Nucleic Acids Research, 1997, Vol. 25, No. 17 3389–3402

Gapped BLAST and PSI-BLAST: a new generation of protein database search programs

Stephen F. Altschul*, Thomas L. Madden, Alejandro A. Schäffer[1], Jinghui Zhang, Zheng Zhang[2], Webb Miller[2] and David J. Lipman

National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, MD 20894, USA, [1]Laboratory of Genetic Disease Research, National Human Genome Research Institute, National Institutes of Health, Bethesda, MD 20892, USA and [2]Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802, USA

Received June 20, 1997; Revised and Accepted July 16, 1997

ABSTRACT

The BLAST programs are widely used tools for searching protein and DNA databases for sequence similarities. For protein comparisons, a variety of definitional, algorithmic and statistical refinements described here permits the execution time of the BLAST programs to be decreased substantially while enhancing their sensitivity to weak similarities. A new criterion for triggering the extension of word hits, combined with a new heuristic for generating gapped alignments, yields a gapped BLAST program that runs at approximately three times the speed of the original. In addition, a method is introduced for automatically combining statistically significant alignments produced by BLAST into a position-specific score matrix, and searching the database using this matrix. The resulting Position-Specific Iterated BLAST (PSI-BLAST) program runs at approximately the same speed per iteration as gapped BLAST, but in many cases is much more sensitive to weak but biologically relevant sequence similarities. PSI-BLAST is used to uncover several new and interesting members of the BRCT superfamily.

INTRODUCTION

Variations of the BLAST algorithm (1) have been incorporated into several popular programs for searching protein and DNA databases for sequence similarities. BLAST programs have been written to compare protein or DNA queries with protein or DNA databases in any combination, with DNA sequences often undergoing conceptual translation before any comparison is performed. We will use the *blastp* program, which compares protein queries to protein databases, as a prototype for BLAST

BLAST is a heuristic that attempts to optimize a specifi similarity measure. It permits a tradeoff between speed an sensitivity, with the setting of a 'threshold' parameter, *T*. A highe value of *T* yields greater speed, but also an increased probabilit of missing weak similarities. The BLAST program requires tim proportional to the product of the lengths of the query sequenc and the database searched. Since the rate of change in databa sizes currently exceeds that of processor speeds, compute running BLAST are subjected to increasing load. However, tl conjunction of several new algorithmic ideas allow a new versic of BLAST to achieve improved sensitivity at substantial augmented speed. This paper describes three major refinemen to BLAST.

(i) For increased speed, the criterion for extending word pai has been modified. The original BLAST program seeks sho word pairs whose aligned score is at least *T*. Each such 'hit' is th extended, to test whether it is contained within a high-scori alignment. For the default *T* value, this extension step consum most of the processing time. The new 'two-hit' method requir the existence of two non-overlapping word pairs on the san diagonal, and within a distance *A* of one another, before extension is invoked. To achieve comparable sensitivity, tl threshold parameter *T* must be lowered, yielding more hits th previously. However, because only a small fraction of these h are extended, the average amount of computation requir decreases.

(ii) The ability to generate gapped alignments has been addc The original BLAST program often finds several alignme involving a single database sequence which, when consider together, are statistically significant. Overlooking any one these alignments can compromise the combined result. I introducing an algorithm for generating gapped alignments. becomes necessary to find only one rather than all the ungapp alignments subsumed in a significant result. This allows the parameter to be raised, increasing the speed of the initial datab programming path graph (4). Our approach considers only alignments that drop in score no more than X_g below the best score yet seen. The algorithm is able thereby to adapt the region of the path graph it explores to the data.

(iii) BLAST searches may be iterated, with a position-specific score matrix generated from significant alignments found in round i used for round $i + 1$. Motif or profile search methods frequently are much more sensitive than pairwise comparison methods at detecting distant relationships. However, creating a set of motifs or a profile that describes a protein family, and searching a database with them, typically has involved running several different programs, with substantial user intervention at

where $N = mn$ is the search space size (8–10). If a protein is compared to a whole database rather than a single sequence, n is the database length in residues. Equation 2 may be inverted to yield $S' = \log_2(N/E)$, the normalized score required to achieve a particular E-value. In a typical current database search, a protein of length 250 might be compared to a protein database of 50 000 000 total residues. To achieve a marginally significant E-value of 0.05, a normalized score of ~38 bits is necessary.

While the theory just outlined has not been proved for gapped local alignments and their associated scores, computational experiments strongly suggest that it remains valid (3,12–15). The statistical parameters λ and K, however, are no longer supplied by

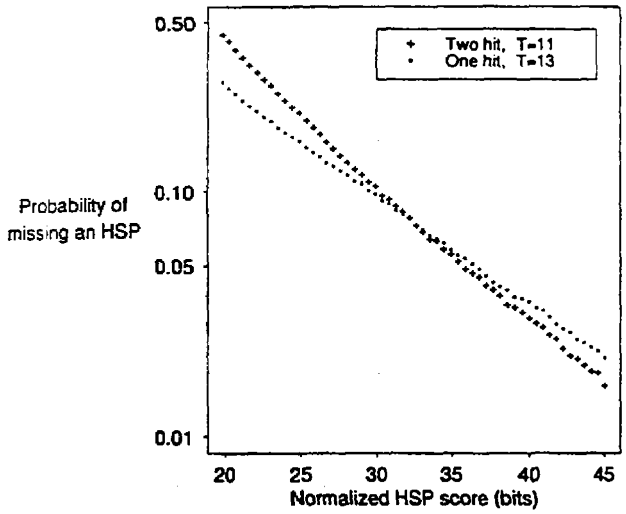


Figure 1. The sensitivity of the two-hit and one-hit heuristics as a function of HSP score. Using the BLOSUM-62 amino acid substitution matrix (18), and the target frequencies q_{ij} implied by equation 3 and the background amino acid frequencies P_i of Robinson and Robinson (20), 100 000 model HSPs were generated for each of the nominal scores 37–92, corresponding to normalized scores 19.9–45.1 bits. It was determined by inspection whether each HSP failed to contain two non-overlapping length-3 word pairs with nominal score at least 11, and within a distance 40 of one another, or a single length-3 word pair with nominal score at least 13. The corresponding probabilities of missing an HSP using the two-hit heuristic with $T = 11$, and the one-hit heuristic with $T = 13$, are plotted as a function of normalized HSP score. The two-hit method is more sensitive for HSPs with score at least 33 bits.

efficiently. Specifically, we choose a window length A, and invoke an extension only when two non-overlapping hits are found within distance A of one another on the same diagonal. Any hit that overlaps the most recent one is ignored. Efficient execution requires an array to record, for each diagonal, the first coordinate of the most recent hit found. Since database sequences are scanned sequentially, this coordinate always increases for successive hits. The idea of seeking multiple hits on the same diagonal was first used in the context of biological database searches by Wilbur and Lipman (19).

