Bootstrapping and Normalization for Enhanced Evaluations of Pairwise Sequence Comparison

RICHARD E. GREEN AND STEVEN E. BRENNER

Invited Paper

The exponentially growing library of known protein sequences represents molecules connected by an intricate network of evolutionary and functional relationships. To reveal these relationships, virtually every molecular biology experiment incorporates computational sequence analysis. The workhorse methods for this task make alignments between two sequences to measure their similarity. Informed use of these methods, such as NCBI BLAST [1], WU-BLAST [2], FASTA [3] and SSEARCH, requires understanding of their effectiveness. To permit informed sequence analysis, we have assessed the effectiveness of modern versions of these algorithms using the trusted relationships among ASTRAL [4] sequences in the Structural Classification of Proteins [5] database classification of protein structures [6]. We have reduced database representation artifacts through the use of a normalization method that addresses the uneven distribution of superfamily sizes. To allow for more meaningful and interpretable comparisons of results, we have implemented a bootstrapping procedure. We find that the most difficult pairwise relations to detect are those between members of larger superfamilies, and our test set is biased toward these. However, even when results are normalized, most distant evolutionary relationships elude detection.

Keywords*—*BLAST, bootstrap, FASTA, homology, normalization, sequence, sequence analysis, Structural Classification of Proteins (SCOP), substitution matrix.

I. INTRODUCTION

The explosive growth of biological sequence databases provides great opportunity for molecular and computational biologists. High-throughput sequencing projects have generated complete genome sequence for scores of microbes and several eukaryotes [7], including humans [8]. Biologists use these comprehensive data in their attempt to discover the biological functions of genes and the proteins they encode. For many proteins, it is possible to make inferences of function based simply on recognizable similarity with previously characterized sequences. Current technology allows between one-third to one-half of the genes within newly sequenced genomes to be annotated on the basis of recognizable sequence similarity to genes of other organisms [9]. Furthermore, as more genomes are sequenced and more genes are characterized, greater fractions of new genomes can be annotated in this way [10].

The ability to make useful inferences based on sequence similarity is based on the relationships between protein sequence, structure, and function—all of which revolve around homology (see Fig. 1). Homologous proteins are those that had a common evolutionary ancestor. The most common means of inferring homology is sequence comparison: experience has demonstrated that significant sequence similarity is a reliable indicator of homology. Because protein structure evolves very slowly, with cores being exceptionally well conserved over billions of years of evolution, homology between two proteins effectively guarantees that they will share similar structures [11]. It is generally believed that some similar protein structures have evolved independently, so structural similarity does not always signify evolutionary relatedness.

Two related proteins with a common ancestor may retain the same ancestral function, and thus play the same role. Similar functions have also evolved many times by convergence [12]. However, homology can provide sufficient clues about function to suggest experiments or inform hypotheses, allowing further characterization of unknown proteins. Sequence similarity detection is crucial in other aspects of computational molecular biology as well. For example, gene-finding, phylogeny reconstruction and analysis, pathway reconstruction, and homology structure modeling all depend heavily on the effectiveness and reliability of sequence comparison methods.

Many methods have been developed for detecting sequence similarity, reflecting the central role it plays in computational biology. Proper use and interpretation of the

Manuscript received April 30, 2002; revised September 8, 2002. This work was supported in part by NIH Grant 1 K22 HG00056 and Grant 5 T32 HG0047, in part by the Searle Scholars' Program (01-L-116), and in part by IBM.

The authors are with the Departments of Plant and Microbial Biology, and Molecular and Cell Biology, University of California, Berkeley, CA 94720-3102 USA (e-mail: brenner@compbio.berkeley.edu).

Digital Object Identifier 10.1109/JPROC.2002.805303

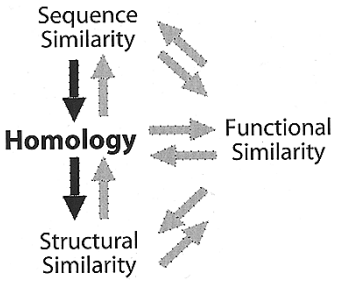


Fig. 1. Inferences from sequence similarity. Detectable similarity between two protein sequences implies a common origin, homology. This, in turn, implies a common three-dimensional structure. Other inferences are less reliable, indicated by lighter arrows.

results of these methods requires an understanding of the relative merits of each. Sequence-based similarity detection methods fall into two broad categories: pairwise and profile. Pairwise methods are those that take as input two single sequences and attempt to generate the optimal alignment between them. Searching a database of known sequences using a pairwise alignment method is a straightforward matter of generating alignments between the query sequence and each of the database sequences. Alignments with the best scores are then examined. Profile methods, on the other hand, generate a statistical model, or profile, of a sequence family and then compare the profile with a given sequence. Using a profile method, therefore, involves both constructing profiles and using them to detect similar sequences. Although profile methods have proven to be more sensitive than pairwise methods, their use requires prior knowledge of the sequence family in question—knowledge that typically derives from pairwise methods.

Sequence similarity detection using pairwise methods generally requires two steps, the first of which is generating the alignment between the sequences. Current pairwise-alignment algorithms for database searching are derivatives of the Needleman–Wunsch dynamic programming algorithm [13] as modified for local alignment by Smith and Waterman [14]. The Smith–Waterman algorithms guarantees the optimal alignment under a given scoring scheme, and the SSEARCH program [15] provides a full implementation. Heuristics that speed up pairwise alignment have been introduced in BLAST [1] and FASTA [3], the two most popular algorithms. WU-BLAST and NCBI BLAST are both implementations of the BLAST algorithm, differing in the way score statistics are generated, as well as some heuristics. For example, WU-BLAST implements and reports Karlin and Altschul sum statistics [16], [17] by default.

Alignments are generated using a scoring scheme that includes a substitution matrix and gap parameters. Substitution matrices for protein sequence alignments are 20×20 matrices that give scaled, log-odds scores for the pairing of any two aligned amino acid residues in an alignment [18]. The score of a given alignment is simply the sum of the matrix values for each position in the alignment, minus the penalty for gaps within the alignment. The optimal alignment is the one that generates the highest score in this way. For local alignments, this may not include all of either sequence.

The second step in pairwise similarity detection is generating a statistical score for the alignment. It has been shown analytically for ungapped alignments [18], [19] and empirically for gapped [20]–[22] alignments that optimal alignment scores follow an extreme value distribution (EVD). Therefore, generating a statistical significance score for an alignment is really a problem of finding appropriate EVD parameters for the raw score in question. The BLAST programs have precomputed EVD data for several sets of scoring parameters based on large-scale computational experiments with simulated data [23]. The FASTA package programs (FASTA and SSEARCH), by default, generate empirical EVD parameters for a given alignment by curve-fitting the distribution of alignment scores generated during the database search in question [24]. By either method, once the EVD parameters are derived, an E value can be generated that represents the significance of the alignment in the context in which it was generated [25]. Statistical scores have proven to be far superior to other measures of alignment quality [6], [24].

II. Methodology

Because the primary aim of similarity search methods is homologue detection, they are typically evaluated by their ability to do this effectively. Homologue detection always requires a balance between sensitivity and specificity. Sensitivity is defined here as the ability to identify the homologues of a given sequence within a database of homologous and nonhomologous sequences (true positive detection). Specificity, by comparison, is the ability to exclude nonhomologues from the list of real homologues (false positive exclusion). The tradeoff between sensitivity and specificity is a consideration for all similarity search methods, since any set of inputs will generate a score. The most powerful methods assign good scores only to real homologues and bad scores only to nonhomologues. Because the number of nonhomologues will typically be vastly greater than the number of homologues in a given database search, specificity is especially important.

A. Constructing the Evaluation Databases

To evaluate the sensitivity and specificity of a sequence comparison method, it is necessary to construct a test dataset of sequences whose evolutionary relationships are known. Classifications in existing databases, such as the Protein Information Resource (PIR) database [26], have been used for this purpose [27], [28]. Custom datasets, such as the Aravind set, have also been expressly derived for evaluating similarity detection methods [29], [30]. Evaluations of new substitution matrices or other scoring parameters have made use of an even wider array of test sets [31], [32]. The power of a given similarity search method is then assessed by its ability to predict known relations while avoiding spurious matches. Naturally, the knowledge of which sequences are related should be derived independently of the method being evaluated. Because a large percentage of sequence database annotation de-

rives from sequence similarity detection, it is not desirable to use this annotation as the basis for constructing evaluation databases. Such resources would not include the truly homologous sequences that have yet to be correctly annotated. Additionally, the evaluation will be polluted with the false annotations that currently corrupt databases [33], [34]. Consequently, using sequence-based classifications for evaluation leads to a circularity, and their use tests consistency with existing methods rather than absolute accuracy.

A solution to this problem is to use structure as a means of inferring evolutionary relationships between pairs of proteins. Because structure evolves more slowly than sequence, structural similarity can be used as a "gold-standard" for determining whether any two sequences are related. To this end, analyses frequently use the classifications in the Structural Classification of Proteins (SCOP) [5], [6], [35]–[37] and CATH [38]–[40] databases, as well as direct structural similarity [41].

The SCOP database provides a hierarchical classification of the structural domains of all solved protein structures. Domains are classified at the level of class, fold, superfamily, and family. ASTRAL [4] provides sequence sets of SCOP domains, filtered at various levels of identity. These domain sequences, along with their SCOP classification information, can be used as test sequences for any similarity detection method, since their relationships are known.

