CABIOS

Vol.6. no.3. 1990
Pages 237 – 245

Improved sensitivity of biological sequence database searches

Douglas L.Brutlag, Jean-Pierre Dautricourt[1], Sunil Maulik[1] and John Relph[1]

Abstract

We have increased the sensitivity of DNA and protein sequence database searches by allowing similar but non-identical amino acids or nucleotides to match. In addition, one can match k-*tuples or words instead of matching individual residues in order to speed the search. A matching matrix specifies which* k-*tuples match each other. The matching matrix can be calculated from a similarity matrix of amino acids and a threshold of similarity required for matching. This permits amino acid similarity matrices or replacement matrices (PAM matrices) to be used in the first step of a sequence comparison rather than in a secondary scoring phase. The concept of matching non-identical* k-*tuples also increases the power of DNA database searches. For example, a matrix that specifies that any 3-tuple in a DNA sequence can match any other 3-tuple encoding the same amino acid permits a DNA database search using a DNA query sequence for regions that would encode a similar amino acid sequence.*

Introduction

Searching biological databases for similar sequences is like looking back in evolutionary time. The greater the time since divergence from a common ancestor, the less similar two homologous sequences will be. The more sensitive the sequence comparison method, the further back in time one can see. Extremely sensitive comparison methods may even detect similarity between sequences arising from convergent evolution rather than from homologs diverging from a common ancestor.

The most sensitive methods for detecting distant relationships in protein sequences have used amino acid replacement matrices or similarity matrices (Dayhoff *et al.*, 1979; Feng *et al.*, 1985). Rather than asserting each pair of amino acids to be matching or not, these matrices allow one to assign a degree of similarity to each pair of amino acids. Several kinds of amino acid matching matrices have been developed, including those that measure the chemical similarity of two amino acids (McLachlan, 1972) and those that measure the similarity of their codons (Fitch, 1966; Fitch and Margoliash, 1967). However, the most sensitive matrix for detecting distant evolutionary relationships, developed by Dayhoff and co-workers (Schwartz and Dayhoff, 1979), measures the frequency with which any amino acid replaces another amino acid in homologous sequences. These matrices of acceptable point mutations (PAM matrices) permit the detection of extremely distant evolutionary relationships.

While PAM matrices are excellent tools for detecting distant evolutionary relationships, the primary difficulty using them is that calculations of similarity are computer intensive. In order to search an entire database, similarity matrices have been abandoned in favor of exact matching of amino acids or groups of amino acids (Dumas and Ninio, 1982; Wilbur and Lipman, 1983). Groups of amino acids (or bases for DNA searches) called words or *k*-tuples can be used to index short identities between a query sequence and a database sequence. Although using *k*-tuples increases the speed of database searches tremendously, there is a considerable loss in sensitivity. To help overcome this loss in sensitivity, sequences initially determined to be similar based on identical *k*-tuples have been rescored for similarity in a secondary evaluation using a PAM matrix (Pearson, 1986; Pearson and Lipman, 1988). However, this use of an amino acid matching matrix still does not detect sequence similarities as remote as mammalian hemoglobins from plant leghemoglobins (Gribskov *et al.*, 1988).

In the method described here, we use an amino acid similarity matrix in the initial step of sequence comparison. In order to retain the speed afforded by exact matching of amino acids or *k*-tuples, we have applied a threshold to the similarity or replacement matrices. All pairs of amino acids with a degree of similarity above an established threshold are considered matching, whether they are identical or not. The threshold can be applied to matrices that measure chemical similarity of amino acids or to matrices that measure the frequency of amino acid replacement and convert these matrices into matching matrices. This method can detect extremely remote sequence similarities, such as the relationship between mammalian hemoglobins and plant leghemoglobins. It can also detect regions of local homology, such as the helix–turn–helix motif common to POU and homeo-box domains of rat pituitary transcription factor (PIT-1) and *Drosophila* homeotic proteins (Ingraham *et al.*, 1988).

The concept of matching similar but non-identical *k*-tuples also has utility in DNA sequence comparisons. For example, by allowing all 3-tuples that encode the same amino acid to match each other, one could use a DNA query to search a DNA

Department of Biochemistry, Beckman Center, Stanford University School of Medicine, Stanford, CA, 94305 and [1]IntelliGenetics Inc., 700 East El Camino Real, Mountain View, CA 94040, USA

