PROTEINS: Structure, Function, and Genetics 17:49–61 (1993)

Performance Evaluation of Amino Acid Substitution Matrices

Steven Henikoff and Jorja G. Henikoff
Howard Hughes Medical Institute, Basic Sciences Division, Fred Hutchinson Cancer Research Center, Seattle, Washington 98104

ABSTRACT **Several choices of amino acid substitution matrices are currently available for searching and alignment applications. These choices were evaluated using the BLAST searching program, which is extremely sensitive to differences among matrices, and the Prosite catalog, which lists members of hundreds of protein families. Matrices derived directly from either sequence-based or structure-based alignments of distantly related proteins performed much better overall than extrapolated matrices based on the Dayhoff evolutionary model. Similar results were obtained with the FASTA searching program. Improved performance appears to be general rather than family-specific, reflecting improved accuracy in scoring alignments. An implementation of a multiple matrix strategy was also tested. While no combination of three matrices performed as well as the single best matrix, BLOSUM 62, good results were obtained using a combination of sequence-based and structure-based matrices. This hybrid set of matrices is likely to be useful in certain situations. Our results illustrate the importance of matrix selection and the value of a comprehensive approach to evaluation of protein comparison tools.**
© 1993 Wiley-Liss, Inc.

Key words: homology searching, mutation data matrix, amino acid sequence, alignment algorithms, database searching

INTRODUCTION

A variety of computer-based tools are in general use for comparing protein sequences. Alignment of two sequences is typically accomplished using a dynamic programming algorithm.[1] Searching of databanks is usually done using local alignment programs such as FASTA,[2] BLAST,[3] or parallel implementations of dynamic programming algorithms.[4–6] Multisequence alignments are carried out using a variety of strategies.[7–12] For the classification of proteins into families all of the above approaches have been used, singly or in combination, as well as alignment-based strategies specifically designed for that task.[13–17] Recently, other computer-based tools have become available that do not directly involve sequence alignment, such as structure-based alignments,[18] neural network methods,[19,20] and indexing methods.[21]

Given the number of tools now available, evaluation of alternative methods for comparing protein sequences is especially important. However, the description of a new method is typically accompanied by a few anecdotal examples showing how much better the new method works than a currently popular method. A more thorough testing strategy was used by Pearson, who evaluated database searching programs by choosing queries from each of the 34 largest superfamilies in the NBRF-PIR annotated database.[22] He concluded that the performance of FASTA at ktup = 1 approaches that using a full dynamic programming algorithm. Later, we applied a similar strategy to the empirical evaluation of amino acid substitution matrices.[23] Matrices in our BLOSUM series were compared to those in the Dayhoff mutation data matrix (MDM_{78})[24] series by measuring the ability of BLAST to detect members from 504 different protein groups listed in the PROSITE catalog using a query chosen from each group. The performance of matrices in the BLOSUM series, especially of BLOSUM 62, was found to be far superior to that of the MDM_{78} matrices,[24] as well as to two other matrices recently introduced as replacements for the widely used MDM_{78} PAM 250 matrix.[25,26] These more comprehensive evaluation strategies are clearly necessary for identifying the most effective alternatives for database searching.

As a result of our previous tests, the fundamental importance of the choice of substitution matrix became evident. Every sequence-based alignment program uses a substitution matrix for scoring, and in nearly every case, one of the MDM_{78} matrices is the default. In particular, the poor performance of the MDM_{78} PAM 250 matrix in our tests calls into question many conclusions that have been based on the

Received January 29, 1993; revision accepted April 1, 1993.
Address reprint requests to Dr. Steven Henikoff, Howard Hughes Medical Institute, Basic Sciences Division, Fred Hutchinson Cancer Research Center, 1124 Columbia Street, M-684, Seattle WA 98104.

© 1993 WILEY-LISS, INC.

failure of a program to detect a homolog in database search. For example, Bowie and co-workers claimed better performance of structure-based profiles than a comparable sequence-based approach (using MDM_{78} PAM 250) for detection of homologs to *E. coli* Crp and Rbp proteins.[18] However, FASTA[2] using BLOSUM 62 clearly outperforms the structure-based profile for both of their queries (unpublished results). So, evaluations of protein comparison programs depend upon the substitution matrix used, whether directly in providing scoring parameters, or indirectly in drawing conclusions about relative performance.

Only recently has much attention been paid to theoretical aspects influencing the choice of a substitution matrix. Altschul has stressed the importance of the information content of a matrix, measured in bit units as relative entropy.[27] His theoretical analysis concluded that the MDM_{78} PAM 250 matrix, with relative entropy less than 0.4 bit, had insufficient information for most effective database searching. This conclusion was confirmed in our tests, which showed that for both the MDM_{78} PAM and the BLOSUM series, matrices with relative entropies of about 0.7 performed the best overall.[23] Altschul also has argued in favor of using multiple matrices for searching, since no single matrix will efficiently score all correct alignments of proteins that differ in evolutionary distance.[28] His idea is that by using a set of matrices that is tailored to different evolutionary distances, it should be possible to efficiently detect alignments that otherwise might be missed. This advantage should offset the cost of an increased background of false positive alignments when results from multiple searches are combined. While Altschul's approach was illustrated for the MDM_{78} series ("ALL-PAM"), it is general and can be implemented for any matrix set.

Here we present comprehensive tests of different matrices and matrix sets using the BLAST and FASTA searching programs. The tested matrices fall into two classes. One class was derived by extrapolation from closely related sequences, based on the PAM (percent accepted mutation) evolutionary model of Dayhoff and co-workers.[29] Of these, the series from Jones and co-workers[26] was found to be a clear improvement over the MDM_{78} series.[24] The other class consists of matrices derived directly from multiple alignment data, either from sequence-based[17] or from structure-based alignments.[30] Matrices of this class are not based on evolutionary models. Searching performance of the structure-based (STR) matrix approached that of the best sequence-based matrix, BLOSUM 62. The second class of matrices was much better for detecting distant relationships than any of the matrices based on the PAM evolutionary model. Overall performance differences did not appear to be family-specific, but rather indicated that for distant alignments, the extrapolated matrices are inherently less accurate, with their accuracy improving for closer relationships. We also tested a three matrix implementation of a multiple matrix strategy[28] and found that for the best performing matrices, the costs outweigh the benefits. However, the benefits can be increased by combining the STR matrix with BLOSUM series matrices; this might prove to be an advantageous strategy, for example, when other information is available for distinguishing true from false positive database matches.

METHODS

Matrix Construction

The MDM_{78} series of matrices was constructed from the 1978 Dayhoff dataset[24] using the National Center for Biotechnology Information (NCBI) PAM program version 1.0.5 (9/3/92). The matrices consisted of integers and were variably scaled to provide more contrast in lower entropy matrices, with the scaling depending on the relative entropy (H) of the matrix: scale = $1/n$ bits where n = maximum[2.0, 2.0/sqrt(H)]. So the maximum scale was half bits ($n = 2$) for H above 0.64 and third bits ($n = 3$) for H between 0.64 and 0.25. The PAM program added columns for B, Z, and X to the 20 core amino acids by computing B pairs as the frequency-weighted average of entries for D and N pairs, Z pairs as the frequency-weighted average of entries for Q and E pairs, and X pairs as the frequency-weighted average of all pairs.

The JTT series of matrices was constructed from pairwise exchange data derived from SWISS-PROT 22 using a version of the CALCPAM program (Release 2, 6/92) that we modified to calculate relative entropy (H) and expected (E) values and to variably scale the matrices based on relative entropy as described above. For matrices of equal relative entropies, JTT PAM values do not correspond precisely to MDM_{78} PAM values. Columns for B, Z, and X were calculated as averages of the rounded scaled scores weighted by the SWISS-PROT 22 amino acid frequencies. The CALCPAM program and pairwise exchange data were provided by David T. Jones.[26]

The GCB matrix[25] was obtained along with the BLAST programs from the anonymous ftp server at NCBI where it was called the "Gonnet" matrix.