Protein domains are the unit of classification within SCOP, and by extension, ASTRAL, because these are the fundamental units of protein evolution and structure. Using domain sequences, rather than whole proteins, allows us to unequivocally identify which domains are involved in any pairwise alignment. Such identification can be difficult when using multidomain sequences, or sequences whose domain organization is unknown. An unfortunate consequence of using isolated domain sequences is that more global methods and parameters may be favored. Each domain sequence is a complete structural and evolutionary unit, so homologous pairs will have similar lengths with meaningful alignments over their entire lengths. By contrast, most typical database queries require identification of regions of similarity within sequence pairs that have both related and unrelated regions.

Within the SCOP hierarchy (see Fig. 2), it is widely acknowledged that domains of the same *superfamily* are descendants of a common ancestor. Domains of different folds are believed to be evolutionarily unrelated. Domains of the same fold but different superfamily currently lack evidence of homology. If such evidence eventually becomes available, superfamilies can be coalesced to reflect this new understanding. We evaluated similarity detection methods and scoring parameter sets by their ability to generate good scores for all the truly homologous sequences (i.e., those within the same superfamily) while simultaneously generating poor scores for all sequences of different folds. Domains classified in the same fold but different superfamilies are treated as undetermined and not considered in our benchmarking.

Our test databases (see Fig. 3) were constructed from the genetic domain sequences within the ASTRAL database (file

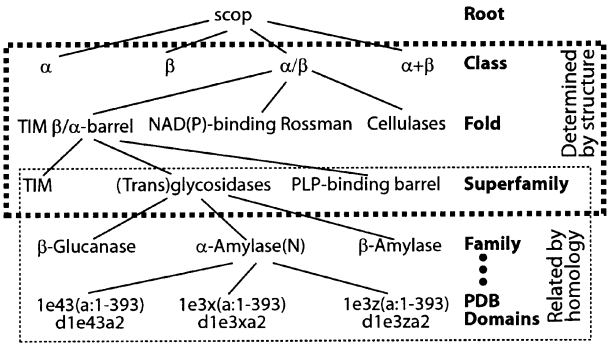


Fig. 2. SCOP hierarchy sample. The two top levels of SCOP, class and fold, are purely based on structural similarity. Domains of the same superfamily rely on common structure and other features as evidence of homology. The superfamily level and all those below reflect homology. The superfamily level is unique in being based on structural information and indicating homology.

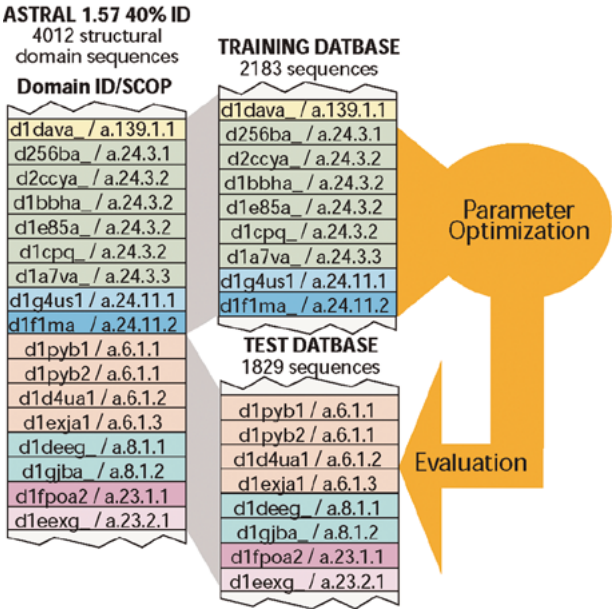


Fig. 3. Training and test databases. The ASTRAL 1.57 database, filtered at 40% sequence identity, was partitioned into training and test databases. Partitioning was done at the level of fold. Parameter optimization on the training database was followed by evaluation on the test database.

astral-scopdom-seqres-gd-sel-gs-bib-40-1.57.fa) based on SCOP release 1.57. We used the set filtered at 40% sequence identity to make the test specific for remote homologue detection, as sequences with greater than 40% sequence identity are easily identifiable as similar [42]. After masking low-complexity regions with SEG [43] (using parameters -w 12, -t 1.8, and -e 2.0), we partitioned this database into two similarly sized databases. Each contained all sequences of every-other fold; there are no sequences in the intersection of the two sets. One dataset, with 2183 sequences, was then used as a training database to determine optimal search parameters (substitution matrix, matrix scaling, and gap penalties) for each of the pairwise search methods. Hereafter, it will be referred to as the training database. The other

database, with 1829 sequences, was used as the test database for each of the pairwise search methods with optimal parameter sets, and will be referred to as the test database. Separating the original ASTRAL set in this way ensures that we do not simply evaluate a particular algorithm's ability to be optimized for the database in question. All datasets are available at http://compbio.berkeley.edu.

An additional database, SNR, was generated for speed evaluation. This 813 418-member database is the union of the SWISS-PROT [44], TrEMBL and TrEMBL-NEW databases of May 28, 2002. This database was masked of trans-membrane helices using TMHMM [45], low complexity regions using SEG [45], and coiled-coil regions using CCP [46].

B. Summarizing Database Homologue Detection Search Results: The CVE Plot

As mentioned previously, the ability of a similarity detection method to report truly homologous sequence matches must be balanced against its ability to refrain from reporting matches between unrelated sequences. This sensitivity versus specificity tradeoff can be rendered graphically by the coverage versus errors per query (CVE) plot (ROC) [6]. CVE plots are related to receiver operating characteristic plots [6], [47], [48] and SPEC-SENS [49], [50] curves, but present the data in a way that is directly interpretable and germane to sequence analysis. A CVE plot is generated by performing a database-versus-database search and ordering the results by significance score (see Fig. 4). Then, we used the SCOP classification information to determine whether each reported match pair was homologous, nonhomologous, or undetermined. At each significance threshold, from highest to lowest, a point on the CVE plot is generated. The x-coordinate of the point is coverage; that is, the number of detected homologue pairs divided by the total number of pairs that exist in the database (true positives/number of homologue pairs). The y-coordinate of the point is errors per query (EPQ), namely the number of nonhomologue pairs reported divided by the size of the query database (false positives/number of queries). The CVE results generated from a perfect homologue detection method would be a single point at the lower right-hand corner (see Fig. 4).

There are benefits of depicting the error rate in this way that allow analyses not possible by other methods. First, EPQ rates are comparable between experiments, even when the databases are not the same. This is because the distribution of false positive scores from a database search is largely independent of the particular database searched. Also, using EPQ allows the direct evaluation of significance scoring schemes (such as E values) because EPQ and significance scores share the same scale. EPQ reports the number of false positives observed per database query whereas significance scores report the number of false positives expected per database query. The EPQ axis in a CVE plot is log-scaled to show performance over a wide error range. This allows consideration of performance at very low error rates.

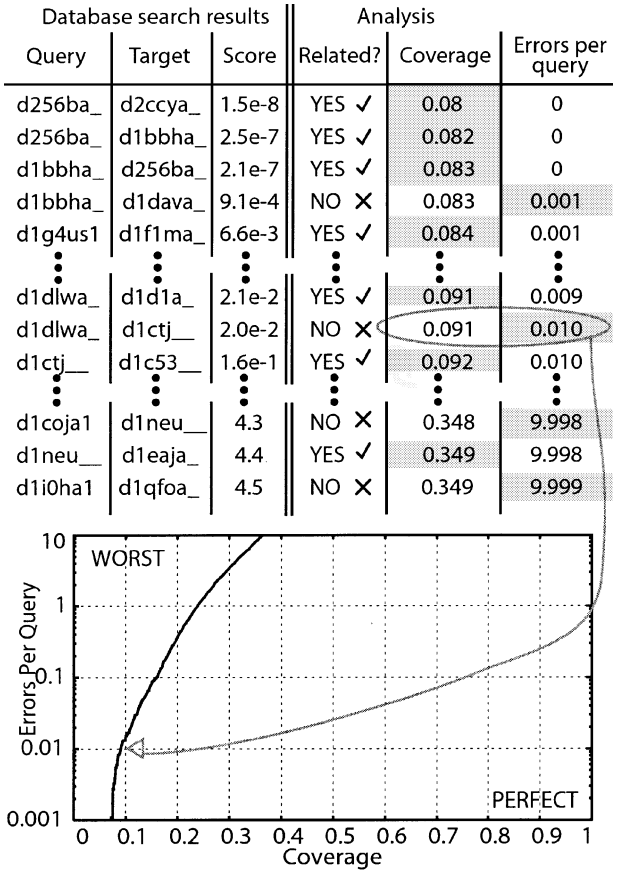


Fig. 4. Generating CVE plots. Results of a database-versus-database search are ordered by significance (columns 1–3). Using SCOP, each match is classified as having identified related sequences, unrelated sequences, or sequences whose relationship is not known. If the matched sequences are related, the coverage is increased. If the matched sequences are not related, then an error was made and the EPQ increases. A point on the CVE plot is generated for each significance level in the list, from most significant to least significant. Note that the significance scores themselves are not shown on the CVE plot. A perfect similarity detection method would correctly identify all relations within the database before making any errors. This would be represented as a single point in the lower right-hand corner of the CVE plot.

C. Superfamily Size Normalization

On CVE plots, the 100% coverage level is defined by the number of homologous relations between members of all superfamilies. The number of these relations within a given superfamily grows quadratically with superfamily membership size. Therefore, any representational biases present within the database are exacerbated, and large families dominate the overall results. There are well-known biases within the database of solved structures and, by extension, within SCOP and ASTRAL. Proteins that are more amenable to structure determination or are deemed more interesting research subjects are over-represented. Because of this bias, performance evaluation may be skewed to favor those methods that detect similarity between members of the larger superfamilies.