Because we require two hits rather than one to invoke an extension, the threshold parameter T must be lowered to retain comparable sensitivity. The effect is that many more single hits are found, but only a small fraction have an associated second hit on the same diagonal that triggers an extension. The great majority of hits may be dismissed after the minor calculation of looking up, for the appropriate diagonal, the coordinate of the most recent hit, checking whether it is within distance A of the current hit's coordinate, and finally replacing the old with the new coordinate. Empirically, the computation saved by requiring fewer extensions more than offsets the extra computation required to process the larger number of hits.

To study the relative abilities of the one-hit and two-hit methods to detect HSPs of varying score, we model proteins using the background amino acid frequencies of Robinson and Robinson (20), and use the BLOSUM-62 substitution matrix (18) for sequence comparison. Given these P_i and s_{ij}, the statistical parameters for ungapped local alignments are calculated to be $\lambda_u = 0.3176$ and $K_u = 0.134$. Using equation 3 above, we may calculate the q_{ij} for which the scoring system is optimized, and employ these target frequencies to generate model HSPs. Finally,

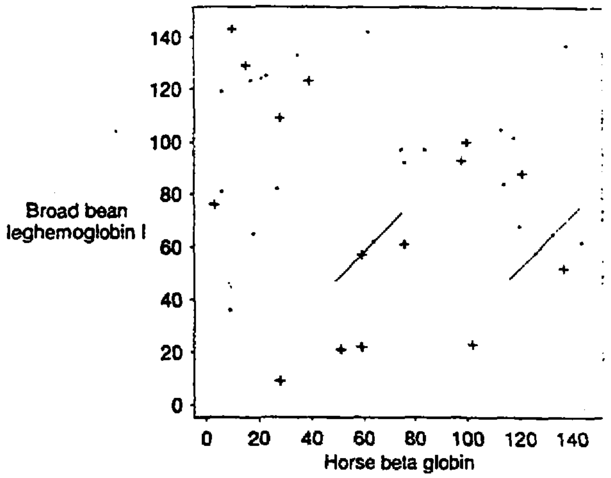


Figure 2. The BLAST comparison of broad bean leghemoglobin I (87: (SWISS-PROT accession no. P02232) and horse β-globin (88) (SWISS-PROT accession no. P02062). The 15 hits with score at least 13 are indicated by plus signs. An additional 22 non-overlapping hits with score at least 11 are indicated by dots. Of these 37 hits, only the two indicated pairs are on the same diagonal and within distance 40 of one another. Thus the two-hit heuristic with $T = 11$ triggers two extensions, in place of the 15 extensions invoked by the one-hi: heuristic with $T = 13$. Because this is just one example, the relative numbers o: hits and extensions at the various settings of T correspond only roughly to the ratios found in a full database search. An ungapped extension of the leftwar: of the two hit pairs yields an HSP with nominal score 45, or 23.6 bits, calculate using λ_u and K_u.

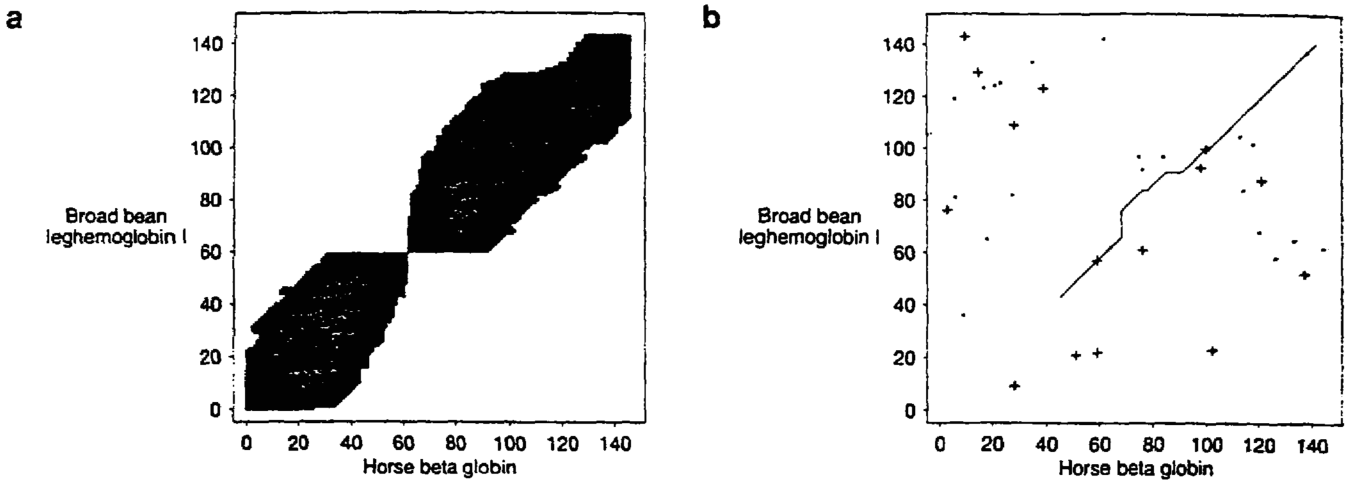
we evaluate the sensitivity of the one-hit and two-hit BLAS heuristics using these HSPs.

The one-hit method will detect an HSP if it somewhere contair a length-W word of score at least T. For $W = 3$ and $T = 13$, Figu: 1 shows the empirically estimated probability that an HSP . missed by this method, as a function of its normalized score. Th two-hit method will detect an HSP if it contains two noi overlapping length-W words of score at least T, with startir positions that differ by no more than A residues. For $W = 3$, $T =$ 1 and $A = 40$, Figure 1 shows the estimated probability that an HS is missed by this method, as a function of its normalized score. Fo HSPs with score at least 33 bits, the two-hit heuristic is moi sensitive.

To analyze the relative speeds of the one-hit and two-h methods, using the parameters studied above, we note that tl two-hit method generates on average ~3.2 times as many hits, b only ~0.14 times as many hit extensions (Fig. 2). Because it tak approximately one ninth as long to decide whether a hit need b extended as actually to extend it, the hit-processing component o the two-hit method is approximately twice as rapid as the sar component of the one-hit method.