The STR matrix was derived from the "All Classes" table of raw scores provided on a diskette accompanying a paper by Overington and co-workers.[30] The table contains 79,983 pairwise substitutions. These are comparable to BLOSUM pair counts,[23] except that cystine and cysteine are separated into C and J columns and rows, and a row is provided for amino acids aligned with gaps. A scoring matrix was computed as follows: (1) The gap row was ignored. (2) J and C were combined. (3) Scores, entropies, and expected values were computed as for the BLOSUM series. (4) The relative entropy of the

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	
	2	-3	-4	-5	-2	-3	-3	-4	1	0	-3	1	-1	-4	-3	-5	-3	0	-4	-4	C
		0	0	0	-1	-1	-1	0	-1	-1	-1	1	-1	-3	-1	-2	-1	-1	0	-2	S
C	11		0	0	-1	-1	0	0	1	1	0	0	1	-1	-1	-2	-1	-1	0	-3	T
S	-4	4		0	0	0	0	0	0	-1	-1	0	0	-4	-1	0	-2	-1	-3	0	P
T	-5	1	5		0	0	1	1	1	1	0	0	0	1	-1	-1	0	-1	-1	0	A
P	-8	-1	-1	7		-1	-1	0	0	0	-1	0	-1	-1	-1	-1	-1	-3	0	-2	G
A	-2	0	-1	-1	4		-1	1	0	0	1	-1	0	0	0	0	-1	0	1	-1	N
G	-6	-1	-3	-2	0	5		0	0	0	1	0	0	-1	0	-2	-1	-2	0	-2	D
N	-6	0	0	-2	-1	-1	5		0	0	-2	0	0	0	0	-1	0	-1	0	-3	E
D	-7	0	-1	-1	-1	-1	2	6		1	0	0	0	1	-2	-1	0	-1	-2	-3	Q
E	-3	-1	0	-1	0	-2	0	2	5		0	0	1	0	-2	0	1	-1	-2	-1	H
Q	-3	-1	0	-2	0	-2	0	0	2	6		2	0	-3	0	-1	0	-1	1	1	R
H	-6	-2	-2	-3	-2	-3	2	0	-2	0	8		0	0	0	0	-1	0	0	0	K
R	-2	0	-1	-2	-1	-2	-1	-2	0	1	0	7		3	0	1	-1	0	0	-1	M
K	-4	-1	0	-1	-1	-3	0	-1	1	1	0	2	5		2	0	-1	1	0	1	I
M	-5	-4	-2	-6	0	-4	-2	-4	-2	1	-2	-4	-1	8		1	0	2	-1	1	L
I	-4	-3	-2	-4	-2	-5	-3	-3	-3	-5	-5	-3	-3	1	6		1	0	0	-1	V
L	-6	-4	-3	-3	-2	-5	-3	-6	-4	-3	-3	-3	-2	3	2	5		1	0	1	F
V	-4	-3	-1	-4	0	-4	-4	-4	-2	-2	-2	-3	-3	0	2	1	5		0	0	Y
F	-2	-3	-3	-5	-3	-6	-3	-5	-4	-4	-2	-4	-3	0	1	2	-1	7		-1	W
Y	-6	-2	-2	-6	-3	-3	-1	-3	-2	-3	0	-1	-2	-1	-1	-2	-1	3	7		
W	-6	-5	-5	-4	-3	-4	-5	-6	-6	-5	-3	-2	-3	-2	-2	-1	-4	2	2	10	
	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	

Fig. 1. STR matrix from structure-based alignments[30] **(lower)** and difference matrix **(upper)** obtained by subtracting the BLOSUM 62 matrix position by position.

matrix was calculated as 0.92 bits and so it was scaled to half bits (Fig. 1, lower). (5) Columns for B, Z, and X were calculated as for the JTT PAM series.

The BLOSUM series of matrices was derived from the Blocks Database v. 5.0 as previously described,[23] except that variable scaling and the calculation of B, Z, and X values were implemented to conform with those features of the NCBI PAM program.

Following a suggestion of S. Altschul (personal communication), a series of INTERVAL matrices was constructed from the Blocks Database v 5.0 using a different method for counting pairs than was used for the BLOSUM series. For the BLOSUM series, segments in a block that were at least x% identical were clustered and pairs were counted between clusters. So pairs were counted only between segments less than x% identical. For example, if $x = 62$, then pairs were counted between segments less than 62% identical; if $x = 100$, then pairs were counted between segments unless they were 100% identical. As x increased, pairs were counted between increasingly similar segments. For INTERVAL matrices, pairs were counted between segments in a block if they were at least $x1$% but not more than $x2$% identical in order to achieve a greater range of relative entropies and to correspond better to specific evolutionary distances.

In each case, calculation of log-odds ratios and scaling was carried out before rounding to give integer matrices, except for the GCB matrix which was used as provided with the BLAST program.

As a control, we used a unitary matrix called $+6/-1$ which scores $+6$ for matches and -1 for mismatches. These values guarantee that the expected value of the matrix is negative, which is necessary for both the BLAST and FASTA algorithms.

Selection of Queries

Groups of related protein sequences were obtained from Prosite 9.0, using the same 560 nonredundant groups that were used by the PROTOMAT system[17] to construct the Blocks Database v. 5.0 (plus PS00574 which was inadvertently omitted from the database). For each group, a set of true positive sequences was identified as follows: all full-length sequences listed in PROSITE.DAT[15] and marked either "T" for true positive, "N" for false negative, or "P" for potential positive were included. Furthermore, if multiple sequences in the group shared the same gene name (all characters in the SWISS-PROT 22[31] ID before the "_" character) and most of the organism name (first three characters after the "_"), then only the longest of these sequences was included. For the 560 groups, there were a total of 11,255 true positive sequences.

To test the performance of the various matrices, queries for each group were selected to be distant from most other members of its group. A second query, distant from the first, was also selected for verification of results. For query 1, a full-length sequence was selected from among the true positive set for each of the 560 Prosite groups. If any of the sequences in a group were excluded from the best path by PROTOMAT, i.e., they did not appear in the blocks for the group, the longest of the most distant of these excluded sequences was selected. Distance was estimated from the SWISS-PROT IDs for the sequences, where two sequences with a shorter common prefix were considered to be more distant. For example, the common prefix of MYF3_HUMAN and MYF4_HUMAN is MYF, and they are considered more distant from one another than

MYC_AVIM2 and MYC_AVIME with a common prefix of MYC_AVIM, whereas there is no common prefix for HAIR_DROME and EMC_DROME. If all the sequences appeared in the blocks for a group so that there were no excluded sequences, the longest of the most distant of them was selected. For example, there were 63 sequences in the PS00038 group, and all of them were included in the best path for the two blocks constructed by PROTOMAT. Twelve of these sequences had no common prefix with any other sequences, and the longest of these 12 was DA_DROME which was selected as query 1.

Query 1 search results were used to select query 2 from each of the 257 most challenging groups as described below. Each query 2 was selected from among the true positive sequences that were missed by the most matrices during the query 1 searches. If there were multiple sequences missed by the most matrices, then the longest of the most distant of these was chosen. For the 257 most challenging groups, there were a total of 7,535 true positive sequences.

Searches

BLASTP[3] version 1.2.9 (9/4/92) was used with default parameters of $W=3$ and $E=10$. This version combines multiple high-scoring segment pairs (HSPs) for the same sequence and calculates a correspondingly modified P-value. FASTA[2] version 1.6c2 (8/92) was used with ktup = 1, gap penalties of -12 and -4, and default parameters. The databank searched was SWISS-PROT 22,[31] consisting of 25,044 sequences.