We took two approaches to neutralizing this effect (see Fig. 5). To each correctly identified relation, both approaches assign a weight that is a function of the size of the superfamily in which it occurs. Under quadratic

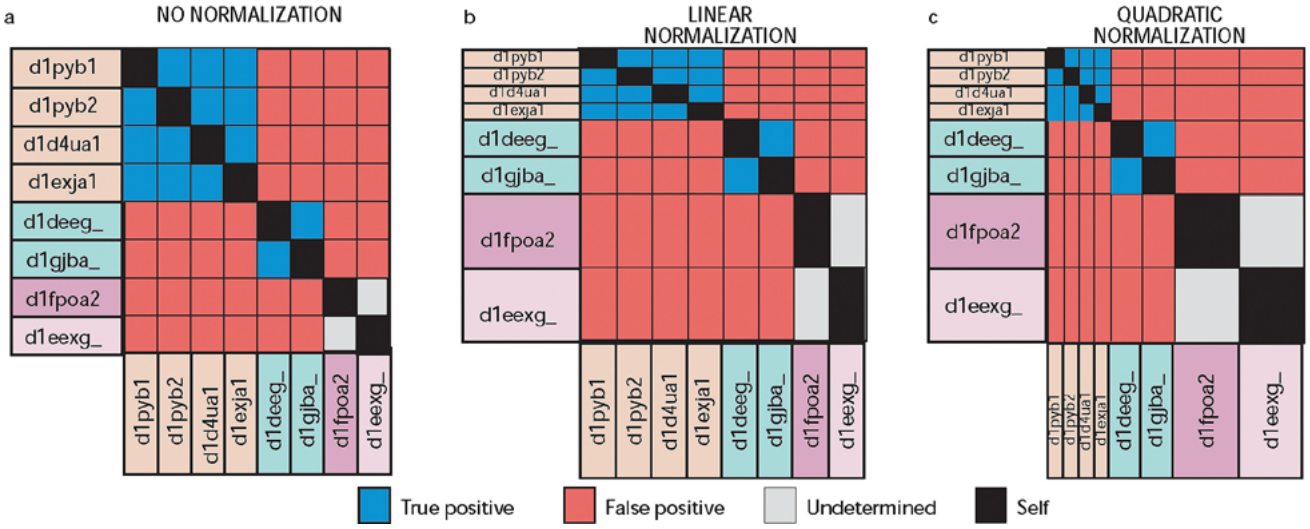


Fig. 5. Normalization schemes. For each normalization scheme, the size of each matrix element represents the weight given to each relationship. (a) The number of correct superfamily level relations, shown in blue, is naturally dominated by large superfamilies. (b) Linear normalization weights each superfamily in linear proportion to the number of sequences it contains. (c) Under quadratic normalization, each superfamily is weighted equally.

normalization each correct pairwise relation identified is weighted by $1/(n^2 - n)$, where n is the number of the sequences within that superfamily, because $n^2 - n$ is the number of relations within each superfamily. Therefore, quadratic normalization weights all superfamilies equally, regardless of size. Under quadratic normalization, the maximum achievable "coverage" is the number of superfamilies in the test database, and the quadratically normalized CVE plots presented reflect this fact.

Linear normalization is a compromise between no normalization and quadratic normalization. Linear normalization is motivated by the fact that sequence superfamilies are not, in fact, represented equally in nature. Furthermore, the representational bias within our test databases reflects, at least to some degree, the unequal representation within the sequence superfamilies found in nature. Therefore, the results generated by larger superfamilies should carry more weight, but not necessarily quadratically more weight, than those from smaller superfamilies. In this normalization scheme, each superfamily is weighted in linear proportion to its size. Each correctly identified pairwise relation is weighted by $1/(n - 1)$. Therefore, the maximum achievable "coverage" is the number of sequences within the test database, and the linearly normalized CVE plots presented reflect this.

D. Bootstrapping Provides Significance of Coverage Versus Error

The CVE line for any two search method/parameter set pairs will likely differ. Therefore, to determine which method is superior at a given error rate, it is a straightforward matter to pick a suitable error rate and rank methods by the coverage generated. However, the significance of any difference between two coverage levels is not immediately apparent. To address the question of performance difference significance, we implemented the bootstrap strategy described in Fig. 6. In brief, the database in question was sampled randomly with replacement n times, where n is the number of sequences in the database. This sampling produces a new bootstrap database in which each sequence is represented zero, one, or more than one times.

CVE data were generated for each bootstrap database using significance scores from the original database search. Repeating the bootstrap procedure many times gives a distribution of the CVE statistic that can be used to reliably estimate its standard error [51]. The results of any two search methods can then be compared in a more meaningful way by analyzing the Z score produced by a two-sample parametric means test using this bootstrapped standard error.

Two bootstrap distributions can also be compared by sampling from them both and computing the fraction of times in which one sample generates higher coverage than the other. The bootstrap overlap fractions given in Figs. 8, 10, and 11 are generated by this method, sampling from the relevant bootstrap distributions 1000 times. Equivalent distributions would be expected to yield a score of 0.5 by this method.

E. Similarity Search Methods Evaluated

We set out to evaluate several of the most commonly used pairwise search tools (see Table 1). All were downloaded from the source given in Table 1, compiled and installed per documented instructions, and run on Linux systems using default options, except where otherwise noted.

III. Results

A. Parameter Optimization and Pairwise Method Evaluation

To conduct as unbiased a test as possible, we partitioned our test database into two nonoverlapping databases (see Fig. 3). Each pairwise method was then evaluated on the

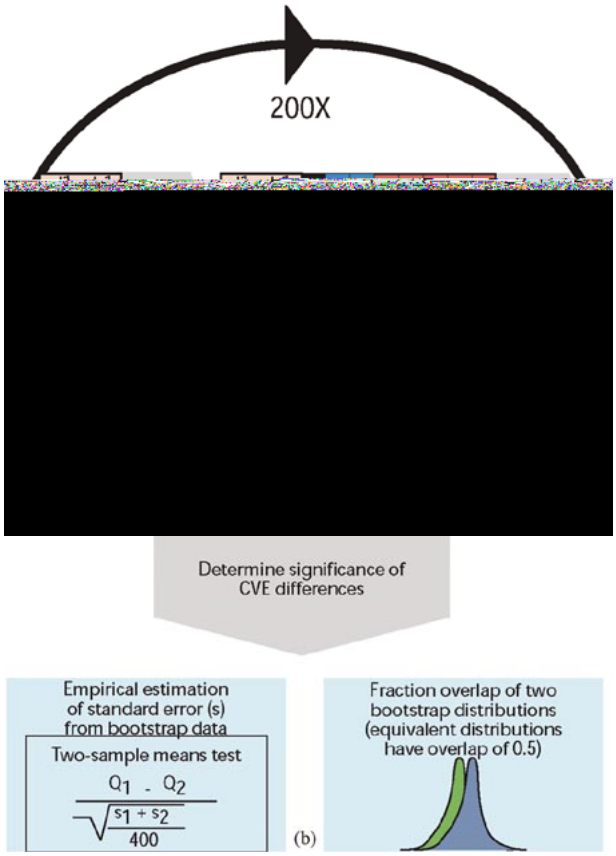


Fig. 6. Bootstrap procedure. (a) The database is sampled with replacement a number of times equal to the number of sequences it originally contains. This generates a bootstrapped database with some sequences left out and others repeated. CVE statistics are generated for each round of bootstrapping. After 200 rounds, a coverage distribution can be obtained for any EPQ rate. The distribution can be used to determine the significance between coverage levels generated by any two pairwise methods or parameter sets. (b) Significance testing is done two ways. A two-sample parametric means test using the standard errors from the two bootstrap distributions is used to generate a Z score. Q_n is the coverage at 0.01 EPQ, and s_n is the sample standard deviation at 0.01 EPQ. An alternative comparison method is to draw 1000 random samples from each bootstrap distribution. The fraction in which the coverage of the first sample is greater than the second is reported. Under this procedure, equivalent methods would score 0.5.

training database using a range of substitution matrices and gap parameters (see Table 2). The training phase involved the evaluation of the results of more than 8 billion pairwise comparisons within 1846 database-versus-database comparisons. The results of these searches were compiled and used to generate CVE plots and statistics. For each pairwise search method, we further evaluated the parameter set that generated the highest coverage at 0.01 EPQ under linear normalization. These optimal parameter sets are given in Table 3. Note that for NCBI BLAST and WU-BLAST, the number of matrices and gap-parameter combinations evaluated was smaller than for SSEARCH and FASTA. A consequence of the statistical scoring scheme employed by NCBI BLAST and WU-BLAST is that only a limited set of matrices have associated scaling parameters precomputed [23]. For other matrix/gap-parameter combinations, the statistical scores are less reliable and therefore left unevaluated in the present study. It is worth noting that the parameter set that yields the highest coverage at 0.01 EPQ may not necessarily be best at other error rates. However, in all cases, the top scoring parameter set at 0.01 EPQ was among the best at EPQ rates in the range of 0.001 to 10. Complete results from the training phase are available at http://compbio.berkeley.edu.

To determine the significance of the performance differences between each of the four pairwise search methods, we performed database-versus-database searches using the test database and the optimal parameter sets listed in Table 2. We present results as CVE plots in Fig. 7(a) in unnormalized, linearly normalized, and quadratically normalized format. It is interesting that when the results are normalized for superfamily size, the coverage invariably increases. This indicates that for any of these methods it is more difficult to detect the relations within larger superfamilies, and de-emphasizing larger superfamilies increases coverage. The SSEARCH algorithm, which fully explores the alignment space, finds the most relationships at most error rates, as expected. The popular NCBI BLAST finds the fewest. The relative order of performance between these four methods remains unchanged under each normalization scheme, with one exception. In the 1–10 EPQ range, WU-BLAST outperforms SSEARCH when results are quadratically normalized.