TRIGGERING THE GENERATION OF GAPPED ALIGNMENTS

Figure 1 shows that even when using the original one-hit methc with threshold parameter $T = 13$, there is generally no greater th a 4% chance of missing an HSP with score >38 bits. While th would appear sufficient for most purposes, the one-hit default parameter has typically been set as low as 11, yielding a execution time nearly three times that for $T = 13$. Why pay th price for what appears at best marginal gains in sensitivity? Tl



c

```
Leghemoglobin  43 FSFLKDSAGVVDSPKLGAHAEKVFGMVRDSAVQLRATGEVV--LDGKDGS------ 90
                  F  L +   V+ +PK+ AH +KV          L + GE V  LD   G+
Beta globin    45 FGDLSNPGAVMGNPKVKAHGKKV----------LHSFGEGVHHLDNLKGTFAALSE 90

Leghemoglobin  91 IHIQKGVLDP-HFVVVKEALLKTIKEASGDKWSEELSAAWEVAYDGLATAI 140
                  +H  K  +DP +F ++   L+  +    G  ++ EL A+++    G+A A+
Beta globin    91 LHCDKLHVDPENFRLLGNVLVVVLARHFGKDFTPELQASYQKVVAGVANAL 141
```

Figure 3. A gapped extension generated by BLAST for the comparison of broad bean leghemoglobin I (87) and horse β-globin (88). (a) The region of the path graph explored when seeded by the alignment of alanine residues at respective positions 60 and 62. This seed derives from the HSP generated by the leftward of the two ungapped extensions illustrated in Figure 2. The X_g dropoff parameter is the nominal score 40, used in conjunction with BLOSUM-62 substitution scores and a cost of 10 + k for gaps of length k. (b) The path corresponding to the optimal local alignment generated, superimposed on the hits described in Figure 2. The original BLAST program, using the one-hit heuristic with $T = 11$, is able to locate three of the five HSPs included in this alignment, but only the first and last achieve a score sufficient to be reported. (c) The optimal local alignment, with nominal score 75 and normalized score 32.4 bits. In the context of a search of SWISS-PROT (26), release 34 (21 219 450 residues), using the leghemoglobin sequence (143 residues) as query, the E-value is 0.54 if no edge-effect correction (22) is invoked. The original BLAST program locates the first and last ungapped segments of this alignment. Using sum-statistics with no edge-effect correction, this combined result has an E-value of 31 (21,22). On the central lines of the alignment, identities are echoed and substitutions to which the BLOSUM-62 matrix (18) gives a positive score are indicated by a '+' symbol.

reason is that the original BLAST program treats gapped alignments implicitly by locating, in many cases, several distinct HSPs involving the same database sequence, and calculating a statistical assessment of the combined result (21,22). This means that two or more HSPs with scores well below 38 bits can, in combination, rise to statistical significance. If any one of these HSPs is missed, so may be the combined result.

The approach taken here allows BLAST to simultaneously produce gapped alignments and run significantly faster than previously. The central idea is to trigger a gapped extension for any HSP that exceeds a moderate score S_g, chosen so that no more than about one extension is invoked per 50 database sequences. (By equation 2, for a typical-length protein query, S_g should be set at ~22 bits.) A gapped extension takes much longer to execute than an ungapped extension, but by performing very few of them the fraction of the total running time they consume can be kept relatively low.

By seeking a single gapped alignment, rather than a collection of ungapped ones, only one of the constituent HSPs need be located for the combined result to be generated successfully. This means that we may tolerate a much higher chance of missing any single moderately scoring HSP. For example, consider a result involving two HSPs, each with the same probability P of being missed at the hit-stage of the BLAST algorithm, and suppose that we desire to find the combined result with probability at least 0.95. The original algorithm, needing to find both HSPs, requires $2P - P^2 \leq 0.05$, or P less than ~0.025. In contrast, the new algorithm requires only that $P^2 \leq 0.05$, and thus can tolerate P as high as 0.22. This permits the T parameter for the hit-stage of the algorithm to be raised substantially while retaining comparable sensitivity—from $T = 11$ to $T = 13$ for the one-hit heuristic. (The two-hit heuristic described above lowers T back to 11.) As will be discussed below, the resulting increase in speed more than compensates for the extra time required for the rare gapped extension.

In summary, the new gapped BLAST algorithm requires two non-overlapping hits of score at least T, within a distance A of one another, to invoke an ungapped extension of the second hit. If the HSP generated has normalized score at least S_g bits, then a gapped extension is triggered. The resulting gapped alignment is reported only if it has an E-value low enough to be of interest. For example, in the pairwise comparison of Figure 2, the ungapped extension invoked by the hit pair on the left produces an HSP with score 23.6 bits (calculated using λ_u and K_u). This is sufficient to trigger a gapped extension, which generates an alignment with score 32.4 bits (calculated using λ_g and K_g) and E-value of 0.5 (Fig. 3). The original BLAST program locates only the first and last ungapped segments of this alignment (Fig. 3c), and assigns them a combined E-value >50 times greater.

THE CONSTRUCTION AND STATISTICAL EVALUATION OF GAPPED LOCAL ALIGNMENTS

The standard dynamic programming algorithms for pairwise sequence alignment perform a fixed amount of computation per cell of a path graph, whose dimensions are the lengths of the two sequences being compared (23–25). In order to gain speed, database search algorithms such as Fasta (2) and an earlier gapped version of BLAST (3) sacrifice rigor by confining the dynamic programming to a banded section of the full path graph (4), chosen to include a region of already identified similarity. One problem with this approach is that the optimal gapped alignment may stray beyond the confines of the band explored. As the width of the band is increased to reduce this possibility, the speed advantage of the algorithm is vitiated.

We have accordingly taken a different heuristic approach to constructing gapped local alignments, which is a simple generalization of BLAST's method for constructing HSPs. The central idea is to consider only cells for which the optimal local alignment score falls no more than X_g below the best alignment score yet found. Starting from a single aligned pair of residues, called the *seed*, the dynamic programming proceeds both forward and backward through the path graph (Zheng Zhang *et al.*, manuscript in preparation) (Figs 3a and 4). The advantage of this approach is that the region of the path graph explored adapts to the alignment being constructed. The alignment can wander arbitrarily many diagonals away from the seed, but the number of cells expanded on each row tends to remain limited, and may even shrink to zero before a boundary of the path graph is encountered (Fig. 4). The X_g parameter serves a similar function to the band-width parameter of the earlier heuristic, but the region of the path graph it implicitly specifies be explored is in general more productively chosen.