BLAST searches using query 1 for each group were carried out as previously described[23], but using the updated searching programs. Each of the 560 queries was first searched against the SWISS-PROT databank using 8 different matrices: 3 from the BLOSUM series (45, 62, 80), 3 from the MDM_{78} series (PAM 250, 160, 120), GCB, and $+6/-1$. A true positive sequence was considered "found" if it was reported by BLAST. In addition to counting the number of true positives found, we also took note of the P-value assigned to each true positive sequence by BLAST. All 8 matrices found all true positive sequences for 303 of the 560 groups. Based on these results, the original 560 groups were reduced to a smaller set of the 257 most challenging groups. BLAST searches of query 2 for each of these most challenging groups with 18 different matrices were then carried out, as well as searches of query 1 for all 560 groups using the 10 additional matrices. For the MDM_{78}, JTT and BLOSUM series, five matrices were tested. The single GCB, STR, and $+6/-1$ matrices were also tested. BLAST searches of query 2 for the most challenging groups were also done with three different INTERVAL matrices.

For pairs of matrices, the results of each of the 560 BLAST searches for query 1 and the 257 BLAST searches for query 2 were compared as follows. First, the number of the groups for which each matrix in the pair found more true positive sequences than the other was counted. In addition, the number of groups for which each matrix had a smaller P-value for more true positive sequences than the other was counted; two P-values were considered equal if the absolute value of their differences was less than 10^{-3}.

FASTA searches were done for the 257 most challenging groups with 18 different matrices. Since FASTA does not report results based on a probability value, we collected 300 results for each search and counted a true positive sequence as "found" if it appeared ahead of 99.5% of the presumed true negative sequences in SWISS-PROT. This corresponds to the evaluation criteria recommended by Pearson.[22] So, if there were n true positive sequences, then true positive hits had to occur among the highest-scoring $(25{,}044\text{-}n)\times 0.005$ hits to be counted. On average, n was about 29. For pairs of matrices with similar relative entropies, the number of groups for which each matrix in the pair found more true positive sequences than the other was counted.

Searches With Multiple Matrices

An implementation of Altschul's multiple matrix strategy[28] used several different combinations of two or three matrices. The sets of matrices were selected to provide an efficiency of at least 0.9 over the entropy range of about 0.25 to 4.0. The efficiency of using one matrix R to score HSPs that are actually modeled by a different matrix C was estimated as the entropy using scores from R and frequencies from C divided by the relative entropy of C, which is computed using both scores and frequencies from C.[27]

Sets of matrices were selected from the MDM_{78}, JTT, BLOSUM, and INTERVAL series. In addition, we tried hybrid sets to extend the range of the BLOSUM series, which achieves a maximum entropy of 1.45.

For each set of three matrices, BLASTP searches were done for the 257 most challenging groups with the default $E=10$ and again with $E=3.33$ as a "tax" for combining the results of the three searches. A true positive was considered "found" if it appeared in the results of any of the three searches. For each set of two matrices, the approach was the same, except searches were done with a "tax" of $E=5$.

The results for each set of matrices were compared with the middle matrix in the set and with the single best matrix overall, BLOSUM 62. The number of groups for which the set found more true positive sequences than BLOSUM 62 was counted, and vice versa.

Implementation

Software developed or adapted for this study was written in the standard C programming language and was compiled for SUN Sparcstation computers. Matrices, query lists, and program code are available from the authors by ftp over the Internet. Contact henikoff@sparky.fhcrc.org.

RESULTS

Evaluation of Matrix Performance Using BLAST

Given a query sequence, a database, and an amino acid substitution matrix, the BLAST algorithm scores ungapped alignments that fulfill heuristically determined criteria.[3] In our previous study,[23] the default heuristic was that in order to be scored, an alignment must include a segment of 4 amino acids [word size (W) = 4] that exceeds a certain threshold score (T), with T adjusted to take into account query and database size and the matrix. In addition, BLAST results are ranked based on statistical significance in the context of the search, so that the program reports all alignments achieving at least a certain level of significance measured in terms of an expected value, E.[32] In our previous study, the default level was $E = 25$. For this study, we used a new version of BLAST with different defaults: W was reduced to 3, T was reduced to a lower level for a given W, and E was reduced to 10. The lower threshhold increases sensitivity, while the more stringent expected value increases selectivity. In addition, the new version combines multiple HSPs; this may result in a sequence being reported even if some of these HSPs individually score below the default expected value. This feature should increase sensitivity, since similarities to a query that are scattered throughout a sequence can contribute to the combined score.

Using this new version of BLAST, we carried out searches with queries from each of the 560 nonredundant groups found in the Prosite 9.0 catalog[15] and with several different substitution matrices. Performance was measured in pairwise comparisons between matrices, so that a matrix was judged to be better for a group if BLAST reported more true positive sequences using that matrix than when using the competing matrix. When each test matrix was compared with BLOSUM 62, the results of searches using these 257 second queries were approximately the same overall as those using the 560 first queries (Fig. 2). Therefore, conclusions drawn from these data are independent of the choice of query.

The best tested matrix overall was BLOSUM 62, in agreement with our previous results.[23] Performance peaked for all series at $H \approx 0.7$ (BLOSUM 62, JTT PAM 150, and MDM PAM 160). For example, BLOSUM 62 ($H = 0.7$) performed better than either BLOSUM 50 ($H = 0.48$), or BLOSUM 80 ($H = 1.0$).

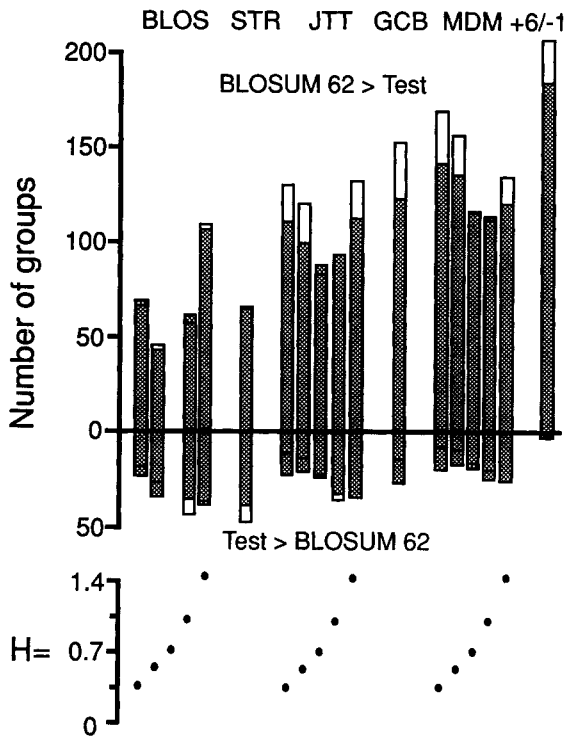

Fig. 2. BLAST performance using two dissimilar queries from each group. Results for each test matrix compared to BLOSUM 62 are shown above with the relative entropy for each test matrix plotted below. The matrix series is indicated at the top. BLOS matrices are (left to right) BLOSUM 45, 50, 62, 80, 100; JTT matrices are JTT PAM 220, 190, 150, 110, 80; MDM matrices are MDM_{78} PAM 250, 210, 160, 120, 80. The number of groups for which BLOSUM 62 detected more true positives than the test matrix (BLOSUM 62 > Test) is shown by the bars above the 0 line, and the number for which the test matrix detected more (Test > BLOSUM 62) is shown below the 0 line. There are no differences when BLOSUM 62 ($H = 0.70$) is the test matrix. Search results using query 1 for 560 groups are indicated by open bars and those using query 2 for 257 groups are indicated by superimposed shaded bars. The close correspondence between the open and shaded bars reflects very similar overall results for the two sets of queries.