Fig. 7(b)–(d) shows the bootstrap distribution of CVE results under each normalization scheme for SSEARCH with optimal parameters. The inset of each CVE plot shows the coverage distribution for these 200 bootstrap samples at 0.01 EPQ. As expected, the histogram for this distribution closely resembles the parameterized Gaussian distribution generated by maximum-likelihood (ML) estimation of the mean and standard deviation computed directly from the bootstrap data. The original CVE line—that is, the SSEARCH CVE line in Fig. 7(a)—is also shown for comparison. This line corresponds to sampling each sequence once and only once

An interesting consequence of bootstrapping the underlying data is that the original CVE line invariably falls toward the higher coverage end of the bootstrap distribution in normalized results. We determined that this occurs because during bootstrap sampling, by chance, some of the smaller superfamilies are not sampled or sampled only once. When this happens, no relations remain for that superfamily. Since the easier relations to detect are primarily within the smaller superfamilies, the effect of eliminating them will be felt more emphatically when results are normalized by superfamily size. As a consequence of this artifact, the bootstrap average of coverage at a given error rate is not in agreement with the coverage at the same error rate in the underlying data [Original CVE line in Fig. 7(b)–(d)]. However, since we are interested in using bootstrapping to generate a measure of the standard error (not the mean), we have disregarded this artifact and used the bootstrap standard error in conjunction with the original coverage at 0.01 EPQ for bootstrap significance testing.

We bootstrap sampled the CVE data from the pairwise alignment results 200 times. The bootstrap coverage distri-

Table 1
Pairwise Methods Evaluated

Method Evaluated	Version	Location
NCBI BLAST	2.2.1	ftp://ftp.ncbi.nih.gov/blast
WU-BLAST	2.0MP-WashU[15-Apr-2002]	http://blast.wustl.edu
FASTA3	3.4t10	http://fasta.bioch.virginia.edu
SSEARCH3	3.4t06	http://fasta.bioch.virginia.edu

The version number and download source for each program is also given.

Table 2
Substitution Matrix and Gap-Parameter Space Explored

PAIRWISE METHOD	MATRIX SPACE	GAP PARAMETER SPACE (OPEN/EXTENSION PENALTIES)
NCBI BLAST and WU-BLAST	PAM30	8-10/1, 5-7/2
	PAM70	9-11/1, 6-8/2
	PAM250	17-21/1, 13-17/2, 11-15/3
	BLOSUM45	16-19/1, 12-16/2, 10-13/3
	BLOSUM50	15-19/1, 12-16/2, 9-13/3
	BLOSUM62	9-13/1, 6-11/2
	BLOSUM80	9-11/1, 6-13/2
	BLOSUM90	9-11/1, 6-9/2
FASTA and SSEARCH	BC1020, BC1030, BC0020, BC0030	9-19/1-3 for all
	PAM30-PAM150	
	VTML160, VTML250	
	BLOSUM40, 45, 50, 55, 60, 62, 65, 70, 80	
	PAM50, 60, …, 150	

We ran each method using the matrices and gap-parameter sets shown. A dash between two numbers indicates a range of gap parameters. For example, 9–19/1–3 indicates the 33 different combinations of gap opening penalties 9, 10, … 19 and gap extension penalties 1, 2, and 3.

Table 3
Optimal Matrix and Gap-Parameter Combinations

PAIRWISE METHOD	OPTIMUM MATRIX	OPTIMUM GAP PARAMTERS (OPEN/EXTENSION)
NCBI BLAST	BLOSUM50	10 / 2
WU-BLAST	BLOSUM50	10 / 2
FASTA	VTML160	14 / 2
SSEARCH	VTML160	14 / 2

The combination of substitution matrix and gap parameters shown is that which generated the highest coverage under linear normalization at 0.01 EPQ for each pairwise method.

bution at 0.01 EPQ from each method is compared in Fig. 8. SSEARCH outperforms the heuristic methods under each normalization scheme, and the difference is significant. Interestingly, the differences between each method do not vary much under the various normalization schemes, indicating that the large superfamily bias affects each method roughly equally. Furthermore, the fraction overlap between bootstrap distributions shows good agreement with the Z score (see Fig. 8).

We examined the size distribution of superfamilies within our test and training dataset to further investigate the correlation between superfamily size and ability to detect remote homologues. Fig. 9(a) shows the distributions of superfamilies, by size, within both training and test databases. Note that overwhelmingly the most common superfamily size is one. These superfamilies are important in that there are no relations to detect within them. Therefore, they serve only as decoys within these experiments, i.e., they can contribute to the errors but not to the coverage. Note also the presence of several very large superfamilies (Immunoglobulins, P-loop NTP hydrolases, etc.). The largest superfamily within the test database is the NAD(P)-binding Rossmann-fold, containing 76 members. Within this superfamily, there are 5700 relations, as compared with the 192 relations within all the 96 superfamilies of size 2 in the test database.

To further investigate the general effect of apparently poor homologue detection within larger superfamilies, we broke down the results of the pairwise-test database search using SSEARCH with optimal parameters [see Fig. 9(b)]. As expected from the normalization trend, there is a general negative correlation between superfamily size and percentage of relationships identified. However, there are several exceptions to this trend. Both the C-type lectin-like superfamily and the N-terminal nucleophile aminohydrolases are superfamilies whose relations are more detectable than others of their size.

B. Statistical Significance of Gap-Parameter Optimization

To determine the significance of choosing optimal gap parameters, we ran SSEARCH using the BLOSUM50 substitution matrix on the test database under a range of gap parameters around the optimum (14 gap opening penalty and 1 gap extension penalty) previously found for this matrix found on the training dataset. We generated CVE plots for each run [see Fig. 10(a)]. There appears to be a range around the optimum in which the coverage does not vary widely.

Next, 200 independent bootstrap samples were generated from the results of each parameter set for significance testing. Means-test Z scores and bootstrap distribution overlap statistics were generated for each gap-parameter set tested against

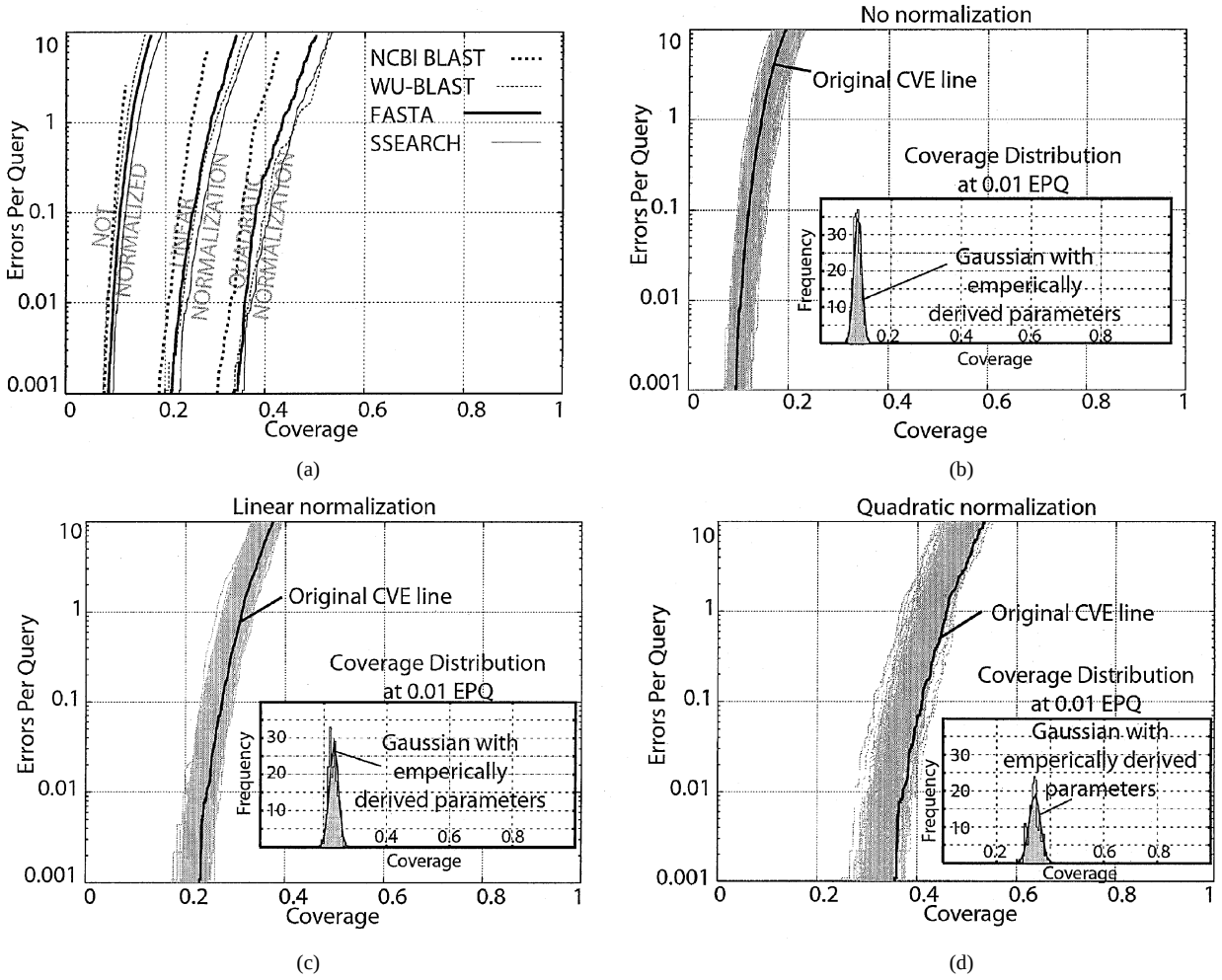


Fig. 7. Pairwise methods comparison and bootstrap sampling. (a) Each of the four pairwise methods, under optimal search parameters, was used to search the test database, generating CVE plots. (b)–(d) CVE lines of 200 bootstrap samples of the SSEARCH results are shown under each normalization scheme. The original CVE line is that shown in (a). The inset histograms show of the frequency of each coverage level at 0.01 EPQ. Superimposed on each histogram is the Gaussian distribution obtained by calculating ML parameters (mean and standard deviation) given the bootstrap data.