An important element for this heuristic is the intelligent choice of a seed. Given an HSP whose score is sufficiently high that it triggers a gapped extension, how does one choose a residue pair to force into alignment? While more sophisticated approaches are

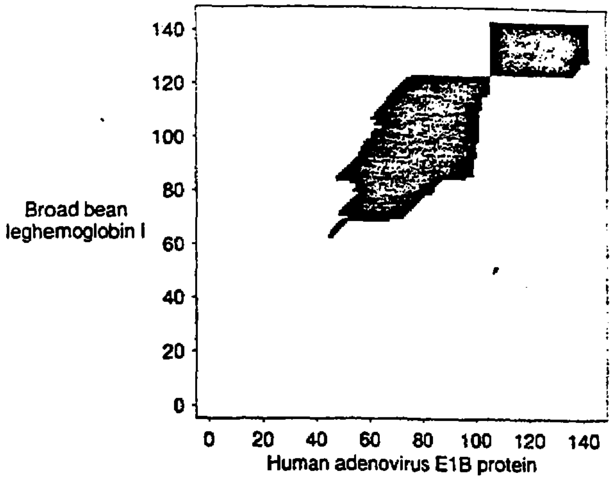


Figure 4. The path graph region explored by BLAST during a gapped extension for the comparison of broad bean leghemoglobin I and the E1B protein small T-antigen from human adenovirus type 4 (89) (SWISS-PROT accession no. P10406). The X_g dropoff parameter is the nominal score 40, used in conjunction with BLOSUM-62 substitution scores and $10 + k$ gap costs. The 22.7 bit HSP that triggers this extension, involving leghemoglobin residues 119–140 and adenovirus residues 101–122, is merely a random similarity, and not part of a larger and higher-scoring alignment. The gapped extension is seeded by the alignment of residues 124 and 106. The optimal alignment score through points in the path graph drops steadily as one moves beyond the triggering HSP, and the reverse extension terminates before the beginning of either protein is reached. A total of 2766 path graph cells are explored, with the reverse extension accounting for 2047 of these cells.

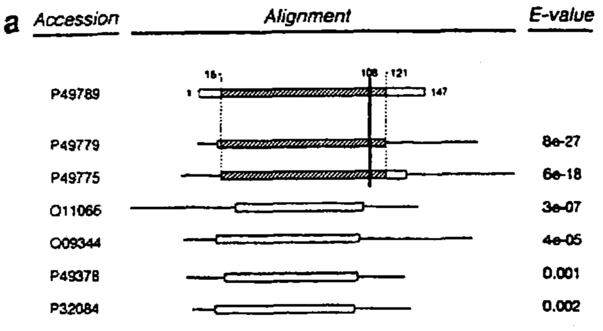
path graph cells, so that a typical two-way gapped extension that does not encounter the end of either sequence is expected to involve ~4000 cells. Because S_g is set so that a gapped extension is invoked less than once per 50 database sequences, fewer than 80 cells need be explored per database sequence.

The execution time required for a gapped extension is ~500 times that for an ungapped extension. However, by triggering

The times required by various steps of the BLAST algorithm vary substantially from one query and one database to another. Table 1 shows typical relative times spent by the original and the gapped BLAST programs on various algorithmic stages. The

[redacted]

iteration takes little more than the same time to run. In related work, Henikoff and Henikoff (39) have described how, short of modifying BLAST so that it may operate on a position-specific score matrix, a single artificial sequence that approximates such



b

```
Histidine triad protein   15 VFLKTELSFALVNRKPVVPGHVLVCPLRPVERFHDLRPDEVADLF  59
                             + ++TE   ALV    + P   L+ P   V+R  +L  ++  DL
Uridylyltransferase      213 IVVETEHWIALVPYWAIWPFETLLLPKTHVKRLTELSDEQSKDLA 257


Histidine triad protein   60 QTTQRVGTVVEKHFHGT-SLTFSMQDGPEAGQTVKH--VHVHVLP 101
                                +++ T  + F  +  +       P  G+  +H  +H H  P
Uridylyltransferase      258 VILKKLTTKYDNLFETSFPYSMGFHAAPFNGEDNEHWQLHAHFYP 302


Histidine triad protein  102 R--KAGDFHRNDSIYEELQKHDKEDFPASWRSEEEMAAEAAALRV 144
                                ++     +    YE L ++              + +- AE AA R+
Uridylyltransferase      303 PLLRSATVRKFMVGYEMLGEN-----------QRDLTAEQAAERL 336
```

c

```
Histidine triad protein   25 LVNRKPVVPGHVLVCPLRPVERFHDLRPDEVADLFQTTQRVGT  67
                             L+N+ PV+PGH L+         +   L P ++      T  ++
Phosphorylase             91 LLNKFPVIPGHTLLVTNEYQHQTDALTPTDL----LTAYKLLC 129


Histidine triad protein   68 VVEKHFHGTSLTFSMQDGPEAGQTVKHVHVHVL--PRKAGDF 107
                              ++                GP +G ++ H H+ +L  P K   F
Phosphorylase            130 ALDNEESDKRHMVFYNSGPASGSSLDHKHLQILQMPEKFVTF 171
```

Figur‹ 5. (a) The multiple alignment generated by PSI-BLAST when the human fragile histidine triad (HIT) protein (61) (SWISS-PROT accession no. P49789) is compa:ed to SWISS-PROT. All pairwise local alignments have E-value ≤0.01, and are identified in SWISS-PROT as belonging to the HIT family. Thick bars within the six database sequences represent segments that align with various segments from the query. In constructing sequence weights for the indicated multiple alignment

[redacted]

smaller than 21, and therefore perhaps a good enough approximation for our purposes. As will be seen, it is not the absolute value of N_C that is important, but rather its relative value from one

performed on a query consisting of a position-specific matrix rather than a simple sequence. The same holds for the ungapped and gapped extension steps of BLAST. One important issue is