Also as shown previously, all BLOSUM matrices tested were superior to matrices with comparable relative entropies based on the PAM model (Fig. 2). This includes the widely-used MDM_{78} PAM 250 matrix and the MDM_{78} PAM 120 matrix which is the default for BLAST. For example, at $H \approx 0.7$, BLOSUM 62 performed much better than either the JTT PAM 150 from Jones and co-workers[26] or the MDM_{78} PAM 160. Furthermore, all matrices from the JTT series were clearly better than their MDM_{78} counterparts. For example, using the second query, JTT PAM 150 was better than MDM_{78} PAM 160 for 81 groups and was worse for 30 groups. The single available matrix from Gonnet and co-workers[25] was a poor performer in these tests, as shown previously.[23] However, the STR matrix from the data of Overington and co-workers[30] performed much better than any of the matrices based on the PAM

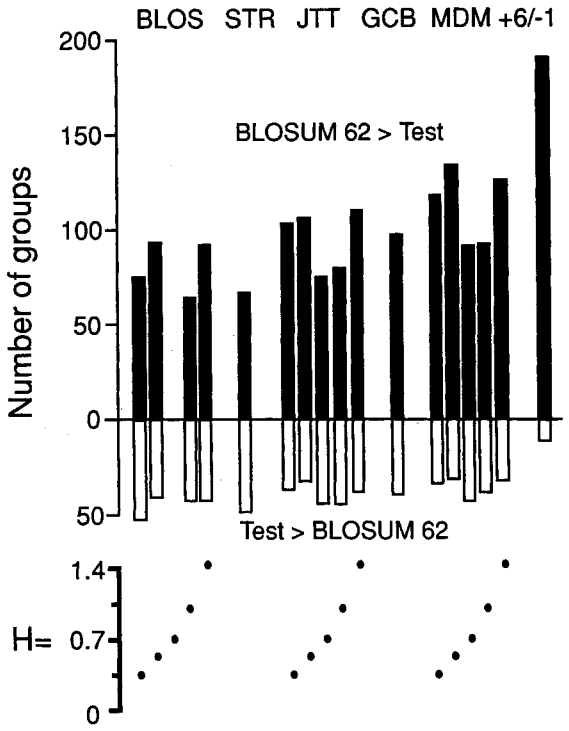

Fig. 3. FASTA performance using 257 second queries. Matrix series and performance criteria are as indicated in the legend to Figure 2.

model. This matrix performed as well as the mathematically comparable BLOSUM 80, though somewhat worse than BLOSUM 62. So, in spite of important changes in the BLAST program and defaults, our previous conclusions are confirmed; that matrices derived directly from alignments of distantly related proteins perform better than those derived by extrapolation from alignments of closely related proteins.

Evaluation of Performance Using FASTA

FASTA was also used for evaluation of matrix performance. Results with the 257 challenging queries identified in the BLAST tests are shown in Figure 3. In general, these results are similar to the BLAST results shown in Figure 2. BLOSUM 62 is still the best matrix tested, with the other BLOSUM matrices and the STR matrix performing better than their counterparts based on the PAM evolutionary model. As with BLAST, all performed better overall than the simple $+6/-1$ control matrix.

In spite of overall similarities between the BLAST and FASTA results, the performance disparities between matrices were smaller using FASTA. This is likely to be due, at least in part, to the Pearson detection criteria that a database hit must be within the top 0.5% of scores.[22] In comparison, BLAST using $E=10$ typically reports fewer results. So sequences that are not reported using BLAST because they fail to achieve a statistically significant score might still be reported using the Pearson detection criteria. It is also possible that FASTA at ktup = 1 is generally more sensitive to detection of true positive alignments than is BLAST, so that fewer sequences are missed overall. Another possibility is that the initial requirement for exact matches in FASTA screens out some challenging HSPs that BLAST can detect with an accurate (but not an inaccurate) matrix because exact matches are not required.

Another difference between BLAST and FASTA results is that the matrices with relative entropy of about 0.36 performed better using FASTA than matrices in the series with relative entropies of about 0.48. This appears to be a FASTA anomaly, because performance improves again as relative entropy increases to 0.7. It seems likely that this anomaly is a consequence of the influence of the substitution matrix on other parameters (e.g., the "joining penalty") that were not changed or scaled to the substitution matrix in our tests. Since FASTA was implemented and tested using the MDM_{78} PAM 250 matrix ($H=0.36$), other parameters might have been set to maximize performance with this matrix.

Sequence-by-Sequence Comparisons

An alternative way to compare matrix performance is possible with BLAST. The statistical significance of an alignment between the query and each true positive sequence provides a quantitative measure of how well the alignment compares with chance alignments estimated from the distribution of search results. All HSPs contributing to the statistical significance of an alignment are reported with a combined Poisson P-value. The better of two matrices should report a lower P-value for detection of a true positive sequence by a query. This method of comparison should be more informative than that in which the criterion is simply the number of sequences detected, as illustrated with an example (Fig. 4). Based on BLAST detection criteria, the query MTG1_HAEGA detects 7 true positive sequences using BLOSUM 62 not detected using MDM_{78} PAM 160 and 2 sequences using MDM_{78} PAM 160 not detected using BLOSUM 62. When the P-values are graphed for all sequences detected using one or the other matrix (lines in Fig. 4), it can be seen that the performance disparity is more striking. Of the 24 sequences reported in either or both searches, use of BLOSUM 62 led to lower P-values in 20 cases (solid lines slanted to the right in Fig. 4, upper panel), while use of MDM_{78} PAM 160 led to lower P-values only in the 2 cases that BLOSUM 62 failed to report (stippled lines slanted to the left). For the corresponding comparison between BLOSUM 62 and JTT PAM 150, each search reported 3 sequences not reported in the other search. However, a comparison of P-values breaks this tie;

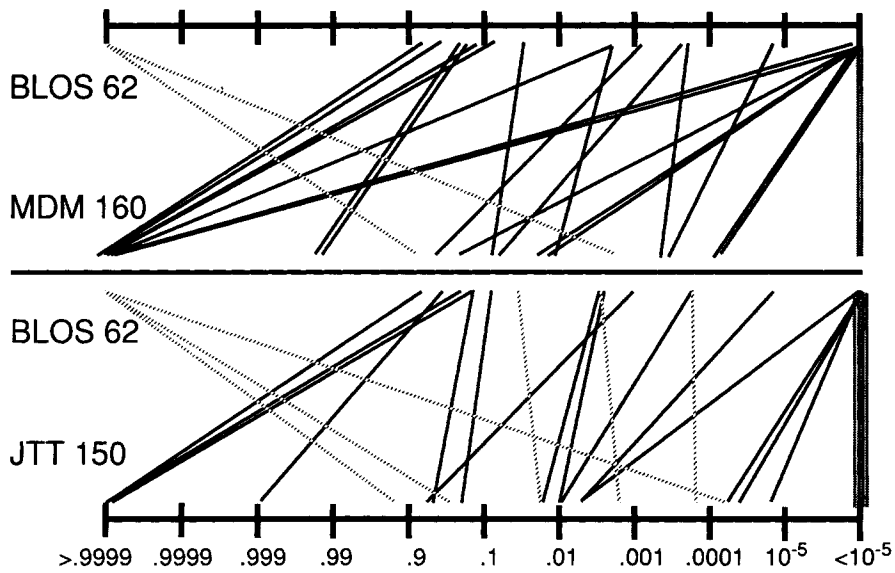

Fig. 4. BLAST search results using MTG1_HAEGA, a member of the C_5 methyltransferase family (Prosite PS00094), as query of SWISS-PROT 22. Poisson *P*-values of individual family members are plotted on a logarithmic scale along the top axes for BLOSUM 62 (BLOS 62), along the bottom axis of the top panel for MDM_{78} PAM 160, and along the bottom axis of the bottom panel for JTT PAM 150. Relative performance of two matrices is evident from the slope of the lines connecting points for each family member. Lines originating at the far left are those not reported by BLAST ($P > .9999$). Solid lines slanting right represent sequences detected at a lower *P*-value for BLOSUM 62 than for the other matrix, whereas stippled lines (slanting left) represent sequences detected at a higher *P*-value for BLOSUM 62.

of the 25 sequences reported in either or both searches, use of BLOSUM 62 led to lower *P*-values in 15 cases, while use of JTT PAM 150 led to lower *P*-values in only 6 cases (Fig. 4, lower panel).