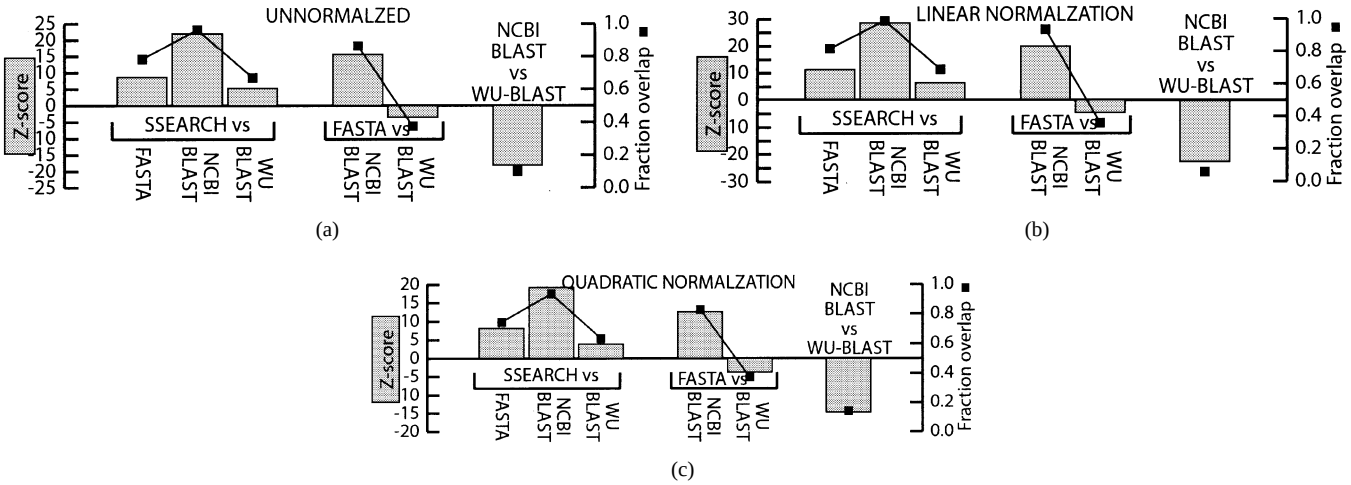
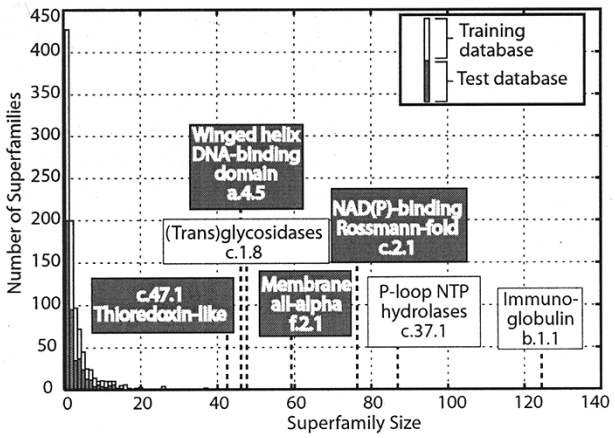
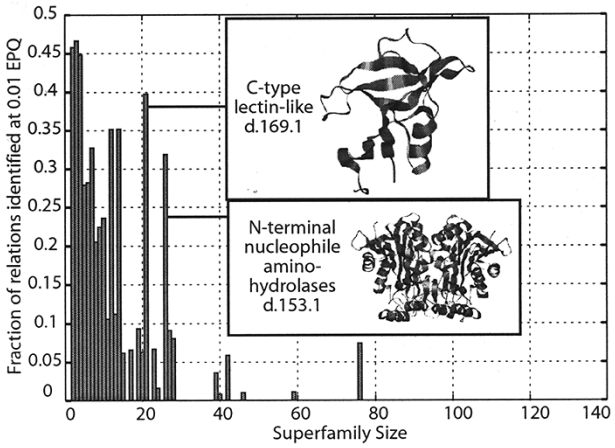


Fig. 8. Significance test of pairwise methods performance differences. (a)–(c) Each pairwise method result was bootstrapped 200 times. Means-test Z scores are given along the left axis and fraction of distribution overlap is given along the right axis. Results are reported under each normalization scheme.



(a)



(b)

Fig. 9. Performance versus superfamily sizes within ASTRAL 1.57. (a) ASTRAL 1.57 at 40% identity was divided into training and test databases. The frequency of superfamilies of each size is shown for each database. (b) The test database was searched using SSEARCH with optimal parameters. The fraction of correctly identified relations at 0.01 EPQ is shown versus superfamily size.

the optimal set, 14/1 [see Fig. 10(b)–(d)]. Each normalization scheme gives similar results: there is a range around the optimum in which one can safely perform database searches without significantly affecting the performance of the search. Using a Z score cutoff of ± 2, this range is 13–15 for the gap opening penalty (with 1 for gap extension penalty) and 1 to 2 for the gap extension penalty (with 14 for opening penalty) for unnormalized results. When results are linearly normalized, this range increases to 13–18 for the gap opening penalty, and 1–3 for the gap extension penalty. Therefore, it is not critical to perfectly optimize gap scores for a given database search. Furthermore, the default substitution matrix/gap-parameter combination for each of the four pairwise methods is optimal or not significantly different than optimal (data not shown).

C. Substitution Matrix Evaluation

Substitution matrices have been developed based on amino acid chemico-physical characteristics [52], genetic code distance [53], and observed evolutionary patterns [54]–[56], and these have been evaluated extensively [31], [57]. The most commonly used matrices are of the BLOSUM [56] and PAM [55] families, both of which are derived from observed patterns of amino acid substitution within real protein sequences. The BLOSUM family is derived from short, ungapped alignments between pairs of sequences grouped by varying percent identities. Because these matrices are computed directly from observed residue exchange data, they make no assumptions about how biological sequence evolution actually happens. This is a fundamental difference between the BLOSUM and the PAM families. All matrices of the PAM family are derived from the PAM1 matrix, which is computed from the observed exchange rate within alignments of real sequences chosen to model a 1% sequence identity difference (1% point accepted mutation). Further distance matrices are then computed by assuming a Markov chain evolutionary model and extrapolating PAM1 to various distances.

For comparison, we also evaluated two newer substitution matrix families. The Blake–Cohen (BC) matrix set [39] is derived from structural alignments of remote homologues, and the VTML family [58] is based on an empirically determined substitution rate-matrix that extends from the Dayhoff evolutionary model.

We employed our family normalization and bootstrapping proceduresto measure the performance differences between families of subsititution matrices. Each substitution matrix family contains several matrices, each scaled to model protein sequence evolution over a given amount of evolutionary change. The first step in this experiment was determining the optimum matrix scale and gap-parameter set for each of the four matrix families tested. This was done using SSEARCH and the training database. The results are shown in Table 4. We performed this analysis with SSEARCH for two reasons. First, it guarantees the optimal alignment under a given scoring scheme. For this reason, matrices that perform best using SSEARCH can be said to best embody the host of factors that affect real biological sequence evolution without artifacts of heuristic methods. Second, the default statistical scoring implemented in SSEARCH estimates scaling parameters by fitting an EVD curve to the observed alignment scores. Therefore, statistical significance scores are automatically calculated for any combination of substitution matrices and gap parameters. In this way, the FASTA package tools, including SSEARCH, are more amenable to such analyzes. However, the results shown in Table 4 are not universally applicable. The heuristics implemented in BLAST, for example, may affect matrix performance.

Fig. 11(a) shows CVE plots for the best performing parameter set for matrices using SSEARCH. Under all three normalization schemes, there is very little difference between VTML and BLOSUM. Both of these matrices appear to be superior to PAM and BC for remote homologue detection.