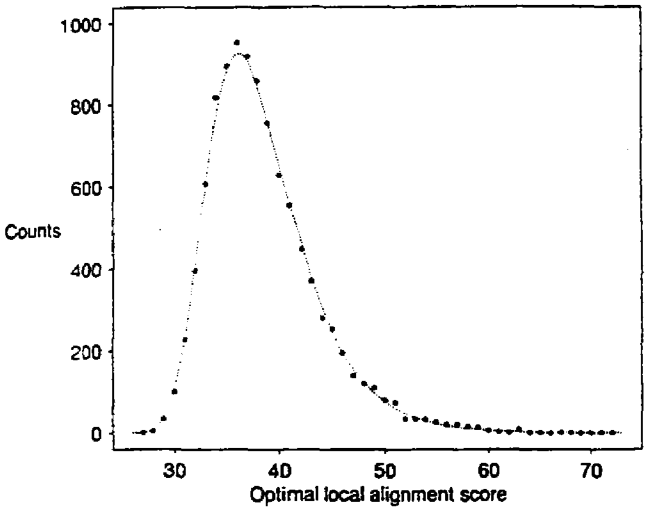


Figure 6. The distribution of optimal local alignment scores from the comparison of a position-specific score matrix with 10 000 random protein sequences. The score matrix was constructed by PSI-BLAST from the 128 local alignments with E-value ≤0.01 found in a search of SWISS-PROT using as query the length-567 influenza A virus hemagglutinin precursor (27) (SWISS-PROT accession no. P03435). The random sequences, each of length 567, were generated using the amino acid frequencies of Robinson and Robinson (20). Optimal local alignment scores were calculated using the position-specific matrix in conjunction with $10 + k$ gap costs. The extreme value distribution that best fits the data (3,15) is plotted. A χ^2 goodness-of-fit test with 34 degrees of freedom has value 41.8, corresponding to a P-value of 0.20.

lowest E-value found, as well as the number of shuffled sequences yielding E-values ≤1 and 10. For comparison, we performed the identical shuffled-database test on the gapped and original versions of BLAST. To reduce the probability that high-scoring alignments were missed due to the heuristic nature of the algorithms, we performed these tests with $T = 9$ rather than the default value of 11. The results are given in Table 2. For the 11 queries, the median of the low PSI-BLAST E-values was 0.87, which corresponds to a median P-value of 0.58 (8,9). The mean numbers of shuffled database sequences with E-values <1 and 10 were 1.0 and 8.7, respectively, within 20% of the expected values of 1.0 and 10.0. The equivalent tests for the ungapped and gapped versions of BLAST also yielded results that diverged from theory by <50%.

The ability to estimate with reasonable accuracy the significance of gapped local matrix-sequence alignments permits us to automate the construction of position-specific score matrices during multiple iterations of the PSI-BLAST program. After each iteration, we generate a new multiple alignment simply by collecting those alignments with E-value lower than a defined threshold. An interactive version of PSI-BLAST allows the user to override either the inclusion or exclusion of specific local alignments. Once a given database sequence has been used in the generation of a position-specific score matrix, low E-values for this sequence are virtually guaranteed in future iterations, for the sequence is to a certain extent being compared with itself. The biological relevance of PSI-BLAST output thus depends critically on avoiding the inappropriate inclusion of sequences in the multiple alignment constructed. Specifically, the utility of the score matrix produced is immediately vitiated by the inclusion of any alignment involving a region of highly biased amino acid composition (57,58).

Table 2. The comparison of various query sequences with a shuffled version of SWISS-PROT

Protein family	SWISS-PROT	Original BLAST			Gapped BLAST			PSI-BLAST		
	accession no.	Low	No. of seqs		Low	No. of seqs		Low	No. of seqs	
	of query	E-value	with E-value		E-value	with E-value		E-value	with E-value	
			≤1	≤10		≤1	≤10		≤1	≤10
Serine protease	P00762	0.86	1	7	3.0	0	4	0.94	1	8
Serine protease inhibitor	P01008	3.9	0	4	0.078	1	9	1.5	0	9
Ras	P01111	3.4	0	8	3.4	0	7	1.1	0	9
Globin	P02232	2.4	0	7	2.8	0	5	8.2	0	2
Hemagglutinin	P03435	0.11	2	11	0.46	3	16	0.87	1	8
Interferon α	P05013	2.4	0	6	0.27	2	4	0.11	2	11
Alcohol dehydrogenase	P07327	1.5	0	2	0.80	1	5	1.5	0	9
Histocompatibility antigen	P10318	0.91	1	7	0.13	1	7	0.0031	2	6
Cytochrome P450	P10635	0.84	2	5	8.5	0	3	0.46	1	15
Glutathione transferase	P14942	1.0	1	10	3.3	0	3	0.30	2	9
H^+-transporting ATP synthase	P20705	0.012	1	8	0.26	2	14	0.79	2	10
Average (median or mean)		1.0	0.7	6.8	0.80	0.9	7.0	0.87	1.0	8.7

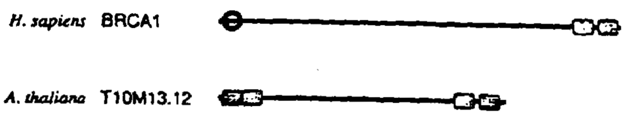
The original and gapped BLAST comparisons use BLOSUM-62 substitution scores (18). All three programs use threshold T parameter set to 9, but the gapped BLAST and PSI-BLAST programs use the two-hit method to trigger ungapped extensions. The original BLAST program has the X dropoff parameter set to nominal score 23. The gapped BLAST and PSI-BLAST comparisons charge gaps of length k a cost of $10 + k$. They have X_u set to 16, and X_g set to 40 for the database search stage and to 67 for the output stage of the algorithms. Gapped alignments are triggered by a score corresponding to ~22 bits. For PSI-BLAST, the query is first compared to the SWISS-PROT database, and the position-specific score matrix generated is then compared to a shuffled version of SWISS-PROT. The median is used for the average of the low E-values, and the mean otherwise.

Table 3. The number of SWISS-PROT sequences yielding alignments with E-value ≤0.01, and relative running times, for Smith–Waterman and various versions of BLAST

Protein family	Query	Smith–Waterman	Original BLAST	Gapped BLAST	PSI-BLAST
Serine protease	P00762	275	273	275	286
Serine protease inhibitor	P01008	108	105	108	111
Ras	P01111	255	249	252	375
Globin	P02232	28	26	28	623
Hemagglutinin	P03435	128	114	128	130
Interferon α	P05013	53	53	53	53
Alcohol dehydrogenase	P07327	138	128	137	160
Histocompatibility antigen	P10318	262	241	261	338
Cytochrome P450	P10635	211	197	211	224
Glutathione transferase	P14942	83	79	81	142
H$^+$-transporting ATP synthase	P20705	198	191	197	207
Normalized running time		36	1.0	0.34	0.87

[redacted]

the rat GalT protein (62) has the only marginally significant *E*-value of 0.012. A PSI-BLAST search, using the score matrix generated from the six alignments illustrated in Figure 5a, can immediately cement confidence in the biological relevance of this similarity. The



[redacted]

for using such costs is only a minor variant on that for traditional affine gap costs. In many cases, the new gap costs generate local alignments that are both more accurate and more statistically significant (86). These costs are potentially of particular value for use with PSI-BLAST, because by imposing alignment only where it is justified, they may lead to the construction of more sensitive position-specific score matrices. Whether it is desirable to use generalized affine gap costs as the default for general purpose database searches awaits detailed empirical study.