C	Cys	93																		
S	Ser	8	17																	
T	Thr	4	15	18																
P	Pro	3	13	10	35															
A	Ala	5	17	17	16	23														
G	Gly	4	17	12	11	20	48													
N	Asn	2	11	9	7	11	12	11												
D	Asp	1	10	8	6	12	13	12	20											
E	Glu	1	10	8	7	12	12	10	18	21										
Q	Gln	1	8	6	8	9	8	7	10	13	16									
H	His	2	6	5	6	7	5	8	7	7	11	26								
R	Arg	2	8	6	7	7	5	6	5	5	8	8	29							
K	Lys	2	12	12	8	11	9	13	11	11	12	9	23	42						
M	Met	1	4	5	3	6	4	2	2	2	3	1	3	9	11					
I	Ile	3	6	8	4	9	5	3	3	4	3	2	3	6	7	18				
L	Leu	1	6	8	6	9	5	5	4	5	6	5	5	7	20	18	60			
V	Val	5	8	11	7	13	9	5	5	6	5	4	4	7	10	20	20	30		
F	Phe	1	4	3	2	4	3	2	1	1	1	3	2	2	4	8	16	6	58	
Y	Tyr	5	3	2	1	3	2	3	1	1	1	4	1	2	1	3	7	3	31	55
W	Trp	1	3	2	1	1	1	1	1	1	1	1	6	3	1	1	4	1	4	3	100
		C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
		Cys	Ser	Thr	Pro	Ala	Gly	Asn	Asp	Glu	Gln	His	Arg	Lys	Met	Ile	Leu	Val	Phe	Tyr	Trp

Fig. 1. A PAM 250 amino acid replacement matrix. The numbers in this matrix are a normalized form of the frequency of one amino acid replacing another amino acid after 250 random substitutions in a sequence 100 amino acids long. The table is based on Dayhoff's original amino acid substitution data (Schwartz and Dayhoff, 1979). A matrix giving the probability of any amino acid substituting for another after a single mutation is first calculated (a PAM 1 matrix, which is in effect the model of the evolutionary events). This matrix is then used as a state transition matrix to calculate the probability of amino acid replacement after 250 random substitutions using Markov methods. The final matrix is calculated by taking the logarithms of these probabilities and then normalizing these logarithms in the range 1−100. The normalization is done primarily to simplify the selection of an appropriate threshold. The program uses the logarithms of the probabilities to determine the optimized score as suggested by Schwartz and Dayhoff (1979).

database for other DNA sequences that would encode a similar protein regardless of the codons used in each sequence. By allowing all alternating pyrimidine−purine dinucleotides to match each other together with a query sequence of alternating purine−pyrimidines, one can find sequences containing runs of alternating purines and pyrimidines.

System and methods

The searching procedure implemented in the FASTDB program begins by evaluating the similarities of *k*-tuples based on a matrix of amino acid similarity or replacement values and a threshold. The matrices are numbers assigned to pairs of residues (amino acids or bases or *k*-tuples of amino acids or bases) which measure the degree of similarity (a number between 1 and 100 interpreted as a percentage) or the frequency of replacement of one amino acid for another in evolution (also normalized to numbers between 1 and 100, Figure 1). A similarity matrix is converted to a matching matrix by comparing the values with the threshold. Residues are considered as matching if the value in the similarity matrix is equal to or greater than the threshold and not matching otherwise (Figure 2). The similarity of two *k*-tuples is calculated by multiplying the similarity values of each constituent amino acid pair from each *k*-tuple. The threshold of similarity is raised to the power *k*, and this number is compared to the similarity of each pair of *k*-tuples to determine if they match or not. Alternatively, one may specify directly each element of a matching matrix indicating whether any two *k*-tuples match or not. If one chooses to use a PAM matrix, the program calculates the matrix based on the observed frequencies of amino acid substitution to any level of sequence divergence (Schwartz and Dayhoff, 1979). A level of 100 random substitutions per 100 amino acids (PAM 100) is usually used to detect closely related protein sequences, while higher levels of substitution (250 random substitutions in 100 amino acids) are used to detect more distantly related proteins. The frequencies of amino acid replacements are calculated using Dayhoff's original substitution data, but with properly normalized Markov matrices and to a high degree of precision in order to avoid problems of asymmetry noted by others (Wilbur, 1985). Other measures of amino acid similarity are also encoded in FASTDB [a unitary matrix, a genetic code matrix (Feng *et al.*, 1985), and McLachlan's structure-genetic code matrix (McLachlan, 1972)]. For DNA queries, a unitary matrix and a genetic code 3-tuple matrix are provided.