To determine the significance of the coverage differences generated by each substitution matrix, we did bootstrap analysis on each. As before, 200 bootstrap samples of each set of results were used to generate a coverage distribution at 0.01 EPQ. Fig. 11(b)–(d) shows the Z scores and fraction

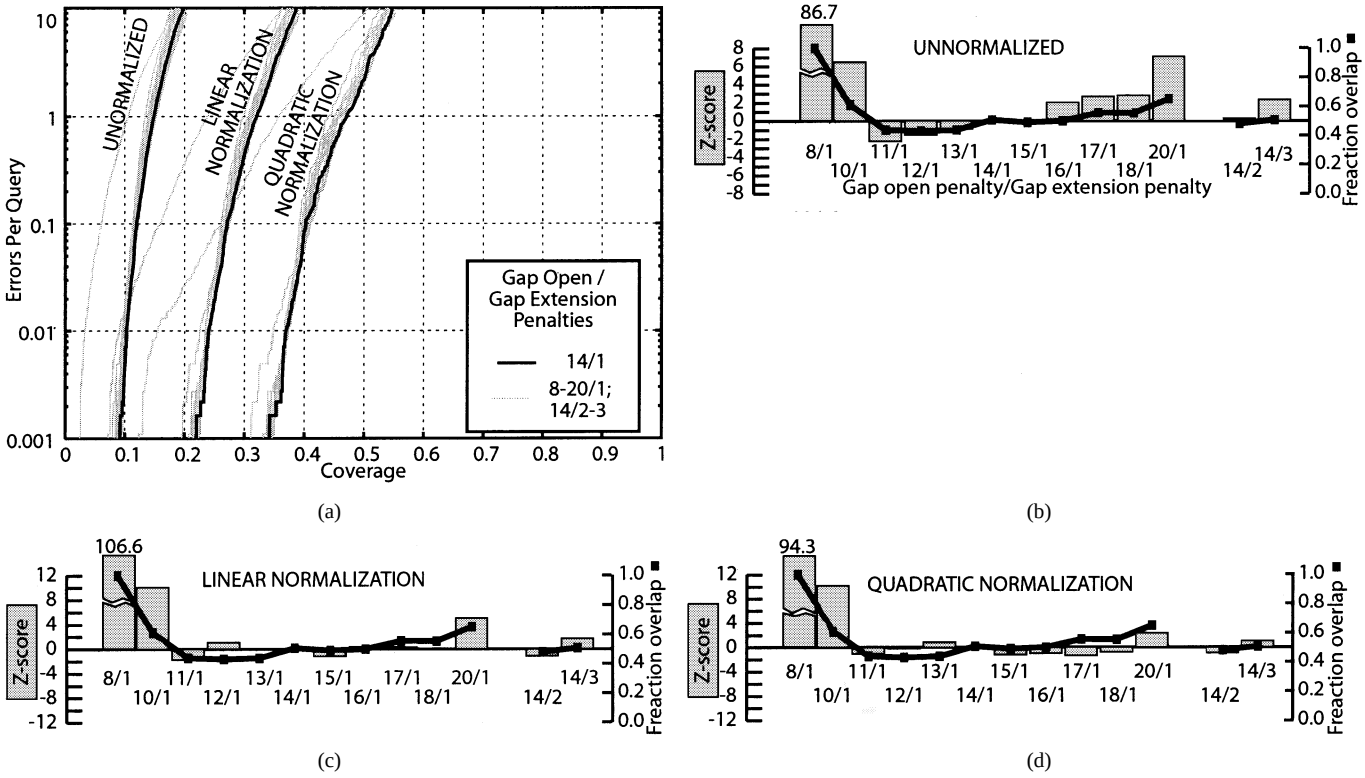


Fig. 10. Gap-parameter performance. SSEARCH was used, with the VTML160 matrix and gap parameters around the optimum (14/1). (a) CVE plots were generated under each normalization scheme. The test used gap open parameters 8, 9, … 20 with extension parameter 1, and also used gap open parameter 14 with extension parameters 2 and 3. (b)–(d) Bootstrapping was done (200 samples) for each and Z score and fraction overlap values were generated for each parameter set versus the optimum set under each normalization scheme.

Table 4
Optimum Matrix Scales and Gap Parameters

MATRIX	OPTIMUM SCALE	OPTIMUM GAP PARAMTERS (OPEN/EXTENSION)
BLOSUM	BLOSUM50	14 / 1
VTML	VTML160	14 / 2
PAM	PAM140	9 / 1
BC	BC1020	19 / 3

A range of matrix and gap parameters was evaluated using SSEARCH and the training database. Those reported here generated the highest coverage under linear normalization at 0.01 EPQ

of bootstrap distribution overlap for each pairwise combination of matrix results. Confirming the results suggested in Fig. 11(a), BLOSUM and VTML perform nearly indistinguishably under all normalization schemes. Both are significantly superior to PAM and BC for remote homologue detection. This is perhaps not surprising, since the superiority of BLOSUM over PAM for remote homologue detection has been established [31]. Furthermore, the BC matrices were developed primarily for generating accurate alignments, not for remote homologue detection.

D. Statistical Score Evaluation

In addition to being able to differentiate between related and nonrelated sequences, similarity detection methods should also give the user a reliable estimate of the significance of any similarity detected. This is especially important when a newly discovered sequence is used as a query and the user cannot be sure that it has any homologues within the search database. Each of the pairwise methods evaluated is capable of generating E value statistical significance scores. E value may be interpreted as the number of matched pairs one would expect by random chance that are as good as or better than the one reported, given the database search performed to find it.

To determine the reliability of the E value significance scores generated by each sequence analysis method, we further analyzed the results of the database searches performed by each method using optimized search parameters. For each incorrectly identified relationship (false positive) we plotted the E value at which it was reported. One should expect to find, for example, one false positive per database query by E value 1. The results of this experiment are shown in Fig. 12. The E values generated by SSEARCH are remarkably close to the ideal line. This suggests that estimating EVD parameters by fitting to the actual database query score distribution is an ideal method. Furthermore, as SSEARCH and FASTA use the same method of calculating significance scores, it appears that the FASTA heuristics remove false positives. This shifts the database score distribution and the EVD parameters, leading to conservative scores.

At higher error ranges, each method nears the idealized score. This beautiful statistical result may be, in part, due to

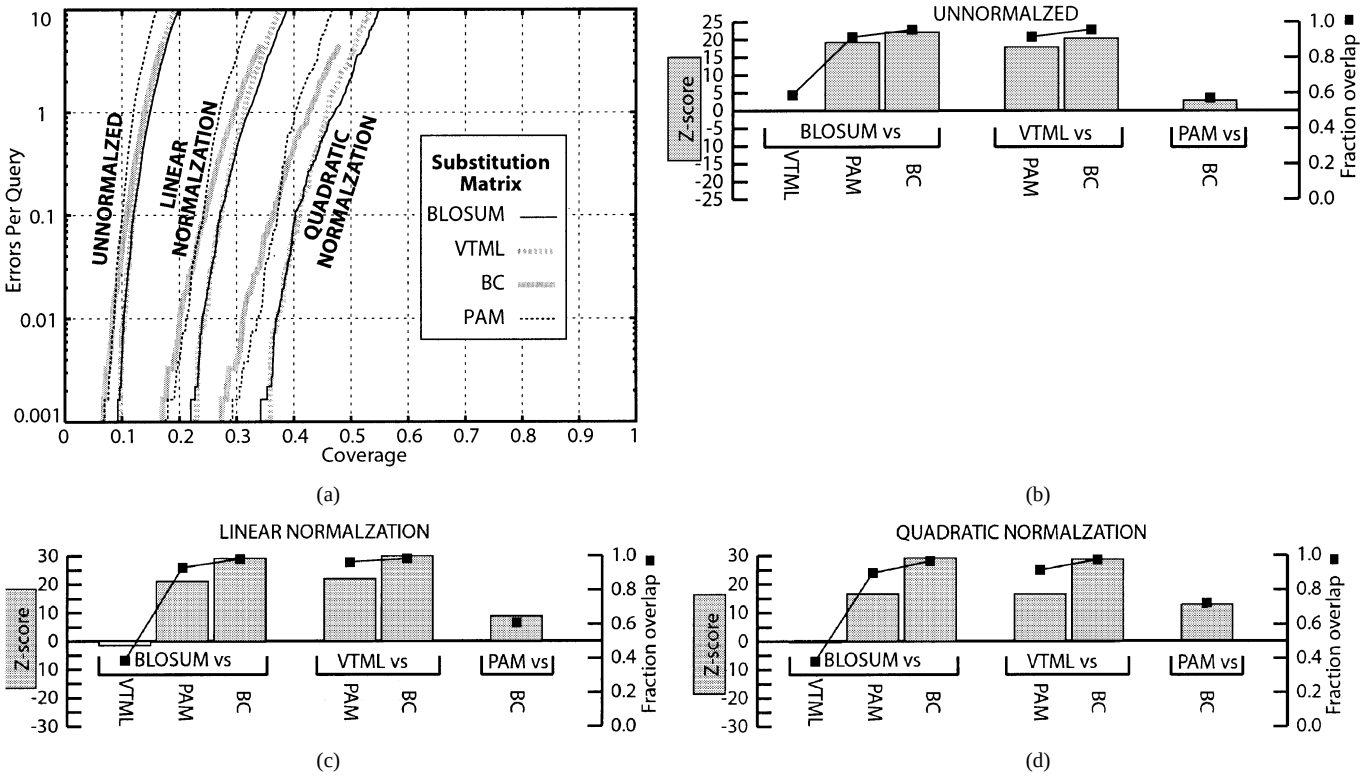


Fig. 11. Substitution matrix evaluation. SSEARCH was used with the optimum matrix from each matrix family with the optimum gap parameters for each. (a) CVE plots, under each normalization scheme were generated. (b)–(d) Bootstrapping was done (200 samples) for each matrix. Z scores and fraction overlap are given under each normalization scheme.

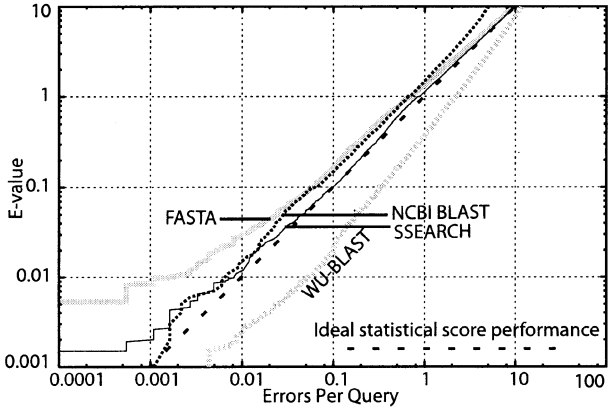


Fig. 12. Reported statistical score evaluation. The statistical scores at which errors occurred were plotted against the number of EPQ. An ideal scoring method would follow the dotted black line and, for example, would report one false positive at the level of one error per query. Methods above the dotted line are conservative, while those below the line exaggerate significance.

the composition of the database sequences used for this evaluation. ASTRAL sequences are all single domain sequences of known structure—typically soluble and globular, and thus generally well-behaved.