Table 4. PSI-BLAST protein database search results using the C-terminus of BRCA1 as query

Protein	Species	GenBank ID number	PSI-BLAST iteration	*E*-value
BARD	*Homo sapiens*	1710175	0	2e-06
T10M13.12[a]	*Arabidopsis thaliana*	2104545	1	4e-06
F26D2.b[b]	*Caenorhabditis elegans*	1914176	1	4e-04
KIAA0259[a]	*H.sapiens*	1665785	1	0.001
F37D6.1	*C.elegans*	1418521	2	4e-06
C19G10.07	*Schizosaccharomyces pombe*	1723501	2	6e-05
KIAA0170	*H.sapiens*	1136400	2	0.002
53BP1	*H.sapiens*	488592	2	0.008
T13F2.3[a]	*C.elegans*	1667334	3	2e-07
K04C2.4	*C.elegans*	470351	3	3e-07
T19E10.1	*C.elegans*	1067065	4	7e-04
Rad4/Cut5	*S.pombe*	730470	4	0.002
REV1	*Saccharomyces cerevisiae*	132409	4	0.003
ECT2	*Mus musculus*	423597	5	1e-04
XRCC1	*M.musculus*	627867	5	6e-04
Crb2	*S.pombe*	1449177	5	0.002
RAP1	*S.cerevisiae*	173558	5	0.006
TcEST030[c]	*Trypanosoma cruzi*	1536857	6	0.001
DPB11	*S.cerevisiae*	1352999	6	0.001
L8543.18	*S.cerevisiae*	1078075	6	0.010
SPAC6G9.12[a]	*S.pombe*	1644324	7	4e-04
YM8021.03	*S.cerevisiae*	1078533	7	0.005
YHR154w	*S.cerevisiae*	731729	7	0.008
C36A4.8[a]	*C.elegans*	1657667	7	0.010
UNE452	*S.cerevisiae*	1151000	8	8e-04
DNA ligase IV	*H.sapiens*	1706482	8	0.008
CDC9	*Candida albicans*	1706483	9	0.006
DNA ligase	*Thermus scotoductus*	1352293	10	0.010
GNF1	*Drosophila melanogaster*	544404	11	0.004
mutT[c]	*M.jannaschii*	2129134	15	0.008
RAD9	*S.cerevisiae*	131817	7	0.74
RAP1 homolog	*K.lactis*	422087	9	0.21
ZK675.2	*C.elegans*	599712	13	3.5
D90904[a]	*Synechocystis* sp.	1652299	15	0.17
TDT	*Mus domestica*	2149634	15	0.46
YGR103w	*S.cerevisiae*	1723693	16	0.017
Pescadillo[a]	*H.sapiens*	2194203	16	0.017
PPOL	*Sarcophaga peregrina*	1709741	16	0.060

Iteration zero refers to the initial BLAST run, using the 215 C-terminal residues of BRCA1 (68) (SWISS-PROT accession no. P38398) as query. Subsequent PSI-BLAST iterations use derived position-specific score matrices in place of the query. The score matrix for iteration $i + 1$ is constructed from alignments achieving an *E*-value ≤0.01 for iteration i. For each protein, the *E*-value is that returned during the PSI-BLAST iteration indicated, and precedes the protein's use for score matrix construction. Only one representative is listed for families of closely related proteins. On its 16th iteration PSI-BLAST uncovered no new proteins with *E*-value ≤0.01, and therefore ceased iteration. At the end of the table are shown BRCT proteins returned by PSI-BLAST with *E*-value >0.01 but ≤10, listed for the iteration in which they achieved their lowest *E*-value.

[a] Recent additions to the database, first identified as BRCT proteins here.

[b] The *C.elegans* F26D2.b protein (74) while a recent addition to the databases, is a close homolog of the previously recognized (66,67) family of *C.elegans* BRCT proteins containing, for example, F37A4.4 (90).

[c] The trypanosome EST (70) and the *M.jannaschii* mutT protein (71) are the only likely false positives.

Position-specific score matrices as input to PSI-BLAST

PSI-BLAST performs three distinct operations: it constructs a multiple alignment from BLAST output data; it processes this alignment into a position-specific score matrix; and it uses this

retains the ability to report accurate statistics, per iteration runs in times not much greater than gapped BLAST, and can be used both iteratively and fully automatically. These developments should enhance significantly the utility of database search methods to the

[redacted]

29 Staden,R. (1984) *Nucleic Acids Res.*, **12**, 505–519.

30 Schneider,T.S., Stormo,G.D., Gold,L. and Ehrenfeucht,A. (1986) *J. Mol. Biol.*, **188**, 415–431.

31 Taylor,W.R. (1986) *J. Mol. Biol.*, **188**, 233–258.

32 Berg,O.G. and von Hippel,P.H. (1987) *J. Mol. Biol.*, **193**, 723–750.

33 Dodd,I.B. and Egan,J.B. (1987) *J. Mol. Biol.*, **194**, 557–564.

34 Gribskov,M., McLachlan,A.D. and Eisenberg,D. (1987) *Proc. Natl. Acad. Sci. USA*, **84**, 4355–4358.

35 Patthy,L. (1987) *J. Mol. Biol.*, **198**, 567–577.

36 Stormo,G.D. and Hartzell,G.W. III (1989) *Proc. Natl. Acad. Sci. USA*, **86**, 1183–1187.

37 Tatusov,R.L., Altschul,S.F. and Koonin,E.V. (1994) *Proc. Natl. Acad. Sci. USA*, **91**, 12091–12095.

38 Yi,T.-M. and Lander,E.S. (1994) *Protein Sci.*, **3**, 1315–1328.

39 Henikoff,S. and Henikoff,J.G. (1997) *Protein Sci.*, **6**, 698–705.

40 Bucher,P., Karplus,K., Moeri,N. and Hofmann,K. (1996) *Comput. Chem.*, **20**, 3–23.

41 Sellers, P.H. (1980) *J. Algorithms*, **1**, 359–373.

42 Lawrence,C.E., Altschul,S.F., Boguski,M.S., Liu,J.S., Neuwald,A.F. and Wootton,J.C. (1993) *Science*, **262**, 208–214.

