems Mol. Biol.*, pp. 53–61, AAAI/MIT Press, Stanford, CA.
  38. Todd, A. E., Orengo, C. A. & Thornton, J. M. (2001). Evolution of function in protein superfamilies, from a structural perspective. *J. Mol. Biol.* **307**, 1113–1143.
  39. Adams, M. D., Celniker, S. E., Holt, R. A., Evans, C. A., Gocayne, J. D., Amanatides, P. G. *et al.* (2000). The genome sequence of *Drosophila melanogaster*. *Science*, **287**, 2185–2195.
  40. Venter, J. C., Adams, M. D., Myers, E. W., Li, P. W., Mural, R. J., Sutton, G. G. *et al.* (2001). The sequence of the human genome. *Science*, **291**, 1304–1351.
  41. Fischer, D. (1999). Rational structural genomics: affirmative action for ORFans and the growth in our structural knowledge. *Protein Eng.* **12**, 1029–1030.
  42. Voges, D., Zwickl, P. & Baumeister, W. (1999). The 26 S proteasome: a molecular machine designed for controlled proteolysis. *Annu. Rev. Biochem.* **68**, 1015–1068.
  43. Falicov, A. & Cohen, F. E. (1996). A surface of minimum area metric for the structural comparison of proteins. *J. Mol. Biol.* **258**, 871–892.
  44. Brenner, S. E., Koehl, P. & Levitt, M. (2000). The ASTRAL compendium for protein structure and sequence analysis. *Nucl. Acids Res.* **28**, 254–256.
  45. Chandonia, J. M., Walker, N. S., Lo Conte, L., Koehl, P., Levitt, M. & Brenner, S. E. (2002). ASTRAL compendium enhancements. *Nucl. Acids Res.* **30**, 260–263.
  46. Chandonia, J. M. & Karplus, M. (1999). New methods for accurate prediction of protein secondary structure. *Proteins: Struct. Funct. Genet.* **35**, 293–306.
  47. Needleman, S. B. & 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.
  48. Altschul, S. F. & Koonin, E. V. (1998). Iterated profile searches with PSI-BLAST—a tool for discovery in protein databases. *Trends Biochem. Sci.* **23**, 444–447.
  49. Hooke, R. & Jeeves, T. A. (1961). Direct search solution of numerical and statistical problems. *J. Assoc. Comput. Mach.* **8**, 212–229.
  50. Miyazawa, S. & Jernigan, R. L. (1985). Estimation of effective interresidue contact energies from protein crystal structures: quasi-chemical approximation. *Macromolecules*, **18**, 534–552.
  51. Flockner, H., Braxenthaler, M., Lackner, P., Jaritz, M., Ortner, M. & Sippl, M. J. (1995). Progress in fold recognition. *Proteins: Struct. Funct. Genet.* **23**, 376–386.
  52. Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H. *et al.* (2000). The Protein Data Bank. *Nucl. Acids Res.* **28**, 235–242.

Edited by M. Levitt

(Received 22 July 2003; accepted 24 July 2003)