In comparing two matrices, we used *P*-values to determine relative performance for each of the 257 most challenging groups. Only those true positive hits for which the difference in *P*-values was greater than 10^{-3} were counted. For these tests, we limited comparisons to matrices with approximately similar relative entropies in the 0.36–1.45 range (Fig. 5). In confirmation of previous results, the BLOSUM matrices were superior overall and the STR matrix performed nearly as well. Matrices based on the PAM evolutionary model were again much worse, although the JTT matrices were still clearly improved over the MDM_{78} series. In addition, an interesting regularity was revealed. The superiority of the BLOSUM series was most evident for the lower entropy (more distant) matrices in comparison to the JTT or the MDM_{78} series. This superiority became less pronounced for higher entropy (less distant) matrices. This improvement is consistent with the notion that distant extrapolation results in matrix inaccuracy. So extrapolation of the Dayhoff mutation rates to 250 PAMs gives very poor performance relative to a mathematically comparable BLOSUM matrix, whereas extrapolation to only 80 PAMs gives somewhat better performance (Fig. 5). The same trend is seen with the JTT series. As in the previous tests, GCB matrix performance was comparable to that of the JTT matrix with relative entropy of about 0.4, consistent with the fact that the GCB matrix is a distant extrapolation based loosely on the PAM evolutionary model. In contrast, the STR matrix based directly on structural alignments performs about as well as matrices in the BLOSUM series.

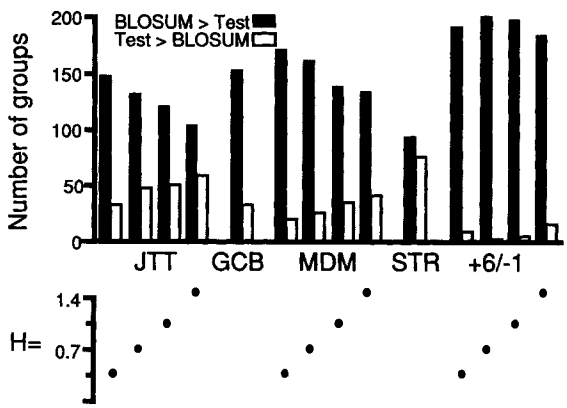

Fig. 5. BLAST search results based on *P*-value differences for 257 second queries. Each bar represents the number of groups for which BLAST reported a higher Poisson *P*-value for one matrix relative to another matrix for matrices of similar relative entropy. BLOSUM matrices are (left to right) BLOSUM 45, 62, 80, 100; JTT matrices are PAM 220, 150, 110, 80; comparable MDM_{78} matrices are PAM 250, 160, 120, 80.

The reduced performance of extrapolated matrices at greater evolutionary distances compared to their

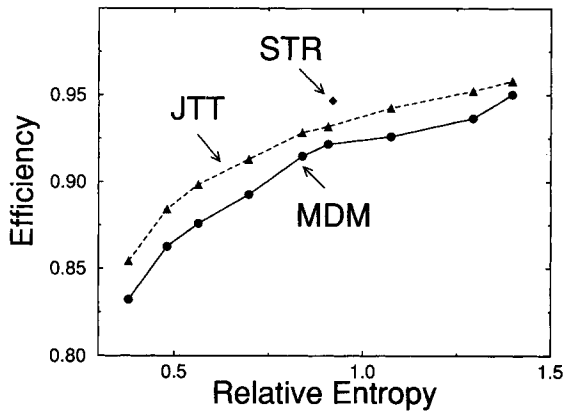

Fig. 6. Efficiency plots for matrices based on different models and datasets relative to the corresponding matrices based on the BLOSUM model as a function of relative entropy. JTT series (▲), MDM_{78} series (●), and the single STR matrix compared to BLOSUM 80 (◆). Relative entropy increases with decreasing evolutionary distance measured in PAMs, so that for the MDM_{78} series, PAM 250 is to the left and PAM 80 is to the right.

BLOSUM counterparts parallels their relative efficiencies[28] (Fig. 6). Both the JTT and MDM_{78} matrices become less efficient relative to their BLOSUM counterparts for more distant extrapolations (smaller H). It is interesting that the STR matrix, which performs very well, shows only slightly higher efficiency relative to the BLOSUM standard than the corresponding JTT matrix. Reduced efficiency of the JTT and MDM_{78} matrices relative to BLOSUM probably corresponds to inaccurate scoring of distant alignments, whereas reduced efficiency of the STR matrix relative to BLOSUM might correspond to accurate scoring of distant alignments that are less well represented by the BLOSUM model. The efficiencies plotted in Figure 6 suggest the potential improvement BLOSUM matrices can provide at various relative entropies.

Performance Differences Are Not Attributable to Group Differences

The performance results involving two distant queries from 257 protein groups can be used to ask whether there is any evidence for group specificity of a matrix. For example, based on P-value differences, BLOSUM 62 performed better than MDM_{78} PAM 160 for 161 of the 257 groups, whereas MDM_{78} PAM 160 performed better than BLOSUM 62 for 25 groups (Fig. 5). One explanation is that BLOSUM 62 is more accurate than MDM_{78} PAM 160 for these particular 161 groups, and MDM_{78} PAM 160 is more accurate than BLOSUM 62 for the 25 groups. Alternatively, BLOSUM 62 might be more accurate overall, and it does worse for the 25 groups only because of chance. To distinguish between these alternatives, we asked whether there was any correlation between query 1 and query 2 results for the 257 common groups. That is, if BLOSUM 62 is more accurate than MDM_{78} PAM 160 for a particular group, then it should perform better for that group using both queries, and vice versa. Alternatively, if BLOSUM 62 is more accurate overall than MDM_{78} PAM 160, then performance using two dissimilar queries is not expected to be correlated for a particular group.

For each group, the difference between the number of sequences scored higher by one matrix and the number scored higher by another matrix was plotted for query 1 on the x-axis and query 2 on the y-axis. A scatter plot representing these differences is expected to form a circular cloud of points if there is no correlation, with the cloud flattening at a 45° angle as correlation increases. Comparisons were made between three comparable JTT and MDM_{78} matrices (Fig. 7, upper panels) and between the corresponding BLOSUM and MDM_{78} matrices (Fig. 7, lower panels). Each comparison between JTT and MDM_{78} matrices showed a slightly flattened cloud and a regression line with positive slope, even though the JTT series is an update based on the PAM model, which mainly corrects inaccuracies caused by insufficient data in the MDM_{78} dataset.[24] Here, better performance of the JTT series over the MDM_{78} series is attributable to better overall accuracy, not group specificity. The weak correlations can be accounted for by the limited sequence similarity between query 1 and query 2 in each group. Similar weak correlations were also seen for comparisons between BLOSUM and MDM_{78} series matrices (Fig. 7, lower panels). Since weak correlations of this magnitude are expected from sequence similarities between query 1 and query 2, there is no evidence for group specificity. Rather, it appears that the performance of a matrix for a group is predictable from the performance of that matrix for all groups.

BLAST Searches Using Multiple Matrices

A set of matrices can be used in combination to detect alignments at all evolutionary distances. The set can be chosen such that for each HSP length, the efficiency of one matrix exceeds a certain percentage of the optimum for that length.[28] Altschul estimated that for protein database searching, 93% is a sufficient minimum efficiency level.[27] For a three matrix (3-mat) set, one matrix would efficiently score HSPs in the most frequently represented range of lengths, a higher entropy (or less distant) matrix would efficiently score shorter HSPs, while a lower entropy (more distant) matrix would efficiently score longer HSPs. For the MDM_{78} series, Altschul recommended PAM 30 (93% efficiency for HSP lengths of 7–19 aa), PAM 120 (19–50 aa), and PAM 250 (50–126 aa) as a potentially useful 3-mat set.[28] We tested this set, along with sets from other series chosen to fulfill the 93% efficiency criteria. For the JTT series,