43 Altschul,S.F., Carroll,R.J. and Lipman,D.J. (1989) *J. Mol. Biol.*, **207**, 647–653.

44 Sibbald,P.R. and Argos,P. (1990) *J. Mol. Biol.*, **216**, 813–818.

45 Sander,C. and Schneider,R. (1991) *Proteins*, **9**, 56–68.

46 Gerstein,M., Sonnhammer,E.L. and Chothia,C. (1994) *J. Mol. Biol.*, **236**, 1067–1078.

47 Henikoff,S. and Henikoff,J.G. (1994) *J. Mol. Biol.*, **243**, 574–578.

48 Thompson,J.D., Higgins,D.G. and Gibson,T.J. (1994) *Comput. Appl. Biosci.*, **10**, 19–29.

49 Eddy,S.R., Mitchison,G. and Durbin,R. (1995) *J. Comput. Biol.*, **2**, 9–23.

50 Gotoh,O. (1995) *Comput. Appl. Biosci.*, **11**, 543–551.

51 Krogh,A. and Mitchison,G. (1995) In Rawlings,C., Clark,D., Altman,R., Hunter,L., Lengauer,T. and Wodak,S. (eds.), *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, Menlo Park, CA, pp. 215–221.

52 Henikoff,J.G. and Henikoff,S. (1996) *Comput. Appl. Biosci.*, **12**, 135–143.

53 Brown,M., Hughey,R., Krogh,A., Mian,I.S., Sjölander,K. and Haussler,D. (1993) In Hunter,L., Searls,D. and Shavlik,J. (eds.), *Proceedings of the First International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, Menlo Park, CA, pp. 47–55.

54 Bailey,T.L. and Gribskov,M. (1996) In States,D.J., Agarwal,P., Gaasterland,T., Hunter,L. and Smith,R. (eds.), *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, Menlo Park, CA, pp. 15–24.

55 Karplus,K. (1995) In Rawlings,C., Clark,D., Altman,R., Hunter,L., Lengauer,T. and Wodak,S. (eds.), *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, Menlo Park, CA, pp. 188–196.

56 Sjölander,K., Karplus,K., Brown,M., Hughey,R., Krogh,A., Mian,I.S. and Haussler,D. (1996) *Comput. Appl. Biosci.*, **12**, 327–345.

57 Wootton,J.C. and Federhen,S. (1993) *Comput. Chem.*, **17**, 149–163.

58 Altschul,S.F., Boguski,M.S., Gish,W. and Wootton,J.C. (1994) *Nature Genet.*, **6**, 119–129.

59 Benson,D.A., Boguski,M.S., Lipman,D.J. and Ostell,J. (1997) *Nucleic Acids Res.*, **25**, 1–6.

60 Holm,L. and Sander,C. (1997) *Structure*, **5**, 165–171.

61 Ohta,M., Inoue,H., Cotticelli,M.G., Kastury,K., Baffa,R., Palazzo,J., Siprashvili,Z., Mori,M., McCue,P., Druck,T., Croche,C.M. and Huebner,K (1996) *Cell*, **84**, 587–597.

62 Heidenreich,R.A., Mallee,J. and Segal,S. (1993) *DNA Seq.*, **3**, 311–318.

63 Maskell,D.J., Szabo,M.J., Deadman,M.E. and Moxon,E.R. (1992) *Mol. Microbiol.*, **6**, 3051–3063.

64 Plateau,P., Fromant,M., Schmitter,J.M., Buhler,J.M. and Blanquet,S. (1989) *J. Bacteriol.*, **171**, 6437–6445.

65 Koonin,E.V., Altschul,S.F. and Bork,P. (1996) *Nature Genet.*, **13**, 266–268.

66 Bork,P., Hofmann,K., Bucher,P., Neuwald,A.F., Altschul,S.F. and Koonin,E.V. (1997) *FASEB J.*, **11**, 68–76.

67 Callebaut,I. and Mornon,J.-P. (1997) *FEBS Lett.*, **400**, 25–30.

68 Miki,Y., Swensen,J., Shattuck-Eidens,D., Futreal,P.A., Harshman,K., Tavtigian,S., Liu,Q., Cochran,C., Bennett,L.M., Ding,W., *et al.* (1994) *Science*, **266**, 66–71.

69 Wu,L.C., Wang,Z.W., Tsan,J.T., Spillman,M.A., Phung,A., Xu,X.L., Yang,M.C., Hwang,L.Y., Bowcock,A.M. and Baer,R. (1996) *Nature Genet.*, **14**, 430–440.

70 Tanaka,T. and Tanaka, M. (1996) DDBJ accession no. D87228.

71 Bult,C.J., White,O., Olsen,G.J., Zhou,L., Fleischmann,R.D., Sutton,G.G., Blake,J.A., FitzGerald,L.M., Clayton,R.A., Gocayne,J.D., *et al.* (1996) *Science*, **273**, 1058–1073.

72 Johnson,A.F., de la Bastide,M., Lodhi,M., Hoffman,J., Hasegawa,A., Gnoj,L., Gottesman,T., Granat,S., Hameed,A., Kaplan,N., *et al.* (1997) GenBank accession no. 2104545.

73 Nagase,T., Seki,N., Ishikawa,K., Ohira,M., Kawarabayasi,Y., Ohara,O., Tanaka,A., Kotani,H., Miyajima,N. and Nomura,N. (1996) *DNA Res.*, **3**, 321–329.

74 Wilson,R., Ainscough,R., Anderson,K., Baynes,C., Berks,M., Bonfield,J., Burton,J., Connell,M., Copsey,T., Cooper,J., *et al.* (1994) *Nature*, **368**, 32–38

75 Barrell,B.G., Rajandream,M.A. and Connor,R.E. (1996) EMBL accession no. Z81317.

76 Johnston,M., Andrews,S., Brinkman,R., Cooper,J., Ding,H., Du,Z., Favello,A., Fulton,L., Gattung,S., Greco,T., *et al.* (1994) GenBank accession no. 662142.

77 Kaneko,T., Sato,S., Kotani,H., Tanaka,A., Asamizu,E., Nakamura,Y., Miyajima,N., Hirosawa,M., Sugiura,M., Sasamoto,S., *et al.* (1996) *DNA Res.*, **3**, 109–136.