IV. Database Growth

It is not obvious how database growth will affect the performance of similarity detection methods. As databases grow, it becomes more likely that there will be present at least a single related sequence for any given query. However, the most useful statistical score, the E value, can be adversely affected by database growth [59]. Even though the raw alignment score for any pair of sequences will not change as databases grow, the E value significance does. This is because E values are calculated as a function of the size of the database that was searched.

Fig. 13(a) shows the growth of the number of solved structures within the Protein Data Bank (PDB) compared with the number of superfamilies within recent SCOP releases. The number of solved structures is growing at a faster rate than the number of superfamilies. This means that newly solved domain structures are more often being classified into existing superfamilies than they are defining new superfamilies. For this reason, many superfamilies are growing and, as shown in Fig. 13(b), this seems to have a negative impact on the ability to detect all true homologues at a given error rate. We ran NCBI BLAST searches, using default parameters, to generate CVE plots from the ASTRAL databases, filtered at 40% sequence identity, corresponding to each SCOP release. The relationships within each subsequent ASTRAL database release are more difficult to detect than those of the previous database.

V. Speed Evaluation of Methods

A final consideration in evaluating sequence comparison methods is speed. Each of the four pairwise methods was evaluated by the speed at which it could perform a set of database searches. The computing environment

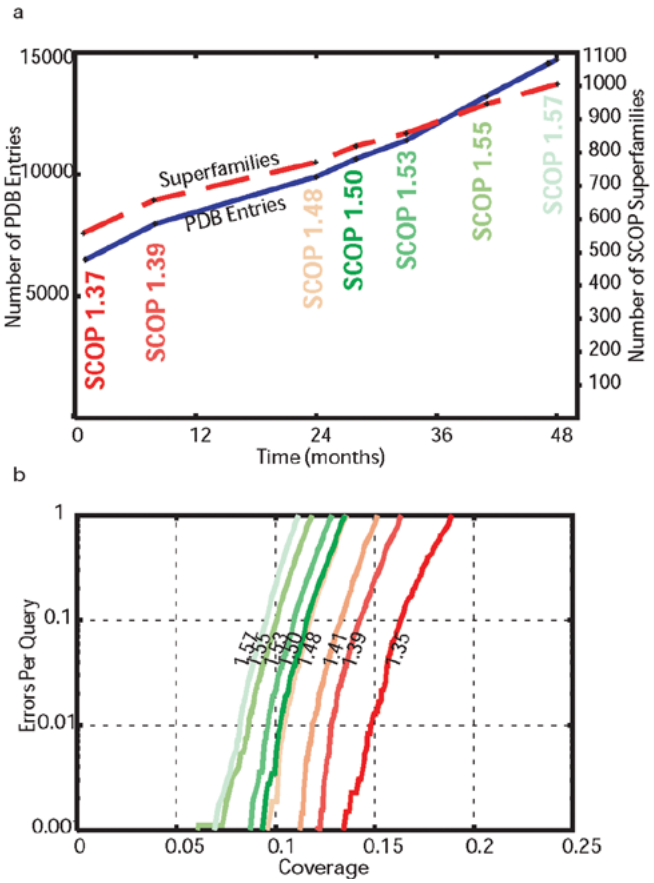


Fig. 13. Effect of database growth. (a) The number of superfamilies and PDB entries (solved structures) are plotted against time for several version of SCOP. (b) NCBI BLAST, with default parameters, was used to generate CVE plots for the ASTRAL database derived from each SCOP release.

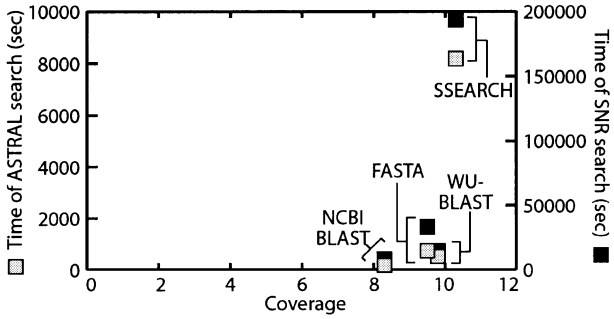


Fig. 14. Time evaluation of pairwise methods. Each pairwise method was used to perform an ASTRAL-1.57 40% sequence identity database-versus-database search and a search of 100 randomly selected sequences against the SNR database. The time required to complete these searches is reported along the y axes. Percent coverage generated at 0.01 EPQ is shown on the x axis.

was controlled in this experiment. A single-processor 1–Ghz Intel Pentium III machine was used to perform database-versus-database searches of the ASTRAL 1.57 sequences, filtered at 40% sequence identity (the training and test database sequences together) and also a search of 100 randomly chosen ASTRAL sequences against the ∼800 000–member SNR database. The results of these time trials are shown in Fig. 14. The time required to complete these database searches is plotted against the coverage generated at 0.01 EPQ. The heuristics employed by both BLAST versions, as well as FASTA, offer a significant performance increase in terms of database search time over the complete Smith–Waterman search used by SSEARCH.

Also of note is the speed difference between WU-BLAST and NCBI BLAST. As the algorithms share a common origin, the performance differences in terms of speed and CVE performance may largely be attributable to differences in the default parameter settings of each. WU-BLAST is set, by default, to perform a more careful search, whereas NCBI BLAST is optimized for speed.

VI. Discussion

The power, speed, and accessibility of pairwise sequence comparison programs have made them some of the most important methods—experimental or computational—for biological discovery. We have evaluated the merits of several of these programs using new tools that address the effect of database compositional bias and allow the significance of performance differences to be measured. Using these tools, we also evaluated the impact of parameter choice. The VTML and BLOSUM family of substitution matrices are most suitable for detecting remote homologues, recognizing a significantly larger fraction of relations than other matrices. Using these matrices, there is a range around the optimum gap-parameter set in which results are not significantly different.

The rigorous SSEARCH program detects a significantly greater fraction of the relations between remote homologues than any of the heuristic methods. Further, the significance scores reported by SSEARCH are remarkably reliable. The price for these benefits is a greater than tenfold time penalty.

The analyses chosen for presentation here were those deemed to be of interest to the community at large. Additional results, datasets, and documented tools implementing the bootstrap and normalization procedures are available from http://compbio.berkeley.edu.

Acknowledgment

The authors would like to acknowledge the helpful discussions with S. Altschul, J.-M. Chandonia, G. Crooks, E. Hill, R. Hughey, K. Karplus, W. Pearson, A. Smith, and J. Spouge.

References

[1] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *J. Mol. Biol.*, vol. 215, pp. 403–410, 1990.

[2] W. Gish. (1996–2002) WU BLAST. [Online]. Available: http://blast.wustl.edu.

[3] W. R. Pearson and D. J. Lipman, "Improved tools for biological sequence comparison," *Proc. Nat. Acad. Sci. USA*, vol. 85, pp. 2444–2448, 1988.

[4] J. M. Chandonia, N. S. Walker, L. Lo Conte, P. Koehl, M. Levitt, and S. E. Brenner, "ASTRAL compendium enhancements," *Nucleic Acids Res.*, vol. 30, pp. 260–263, 2002.

[5] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, "SCOP: A structural classification of proteins database for the investigation of sequences and structures," *J. Mol. Biol.*, vol. 247, pp. 536–540, 1995.