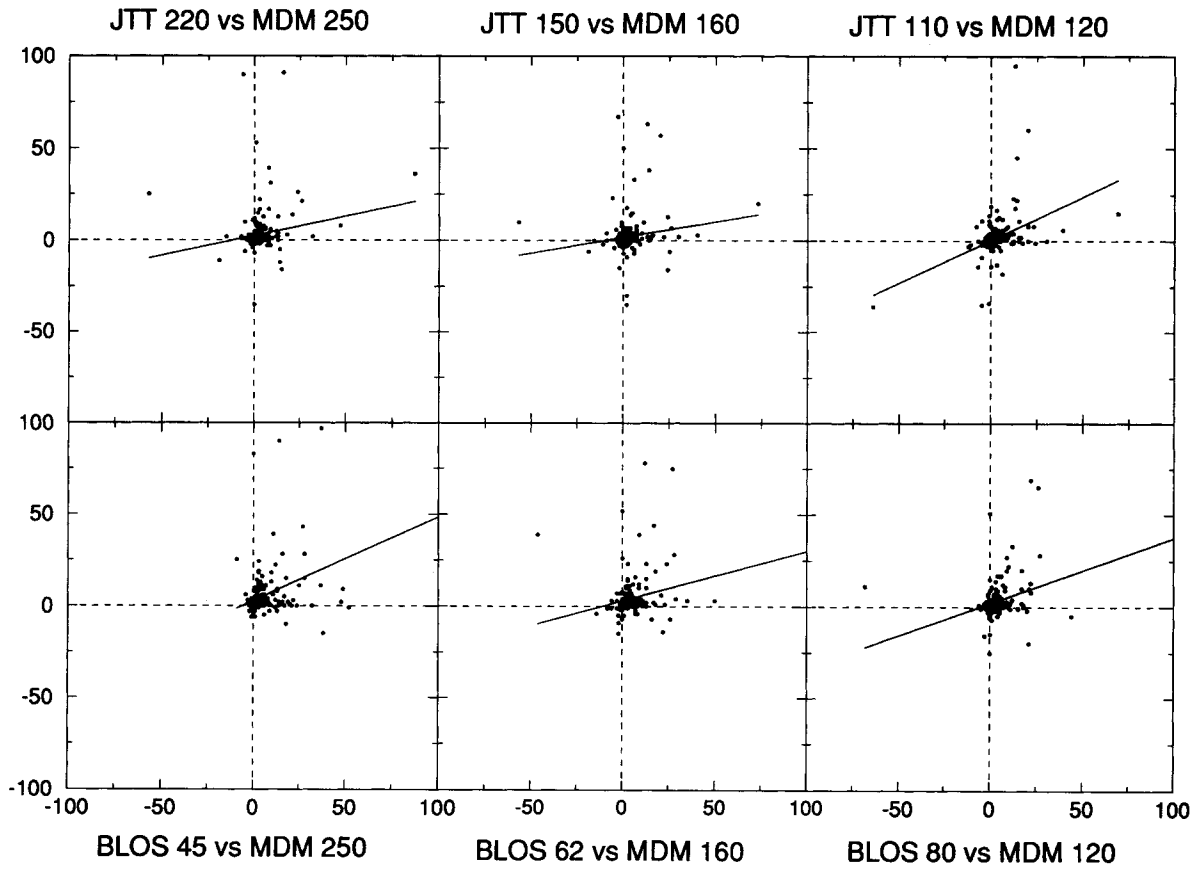

Fig. 7. Scatter plots representing *P*-value comparisons (e.g., Fig. 5) between pairs of matrices. BLAST was used with two queries from each of the 257 Prosite groups searched. For query 1 (*x*-axis) and query 2 (*y*-axis), a single point is shown that represents the number of sequences detected at a lower *P*-value for the first matrix (e.g., JTT PAM 220 in the upper left panel) than for the second matrix (e.g., MDM_{78} PAM 250 in the upper left panel). The linear regression line is shown for each set (generated by the xvgr program from Paul Turner). In each of the lower panels, a single point in the upper right quadrant is not shown because for one of the queries more than 100 sequences obtained lower *P*-values for the BLOSUM matrix than for the corresponding MDM_{78} matrix.

the set that consists of PAM 50 + 150 + 270 covers a comparable range of HSP lengths with comparable efficiencies. Since no high entropy matrix is available for the BLOSUM series that fulfills the 93% efficiency criteria, we tested a set with a higher efficiency (97%), consisting of BLOSUM 45 + 62 + 100. We also constructed a set of three INTERVAL matrices fulfulling the 93% efficiency criteria (INTERVAL 0–24, 24–40, and 40–100%).

One type of comparison was between a 3-mat set and the middle matrix, such as MDM_{78} PAM 30 + 120 + 250 versus MDM_{78} PAM 120 (Fig. 8, top left). In this case, 3-mat was a slight improvement over PAM 120, even when a tax was imposed for combining results of 3 searches. However, in all other cases, the middle matrix alone was better than 3-mat. For the JTT series, PAM 150 slightly outperformed PAM 50 + 150 + 270, for the INTERVAL series, the middle matrix slightly outperformed the 3-mat set, and for the BLOSUM series, BLOSUM 62 strongly outperformed BLOSUM 45 + 62 + 100. The degree of performance of the middle matrix relative to the 3-mat set follows that of the middle matrices relative to one another: BLOSUM 62 > INTERVAL 24–40% ≈ JTT PAM 150 > MDM_{78} PAM 120 (Figs. 2, 3, 5, and data not shown). Our interpretation is that an inaccurate middle matrix provides more opportunity for flanking matrices to detect additional true positive alignments. So, the inaccuracy of MDM_{78} PAM 120 exacerbates its inefficiency in scoring alignments close to the edges of its targeted range. However, the higher inherent accuracy of BLOSUM 62 makes it more accurate near the edges of its range, so fewer undetected true positive alignments are available for flanking matrices to report. Comparisons of the different 3-mat sets to BLOSUM 62 are consistent with this interpretation, in that all do worse relative to BLOSUM 45 + 62 + 100 (Fig. 8, top middle), even for hybrid matrices that include BLOSUM 62 as the middle matrix (Fig. 8, top right and data not shown). This would be the case if all BLOSUM matrices are more accurate in BLAST tests than are the corresponding matrices in each of the other series.

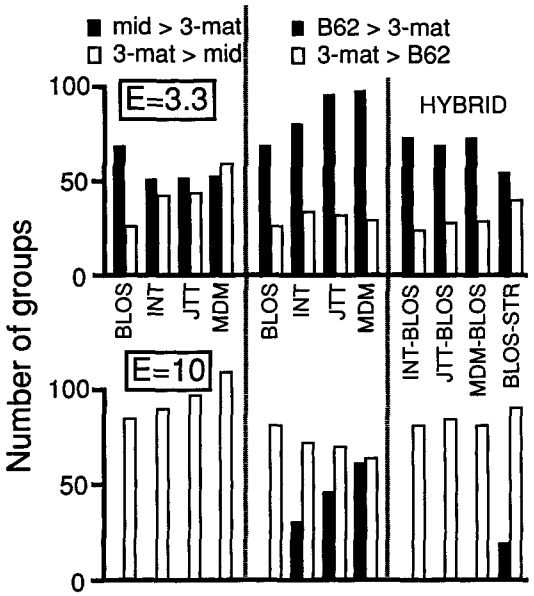

Fig. 8. 3-mat performance for 257 second queries. Each comparison shows the number of groups for which a single matrix detected more true positives at $E=10$ than a 3-mat set at $E=3.33$ **(top)**. The same comparisons are also shown using $E=10$ for both the single matrix and 3-mat **(bottom)**. The set of bars on the left shows comparisons between each 3-mat set and the middle matrix of the set. Single series matrices are (left to right) BLOSUM 45+62+100 (BLOS), INTERVAL 0–24% + 24–40% + 40–100% (INT), JTT PAM 50+150+270 (JTT), MDM_{78} PAM 30+120+250 (MDM). Hybrid matrices are INTERVAL 0–24% + 40–100% + BLOSUM 62 (INT-BLOS), JTT PAM 50+220 + BLOSUM 62 (JTT-BLOS), MDM_{78} PAM 30+250 + BLOSUM 62 (MDM-BLOS), BLOSUM 45+100 + STR (BLOS-STR). Comparisons between each 3-mat set and BLOSUM 62 for the same 3-mat sets are shown by the middle set of bars and for hybrid 3-mat sets are shown for the set of bars on the right.