78 Tabata,S. (1996) GenBank accession no. 1651660.

79 Allende,M.L., Amsterdam,A., Becker,T., Kawakami,K., Gaiano,N. and Hopkins,N. (1996) *Genes Dev.*, **10**, 3141–3155.

80 Hernandez,K., Weber,N., Wipfli,P. and Schmidheini,T. (1996) EMBL accession no. Z72888.

81 Sonnhammer,E.L. and Durbin,R. (1994) *Comput. Appl. Biosci.*, **10**, 301–307.

82 Gotoh,O. (1982) *J. Mol. Biol.*, **162**, 705–708.

83 Fitch,W.M. and Smith,T.F. (1983) *Proc. Natl. Acad. Sci. USA*, **80**, 1382–1386.

84 Altschul,S.F. and Erickson,B.W. (1986) *Bull. Math. Biol.*, **48**, 603–616.

85 Myers,E.W. and Miller,W. (1988) *Comput. Appl. Biosci.*, **4**, 11–17.

86 Altschul,S.F. *Proteins*, in press.

87 Richardson,M., Dilworth,M.J. and Scawen,M.D. (1975) *FEBS Lett.*, **51**, 33–37.

88 Matsuda,G., Maita,T., Braunitzer,G. and Schrank,B. (1980) *Hoppe-Seyler Z. Physiol. Chem.*, **361**, 1107–1116.

89 Tokunaga,O., Yaegashi,T., Lowe,J., Dobbs,L. and Padmanabhan,R. (198 *Virology*, **155**, 418–433.

90 Fulton,L. and Waterston,R. (1995) GenBank accession no. 1176713.

An iterative algorithm for finding the solution was described in Huang, Blostein, and Margerum [3]; a noniterative algorithm based on quaternions in Faugeras and Hebert [4]. In this correspondence, we describe a new noniterative algorithm which involves the singular value decomposition (SVD) of a 3 × 3 matrix. The computer time requirements of the three algorithms are compared.

After the submission of our correspondence, it was brought to our attention that an algorithm similar to ours had been developed independently by Professor B. K. P. Horn, M.I.T., but not published.

II. Decoupling Translation and Rotation

B. Derivation

Expanding the right-hand side of (9),

$$\begin{aligned}\Sigma^2 &= \sum_{i=1}^{N} (q_i' - Rq_i)^t (q_i' - Rq_i)\\ &= \sum_{i=1}^{N} (q_i'^t q_i' + q_i^t R^t R q_i - q_i'^t R q_i - q_i^t R^t q_i')\\ &= \sum_{i=1}^{N} (q_i'^t q_i' + q_i^t q_i - 2q_i'^t R q_i).\end{aligned}$$

Therefore, minimizing Σ^2 is equivalent to maximizing

$$\begin{aligned}F &= \sum_{i=1}^{N} q_i'^t R q_i\\ &= \text{Trace}\left(\sum_{i=1}^{N} R q_i q_i'^t\right) = \text{Trace}\,(RH) \qquad (14)\end{aligned}$$

where

$$H \triangleq \sum_{i=1}^{N} q_i q_i'^t. \qquad (11)$$

Lemma: For any positive definite matrix AA^t, and any orthonormal matrix B,

$$\text{Trace}\,(AA^t) \geq \text{Trace}\,(BAA^t).$$

Proof of Lemma: Let a_i be the ith column of A. Then

$$\text{Trace}\,(BAA^t) = \text{Trace}\,(A^t BA)$$

[redacted]

Therefore, the SVD algorithm may give either. We shall see presently that this situation can be easily resolved.

3) $\{q_i\}$ are colinear—There are infinitely many rotations and reflections which will make $\Sigma^2 = 0$.

Now we come back to the coplanar case. From examining the elements of the 3×3 matrix H, it can readily be shown that the points $\{q_i\}$ are coplanar, if and only if one of the three singular values of H is zero. Let the three singular values be $\lambda_1 > \lambda_2 > \lambda_3 = 0$. Then

$$H = \lambda_1 u_1 v_1^t + \lambda_2 u_2 v_2^t + 0 \cdot u_3 v_3^t \qquad (17)$$

where u_i and v_i are columns of U and V, respectively. Note that changing the sign of u_3 or v_3 will not change H. Therefore, if $X = VU^t$ minimizes Σ^2, so does $X' = V'U^t$ where

$$V' = [v_1, v_2, -v_3]. \qquad (18)$$

If X is a reflection, then X' is a rotation, and vice versa. Thus, if the SVD algorithm gives a solution X with $\det(X) = -1$, we form $X' = V'U^t$ which is the desired rotation.

We mention, in passing, that the points $\{q_i\}$ are colinear, if and only if, two of the three singular values of H are equal.

V. Degeneracy: Noisy Case

If either $\{q_i\}$ or $\{q_i'\}$ are coplanar, then it can readily be shown that the discussion on the coplanar case in Section IV is still valid, except of course now the minimum of Σ^2 is no longer zero. Hence, if the SVD algorithm gives a reflection $X = VU^t$, we can form the desired rotation $X' = V'U^t$. A special case of interest is when $N = 3$. Then both $\{q_i\}$ and $\{q_i'\}$ are coplanar point sets.

[redacted]

TABLE I
VAX 11/780 CPU TIME PER RUN IN ms

Number of Point Correspondences	Method Used		
	SVD	Quaternion	Iterative
3	54.6	26.6	126.8 (25)
7	41.6	32.4	108.2 (12)
11	37.0	41.0	105.2 (8)
16	39.4	45.6	94.2 (5)
20	40.4	45.2	135.0 (6)
30	44.2	48.3	111.0 (6)

used in finding the SVD (subroutine LSVDF) and in doing the eigen analysis (subroutine EIGRS) for the quaternion method. For the iterative method, the initial guess solution was zero in all cases.

We observe that the computer time requirements of the SVD and the quaternion algorithms are comparable, while the time for the iterative method is much longer. However, in the iterative method, the solutions were calculated to 7-digit accuracy. If we can accept 10 percent accuracy, then the number of iterations are reduced by a factor of 2 to 3. Furthermore, the rate of convergence can be increased by overrelaxation.

References

[1] S. D. Blostein and T. S. Huang, "Estimating 3-D motion from range data," in *Proc. 1st Conf. Artificial Intelligence Applications*, Denver, CO, Dec. 1984, pp. 246-250.

[2] D. Cyganski and J. A. Orr, "Applications of tensor theory to object recognition and orientation determination," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7, pp. 663-673, Nov. 1985.

[3] T. S. Huang, S. D. Blostein, and E. A. Margerum, "Least-squares estimation of motion parameters from 3-D point correspondences," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Miami Beach, FL, June 24-26, 1986.

[4] O. D. Faugeras and M. Hebert, "A 3-D recognition and positioning algorithm using geometrical matching between primitive surfaces," in *Proc. Int. Joint Conf. Artificial Intelligence*, Karlshrue, West Germany, Aug. 1983, pp. 996-1002.

[5] M. Fischler and Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, June 1981.