[6] S. E. Brenner, C. Chothia, and T. J. Hubbard, "Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships," in *Proc. Nat. Acad. Sci. USA*, vol. 95, 1998, pp. 6073–6078.
[7] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, B. A. Rapp, and D. L. Wheeler, "GenBank," *Nucleic Acids Res.*, vol. 30, pp. 17–20, 2002.
[8] E. S. Lander *et al.*, "Initial sequencing and analysis of the human genome," *Nature*, vol. 409, pp. 860–921, 2001.
[9] L. Stein, "Genome annotation: From sequence to biology," *Nat. Rev. Genetics*, vol. 2, pp. 493–503, 2001.
[10] D. Fischer and D. Eisenberg, "Predicting structures for genome proteins," *Curr. Opinion Structural Biol.*, vol. 9, pp. 208–211, 1999.
[11] C. Chothia and A. M. Lesk, "The relation between the divergence of sequence and structure in proteins," *EMBO J.*, vol. 5, pp. 823–826, 1986.
[12] G. Dodson and A. Wlodawer, "Catalytic triads and their relatives," *Trends Biochem. Sci.*, vol. 23, pp. 347–352, 1998.
[13] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *J. Mol. Biol.*, vol. 48, pp. 443–453, 1970.
[14] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *J. Mol. Biol.*, vol. 147, pp. 195–197, 1981.
[15] W. R. Pearson, "Searching protein sequence libraries: Comparison of the sensitivity and selectivity of the Smith–Waterman and FASTA algorithms," *Genomics*, vol. 11, pp. 635–650, 1991.
[16] S. Karlin and S. F. Altschul, "Applications and statistics for multiple high-scoring segments in molecular sequences," in *Proc. Nat. Acad. Sci. USA*, vol. 90, 1993, pp. 5873–5877.
[17] S. F. Altschul and W. Gish, "Local alignment statistics," *Methods Enzymol.*, vol. 266, pp. 460–480, 1996.
[18] S. F. Altschul, "Amino acid substitution matrices from an information theoretic perspective," *J. Mol. Biol.*, vol. 219, pp. 555–565, 1991.
[19] S. Karlin and S. F. Altschul, "Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes," *Proc. Nat. Acad. Sci. USA*, vol. 87, pp. 2264–2268, 1990.
[20] T. F. Smith, M. S. Waterman, and C. Burks, "The statistical distribution of nucleic acid similarities," *Nucleic Acids Res.*, vol. 13, pp. 645–656, 1985.
[21] S. F. Altschul and B. W. Erickson, "A nonlinear measure of subalignment similarity and its significance levels," *Bull. Math. Biol.*, vol. 48, pp. 617–632, 1986.
[22] J. F. Collins, A. F. Coulson, and A. Lyall, "The significance of protein sequence similarities," *Comput. Appl. Biosci.*, vol. 4, pp. 67–71, 1988.
[23] S. F. Altschul, R. Bundschuh, R. Olsen, and T. Hwa, "The estimation of statistical parameters for local alignment score distributions," *Nucleic Acids Res.*, vol. 29, pp. 351–361, 2001.
[24] W. R. Pearson, "Empirical statistical estimates for sequence similarity searches," *J. Mol. Biol.*, vol. 276, pp. 71–84, 1998.
[25] S. F. Altschul, M. S. Boguski, W. Gish, and J. C. Wootton, "Issues in searching molecular sequence databases," *Nat. Genetics*, vol. 6, pp. 119–129, 1994.
[26] C. H. Wu, H. Huang, L. Arminski, J. Castro-Alvear, Y. Chen, Z. Z. Hu, R. S. Ledley, K. C. Lewis, H. W. Mewes, B. C. Orcutt, B. E. Suzek, A. Tsugita, C. R. Vinayaka, L. S. Yeh, J. Zhang, and W. C. Barker, "The protein information resource: An integrated public resource of functional annotation of proteins," *Nucleic Acids Res.*, vol. 30, pp. 35–37, 2002.
[27] W. R. Pearson, "Comparison of methods for searching protein-sequence databases," *Protein Sci.*, vol. 4, pp. 1145–1160, 1995.
[28] ——, "Effective protein sequence comparison," in *Computer Methods for Macromolecular Sequence Analysis*, ser. Methods in Enzymology, 1996, vol. 266, pp. 227–258.
[29] A. A. Schaffer, Y. I. Wolf, C. P. Ponting, E. V. Koonin, L. Aravind, and S. F. Altschul, "IMPALA: Matching a protein sequence against a collection of PSI-BLAST-constructed position-specific score matrices," *Bioinformatics*, vol. 15, pp. 1000–1011, 1999.
[30] A. A. Schaffer, L. Aravind, T. L. Madden, S. Shavirin, J. L. Spouge, Y. I. Wolf, E. V. Koonin, and S. F. Altschul, "Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements," *Nucleic Acids Res.*, vol. 29, pp. 2994–3005, 2001.
[31] S. Henikoff and J. G. Henikoff, "Performance evaluation of amino acid substitution matrices," *Proteins*, vol. 17, pp. 49–61, 1993.
[32] L. Falquet, M. Pagni, P. Bucher, N. Hulo, C. J. Sigrist, K. Hofmann, and A. Bairoch, "The PROSITE database, its status in 2002," *Nucleic Acids Res.*, vol. 30, pp. 235–238, 2002.
[33] S. Bocs, A. Danchin, and C. Medigue, "Re-annotation of genome microbial CoDing-Sequences: Finding new genes and inaccurately annotated genes," *BMC Bioinformatics*, vol. 3, p. 5, 2002.
[34] S. E. Brenner, "Errors in genome annotation," *Trends Genetics*, vol. 15, pp. 132–133, 1999.
[35] K. Karplus, C. Barrett, and R. Hughey, "Hidden Markov models for detecting remote protein homologies," *Bioinformatics*, vol. 14, pp. 846–856, 1998.
[36] E. Lindahl and A. Elofsson, "Identification of related proteins on family, superfamily and fold level," *J. Mol. Biol.*, vol. 295, pp. 613–625, 2000.
[37] S. E. Brenner, T. Hubbard, A. Murzin, and C. Chothia, "Gene duplications in H. influenzae," *Nature*, vol. 378, p. 140, 1995.
[38] C. A. Orengo, A. D. Michie, S. Jones, D. T. Jones, M. B. Swindells, and J. M. Thornton, "CATH—A hierarchic classification of protein domain structures," *Structure*, vol. 5, pp. 1093–1108, 1997.
[39] J. D. Blake and F. E. Cohen, "Pairwise sequence alignment below the twilight zone," *J. Mol. Biol.*, vol. 307, pp. 721–735, 2001.
[40] V. Geetha, V. Di Francesco, J. Garnier, and P. J. Munson, "Comparing protein sequence-based and predicted secondary structure-based methods for identification of remote homologs," *Protein Eng.*, vol. 12, pp. 527–534, 1999.
[41] C. Sander and R. Schneider, "Database of homology-derived protein structures and the structural meaning of sequence alignment," *Proteins*, vol. 9, pp. 56–68, 1991.
[42] B. Rost, "Twilight zone of protein sequence alignments," *Protein Eng.*, vol. 12, pp. 85–94, 1999.
[43] J. C. Wootton and S. Federhen, "Analysis of compositionally biased regions in sequence databases," *Meth. Enzymol.*, vol. 266, pp. 554–571, 1996.
[44] A. Bairoch and R. Apweiler, "The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000," *Nucleic Acids Res.*, vol. 28, pp. 45–48, 2000.
[45] A. Krogh, B. Larsson, G. von Heijne, and E. L. Sonnhammer, "Predicting transmembrane protein topology with a hidden Markov model: Application to complete genomes," *J. Mol. Biol.*, vol. 305, pp. 567–580, 2001.
[46] A. Lupas, "Prediction and analysis of coiled-coil structures," *Meth. Enzymol.*, vol. 266, pp. 513–525, 1996.
[47] M. H. Zweig and G. Campbell, "Receiver-operating characteristic (ROC) plots: A fundamental evaluation tool in clinical medicine," *Clin. Chem.*, vol. 39, pp. 561–577, 1993.
[48] M. Gribskov and N. L. Robinson, "Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching," *Comput. Chem.*, vol. 20, pp. 25–33, 1996.
[49] D. W. Rice and D. Eisenberg, "A 3D–1D substitution matrix for protein fold recognition that includes predicted secondary structure of the sequence," *J. Mol. Biol.*, vol. 267, pp. 1026–1038, 1997.
[50] J. Hargbo and A. Elofsson, "Hidden Markov models that use predicted secondary structures for fold recognition," *Proteins*, vol. 36, pp. 68–76, 1999.
[51] B. Efron and J. T. Robert, *An Introduction to the Bootstrap*. Boca Raton, FL: Chapman & Hall/CRC, 1993.
[52] R. Grantham, "Amino acid difference formula to help explain protein evolution," *Science*, vol. 185, pp. 862–864, 1974.
[53] W. M. Fitch, "An improved method of testing for evolutionary homology," *J. Mol. Biol.*, vol. 16, pp. 9–16, 1966.
[54] G. H. Gonnet, M. A. Cohen, and S. A. Benner, "Exhaustive matching of the entire protein sequence database," *Science*, vol. 256, pp. 1443–1445, 1992.
[55] M. O. Dayhoff, R. M. Schwartz, and B. C. Orcutt, "A model of evolutionary change in proteins. Matrices for detecting distant relationships," in *Atlas of Protein Sequence and Structure*. Washington, DC: National Biomedical Research Foundation, 1978, vol. 5, pp. 345–358.
[56] S. Henikoff and J. G. Henikoff, "Amino acid substitution matrices from protein blocks," *Proc. Nat. Acad. Sci. USA*, vol. 89, pp. 10 915–10 919, 1992.
[57] G. Vogt, T. Etzold, and P. Argos, "An assessment of amino acid exchange matrices in aligning protein sequences: The twilight zone revisited," *J. Mol. Biol.*, vol. 249, pp. 816–831, 1995.
[58] T. Muller, R. Spang, and M. Vingron, "Estimating amino acid substitution models: A comparison of Dayhoff's estimator, the resolvent approach and a maximum likelihood method," *Mol. Biol. Evol.*, vol. 19, pp. 8–13, 2002.

[59] R. Spang and M. Vingron, "Limits of homology detection by pairwise sequence comparison," *Bioinformatics*, vol. 17, pp. 338–342, 2001.

Richard E. Green received the B.Sc. degree in genetics from the University of Georgia, Athens, in 1997.

He is currently working toward the Ph.D. degree in molecular and cell biology at the University of California, Berkeley. He has presented research at the 1997 Drosophila Research Conference and the 2002 ISMB Conference. He has also coauthored several papers in the fields of molecular biology and computational biology. His current research interests include understanding the impact of alternative splicing on gene expression and development and evaluation of sequence comparison algorithms.

Steven E. Brenner received the A.B. degree from Harvard University, Cambridge, MA, in 1992 and the Ph.D. degree from the MRC Laboratory of Molecular Biology and the University of Cambridge, Cambridge, UK, in 1997. He was a postdoctoral researcher at Stanford University, Stanford, CA, from 1997 to 1999.

He is currently Assistant Professor and leader of a computational genomics research group at the University of California, Berkeley. He has authored or coauthored more than 35 scientific papers and reviews, as well as one book. His research interests include computational approaches for structural genomics and sequence analysis, and the use of both of these to infer molecular function. Recent directions include the study of RNA structure and of the coupling of alternative splicing and nonsense-mediated decay as a general means of gene regulation.

Dr. Brenner is a director of the International Society for Computational Biology and a founding director of the Open Bioinformatics Foundation. He has received several awards and fellowships, including being a Searle Scholar.