A much better result was seen when the STR matrix was used as a middle matrix of a hybrid 3-mat series. The best such 3-mat series included BLOSUM 45 and BLOSUM 100 as flanking matrices, with performance that was somewhat worse than that of BLOSUM 62 alone, but better than any 3-mat set tested that did not include the STR matrix (Fig. 8, top right). This hybrid 3-mat set performed better than the 2-mat combination of the STR matrix and BLOSUM 62 (data not shown). To some extent, improvement of this hybrid 3-mat over BLOSUM 45+62+100 might be an artifact of imposing the same tax for combining 3 sets of search results. Since matrices in the same series have more in common than a corresponding hybrid set, the tax should be lower (S. Altschul, personal communication). Nevertheless, the relatively good performance of this hybrid 3-mat set recommends it for applications in which a multiple matrix strategy is advantageous.

DISCUSSION

Extrapolated Matrices Perform Poorly

In sequence alignment applications, choices are made between competing alignments based on the sum of scores over all aligned positions. As pointed out by Altschul, any matrix of values used for scoring alignments is a log-odds matrix, even when the model underlying these scores is not based on observed substitutions.[27] The model underlying a matrix might be implicit, such as for a simple unitary matrix in which the odds are the same for all matches in correct alignments, as are the odds for all mismatches. The model might be explicit, such as for the PAM evolutionary model in which log-odds scores are extrapolated from substitution rates estimated from alignments of closely related proteins.[24] These extrapolated log-odds matrices have become the standard in searching and alignment programs. However, most alignment applications depend upon accurate alignment of the most highly conserved regions of proteins; these are not explicitly represented in the PAM model. Previously, we investigated an explicit model in which log-odds scores derive directly from ungapped blocks representing the most highly conserved regions of proteins.[23] Our BLOSUM matrices outperformed mathematically comparable matrices based on the PAM evolutionary model. In the more comprehensive evaluation of matrix performance reported here, our previous results are confirmed and extended. The single best matrix from that study, BLOSUM 62, was still the best matrix in more sensitive BLAST test, a result confirmed using FASTA. Overall improvement relative to MDM_{78} PAM 120, the BLAST default, was again found to be quite substantial, comparable to the improvement of MDM_{78} PAM 120 relative to a simple unitary matrix. In addition, we found that a matrix derived directly from the structural alignments of Overington and co-workers[30] also performed well relative to the extrapolated matrices based on the PAM model. Like the BLOSUM matrices, the STR matrix is not based on an evolutionary model.

We also presented evidence that improved performance in our tests reflects higher accuracy in scoring distant alignments. The performance of matrices based on the PAM evolutionary model was found to deteriorate with increasing evolutionary distances, suggesting that extrapolation is an inaccurate means of modeling distant relationships. Deterioration with extrapolation was seen both for matrices based on the 1978 MDM_{78} dataset and for those based on the more accurate JTT dataset. Furthermore, using two queries from each group, we were unable to find convincing correlations attributable to improvements in group-specific performance. The lack of detectable group specificity contradicts the intuitive notion that group-specific matrices might be beneficial.[33] In fact, group-specific matrices based on the BLOSUM model perform poorly relative to BLOSUM 62 in detecting members of the same group (unpublished results).

A potential bias in this analysis is that it em-

ployed the same PROSITE 9.0 groups that were used to make the Blocks Database from which the BLOSUM series was derived. To an extent, this concern is addressed by our method of choosing queries, in that preference was given to any sequence that was excluded from the Blocks Database. In addition, we have tested the 65 new groups present in PROSITE 10.0 but not in PROSITE 9.0. These groups do not contribute to the BLOSUM series, although some of them might have contributed to the GCB, JTT, or STR matrices. The results of BLAST tests using 65 first queries and 20 second queries (data not shown) are very similar to those presented in Figures 2 and 5. For example, in the sequence-by-sequence comparisons for 257 second queries shown in Figure 5, BLOSUM 45 was better than MDM PAM 250 in 170 groups and worse in 19 groups, whereas for the 20 new second queries, BLOSUM 45 was better in 14 groups and worse in one group.

We conclude that for scoring local alignments in searching applications, the PAM evolutionary model is inadequate. While our comprehensive tests were limited to local alignment programs, our conclusions are likely to hold for other applications in which it is important to favor correct alignment of the most highly conserved regions of proteins. In these cases, the best performing matrix is BLOSUM 62.

The STR matrix used in these tests shows many differences from matrices in the BLOSUM series. For example, relative to BLOSUM 62, the STR matrix favors matches between hydrophobic residues (M, I, L, V, and F) and disfavors mismatches between hydrophobic and hydrophilic residues (Fig. 1, top). A likely basis for these differences is that BLOSUM represents alignments in conserved regions, whereas the STR matrix includes alignments of more mutable regions that a sequence-based method cannot accurately align. Therefore, it is possible that the STR matrix might be a good choice for applications in which alignment of more mutable regions is important, such as for globally aligning homologous proteins. Unfortunately, any evaluation of this possibility is likely to be severely complicated by the necessity for gap penalties in such applications. Since gaps are far more frequent in mutable than in conserved regions, inaccurate gap penalties could override any improvement gained from the use of an accurate substitution matrix. Comprehensive evaluations of gap penalties, perhaps aided by parallel implementation of a local dynamic programming algorithm[6] are needed. Meanwhile, any one of the accurate matrices tested here is likely to suffice for aligning mutable regions.

For the construction of evolutionary trees based on protein sequence data, the PAM model seems appropriate (assuming that the extrapolations are not too distant). For these applications, the JTT series should be used, since it proved to be more accurate than the widely used MDM_{78} series in our local alignment tests. While it is possible that the more elaborate approach of Gonnet and co-workers[25] can provide useful matrices for evolutionary applications, our tests failed to show that the published GCB matrix provided any improvement over a comparable matrix from Jones and co-workers.[26]

Recommended Matrices for a Multiple Matrix Strategy

We also attempted to evaluate a multiple matrix strategy by combining BLAST search results for three different matrices. Our particular 3-mat implementation proved to have no practical advantage over using BLOSUM 62. There were too few true positive hits that were missed by BLOSUM 62 but detected by one of the flanking matrices to overcome the tax imposed to compensate for the longer list of results. However, in the absence of the tax, several 3-mat sets detected true positives in a large fraction of the groups that were not detected by a single matrix (Fig. 8, bottom). Therefore, a different implementation that avoids the tax might be advantageous. For example, high scoring hits from a search using the middle matrix can be rescored using the flanking matrices.[28] In this way, a true positive alignment that is scored inefficiently by the middle matrix can be promoted without significantly increasing the background of false positives. This strategy is analogous to that implemented in the BLAST3 multiple alignment searching program.[34] Another possible way that a multiple matrix strategy might be useful is where biological clues are available but are not represented in the sequence per se. For instance, if the function of the query is known or suspected, an interesting relationship might be uncovered by screening database hits of marginal statistical significance. Both possible strategies should profit from using the best 3-mat combination identified here, a hybrid consisting of the STR matrix flanked by BLOSUM 45 and BLOSUM 100. This matrix set should efficiently score true positive alignments over a broad range of HSP lengths. Since BLOSUM 45+62+100 might have been too heavily taxed in our tests, this 3-mat set could be an alternative choice.

Evaluation Methodology

The BLAST searching program[3] is well suited for evaluating amino acid substitution matrices. BLAST is sufficiently fast that hundreds of database searches can be carried out per day on an ordinary desktop computer. Unlike other searching programs that achieve speed using a hash table, the BLAST hash table consists of substitution matrix scores rather than exact matches, a feature that probably contributes to its sensitivity to matrix accuracy. Furthermore, BLAST results lists based on Poisson *P*-values provide quantitative measurements of how

well true positive hits score relative to the expected distribution of true negatives. In contrast, choosing an arbitrary percentile rank as was done with FASTA (Ref. 22 and Fig. 3) provides only approximate evaluation of true positive detection that is relatively insensitive to the distribution of true negatives. Also, unlike searching programs that insert gaps[35] or that link diagonals,[2] BLAST does not employ gap penalties, so that the only externally provided parameters that are necessary are present in the substitution matrix. Gap penalties can cause complications because it is not clear just how they should be chosen or scaled relative to substitution scores for any particular application. For example, we detected an anomaly in our FASTA tests that might have occurred because the gap penalties were held constant for different substitution matrices. Presumably, these problems occurred because gap penalties were chosen to maximize performance relative to the MDM_{78} PAM 250 matrix and other program parameters. In spite of these complications, FASTA results were similar to those using BLAST, with BLOSUM 62 outperforming MDM_{78} PAM 250, the FASTA default. Therefore, it is likely that FASTA parameters and gap penalties can be adjusted for even better performance using BLOSUM 62.

While this study was confined to evaluating substitution matrices, our approach should be more generally applicable. For example, sequence filters designed to remove high entropy sequence segments[36,37] can be evaluated using the same programs and Prosite-derived lists developed for this study. We anticipate that comprehensive and fully objective evaluation procedures will replace the anecdotal tests that frequently accompany new searching and alignment tools.

CONCLUSION

Major improvements are possible using substitution matrices based directly on alignments representing distant relationships rather than those based on extrapolations from mutation rates. These overall improvements can be more substantial than performance differences among currently popular searching and alignment programs.[23,38] In contrast to the major effort that is often required to implement searching programs, sometimes involving expensive parallel hardware,[4–6,39] changing a substitution matrix requires only trivial effort. Nevertheless, an inaccurate extrapolated matrix introduced 15 years ago is the current default for most searching and alignment programs. These programs are used by biochemists, geneticists, and molecular biologists who typically are not aware that the inferences they draw might very much depend upon the choice of substitution matrix. These users are rarely given a choice. The comprehensive evaluations presented here argue for a change in the status quo.

ACKNOWLEDGMENTS

We thank Stephen Altschul for advice on multiple matrix strategies and for critical comments on the manuscript. This work was supported by a grant from the National Institutes of Health.

REFERENCES

1. Pearson, W.R., Miller, W. Dynamic programming algorithms for biological sequence comparison. Methods Enzymol. 210:575–601, 1992.
2. Pearson, W.R. Rapid and sensitive sequence comparison with FASTP and FASTA. Methods Enzymol. 183:63–98, 1990.
3. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J. Basic local alignment search tool. J. Mol. Biol. 215: 403–410, 1990.
4. Collins, J.F., Coulson, A.F.W. Significance of protein sequence similarities. Methods Enzymol. 183:474–487, 1990.
5. Vogt, G., Argos, P. Searching for distantly related proteins sequences in large databases by parallel processing on a transputer machine. CABIOS 8:49–55, 1992.
6. Brutlag, D.L., Dautricourt, J.-P., Diaz, R., Fier, J., Stamm, R. BLAZE: An implementation of the Smith-Waterman sequence comparison algorithm on a massively parallel computer. CABIOS, in press, 1993.
7. Feng, D.F., Doolittle, R.F. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. J. Mol. Evol. 25:351–360, 1987.
8. Barton, G.J., Sternberg, M.J. A strategy for the rapid multiple alignment of protein sequences. J. Mol. Biol. 198: 327–337, 1987.
9. Higgins, D.G., Sharp, P.M. CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. Gene 73:237–244, 1988.
10. Corpet, F. Multiple sequence alignment with hierarchical clustering. Nucleic Acids Res. 16:10881–10890, 1988.
11. Lipman, D.J., Altschul, S.F., Kececioglu, J.D. A tool for multiple sequence alignment. Proc. Natl. Acad. Sci. U.S.A. 86:4412–4415, 1989.
12. Smith, R.F., Smith, T.F. Pattern-induced multi-sequence alignment (PIMA) algorithm employing secondary structure-dependent gap penalties for use in comparative protein modelling. Protein Eng. 5:35–41, 1992.
13. Taylor, W.R. Identification of protein structure homology by consensus template alignment. J. Mol. Biol. 188:233–258, 1986.
14. Smith, R.F., Smith, T.F. Automatic generation of primary sequence patterns from sets of related protein sequences. Proc. Natl. Acad. Sci. U.S.A. 87:118–122, 1990.
15. Bairoch, A. PROSITE: A dictionary of sites and patterns in proteins. Nucleic Acids Res. 20:2013–2018, 1992.
16. Smith, H.O., Annau, T.M., Chandrasegaran, S. Finding sequence motifs in groups of functionally related proteins. Proc. Natl. Acad. Sci. U.S.A. 87:826–830, 1990.
17. Henikoff, S., Henikoff, J.G. Automated assembly of protein blocks for database searching. Nucleic Acids Res. 19: 6565–6572, 1991.
18. Bowie, J.U., Luthy, R., Eisenberg, D. A method to identify protein sequences that fold into a known three-dimensional structure. Science 253:164–170, 1991.
19. Wu, C.H., Whitson, G., McLarty, J., Ermongkonchai, A., Chang, T.-C. Protein classification artificial neural system. Prot. Sci. 1:667–677, 1992.
20. Frishman, D., Argos, P. Recognition of distantly related protein sequences using conserved motifs and neural networks. J. Mol. Biol. 228:951–962, 1992.
21. van Heel, M. A new family of powerful multivariate statistical sequence analysis techniques. J. Mol. Biol. 220: 877–887, 1991.
22. Pearson, W.R. Searching protein sequence libraries: comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms. Genomics 11:635–650, 1991.

23. Henikoff, S., Henikoff, J.G. Amino acid substitution matrices from protein blocks. Proc. Natl. Acad. Sci. U.S.A. 89:10915–10919, 1992.
24. Dayhoff, M. "Atlas of Protein Sequence and Structure," Vol. 5, suppl. 3. Washington, D.C.: National Biomedical Research Foundation, 1978: 345–358.
25. Gonnet, G.H., Cohen, M.A., Benner, S.A. Exhaustive matching of the entire protein sequence database. Science 256:1443–1445, 1992.
26. Jones, D.T., Taylor, W.R., Thornton, J.M. The rapid generation of mutation data matrices from protein sequences. CABIOS 8:275–282, 1992.
27. Altschul, S.F. Amino acid substitution matrices from an information theoretic perspective. J. Mol. Biol. 219:555–565, 1991.
28. Altschul, S.F. A protein alignment scoring system sensitive at all evolutionary distances. J. Mol. Evol. 36:290–300, 1993.
29. Dayhoff, M.O., Eck, R.V., eds. "Atlas of Protein Sequence and Structure," Vol. 3. Silver Spring, MD: National Biomedical Research Foundation, 1968: 33.
30. Overington, J., Donnelly, D., Johnson, M.S., Sali, A., Blundell, T.L. Environment-specific amino acid substitution tables: Tertiary templates and prediction of protein folds. Prot. Sci. 1:216–226, 1992.
31. Bairoch, A., Boeckmann, B. The SWISS-PROT protein sequence data bank. Nucleic Acids Res. 20:2019–2022, 1992.
32. Karlin, S., Altschul, S.F. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. Proc. Natl. Acad. Sci. U.S.A. 87: 2264–2268, 1990.
33. George, D.G., Barker, W.C., Hunt, L.T. Mutation data matrix and its uses. Methods Enzymol. 183:333–351, 1990.
34. Altschul, S.F., Lipman, D.J. Protein database searches for multiple alignments. Proc. Natl. Acad. Sci. U.S.A. 87: 5509–5513, 1990.
35. Smith, T.F., Waterman, M.S. Identification of common molecular subsequences. J. Mol. Biol. 147:195–197, 1981.
36. Claverie, J.M., States, D.J. Information enhancement methods for large scale sequence analysis. Comput. Chem., in press, 1993.
37. Wootton, J.C., Federhen, S. Statistics of local complexity in amino acid sequences and sequence databases. Comput. Chem., in press, 1993.
38. Henikoff, S. Comparative sequence analysis:finding genes. In "Biocomputing: Genome Sequence Analysis." D.W. Smith, ed. San Diego: Academic Press. In press, 1993.
39. Miller, P.L., Nadkarni, P.M., Carriero, N.M. Parallel computation and FASTA: confronting the problem of parallel database search for a fast sequence comparison algorithm. CABIOS 7:71–78, 1991.