Once the matching *k*-tuple matrix is determined, then the

C	Cys	1																		
S	Ser	0	1																	
T	Thr	0	1	1																
P	Pro	0	1	0	1															
A	Ala	0	1	1	1	1														
G	Gly	0	1	1	1	1	1													
N	Asn	0	1	0	0	1	1	1												
D	Asp	0	0	0	0	1	1	1	1											
E	Glu	0	0	0	0	1	1	0	1	1										
Q	Gln	0	0	0	0	0	0	0	1	1	1									
H	His	0	0	0	0	0	0	0	0	0	1	1								
R	Arg	0	0	1	0	0	0	0	0	0	0	0	1							
K	Lys	0	1	1	0	1	0	1	1	1	1	0	1	1						
M	Met	0	0	0	0	0	0	0	0	0	0	0	0	0	1					
I	Ile	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1				
L	Leu	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1			
V	Val	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	1	1		
F	Phe	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	
Y	Tyr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
W	Trp	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
		C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
		Cys	Ser	Thr	Pro	Ala	Gly	Asn	Asp	Glu	Gln	His	Arg	Lys	Met	Ile	Leu	Val	Phe	Tyr	Trp

Fig. 2. A PAM 250 1-tuple matching matrix for a threshold of 11. This matrix shows the pairs of amino acids that are considered as matching, based on the PAM 250 matrix (Figure 1) and using a matching threshold of 11. The order of amino acids has been chosen to show the groupings in which most will match each other. These groups of amino acids that often substitute for each other in evolution are also chemically similar to each other.

initial comparison of the query sequence to the database is performed in a manner analogous to that used by the FastP, FastN and FastA programs (Lipman and Pearson, 1985; Pearson, 1986; Pearson and Lipman, 1988). In FASTDB, however, one has the ability to specify the mismatch penalty in addition to the joining penalty, the size of the *k*-tuple, the degree of sequence divergence and the threshold. FASTDB considers all *k*-tuples that match a given database sequence *k*-tuple in building the matching diagonals. The top 10 matching diagonals for each database sequence are scored based on the formula:

$$\text{Score} = F \times K \times M - N \times P$$

where:

F = K-tuple matching factor (number of matching k-tuples in unitary matrix/number of matching k-tuples in matching matrix)
K = k-tuple length
M = number of k-tuple matches in region
N = number of k-tuple mismatchers in region
P= mismatch penalty

After the 10 best diagonals are found by this rapid scoring scheme, these diagonals are rescored by summing the logarithms of the similarities for each *k*-tuple match. These diagonals are joined to other diagonals using insertion/deletion gaps (indels) if the two diagonals are non-overlapping and if the scores of both diagonals exceed the joining penalty. The score for the joined diagonals is the sum of scores of the joined diagonals minus a joining penalty for each join that is performed (Figure 3). This method guarantees that each pair of joined diagonals has a total score greater than the individual diagonals that form it.

An initial score is saved for each sequence in the database if it is above a specified cutoff. When the number of scores saved exceeds 5000, the cutoff is raised dynamically and the lower 1000 scores are discarded. At the end of the database search, the sequences with the top 4000−5000 initial scores are retained. A distribution of the initial scores is plotted on a logarithmic scale and the mean and standard deviation of the population are calculated (Figure 4). The standard deviation is recalculated, excluding values that are > 4 SDs above the mean. Using a logarithmic plot increases the sensitivity of detecting clusters of similar sequences located at the ends of the distribution of the similarity scores. At this point, one can choose to print the names of any number of the top sequences sorted by their initial scores.

To increase further the sensitivity of finding the best matches, FASTDB optionally allows one to re-evaluate the most similar alignments based on an optimized score. An optimized score is calculated for each of the top 4000−5000 alignments according to the method of Smith and Waterman (1981). The user

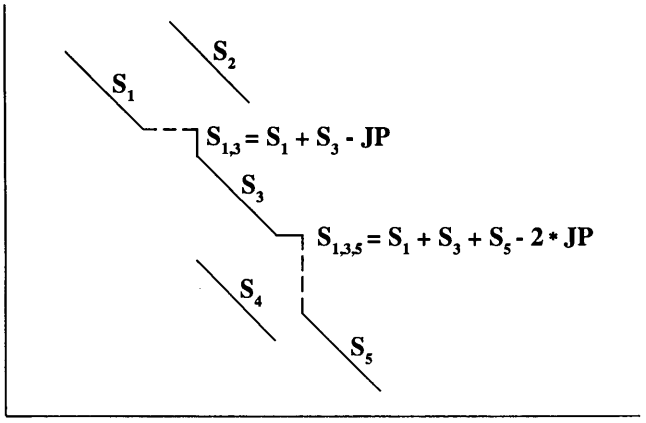

Fig. 3. Joining of non-overlapping regions in the initial scan. This plot shows the diagonals containing the best matching regions between two sequences. The best non-overlapping initial regions (in this example, regions 1, 3 and 5) are joined together if the combined score of those two regions minus a joining penalty is greater than the either of the regions individual scores. S_n represents the scores of individual regions with the largest number of matching *k*-tuples and JP is the joining penalty.

is allowed to specify the gap penalty, gas-size penalty and window size in this calculation. The sequences are then ranked by these optimized scores. A new mean and standard deviation are calculated, excluding the 2% most extreme values. A distribution of the optimized scores contributing to the calculations of the mean and standard deviation is then plotted.

Results

Detecting distantly related protein sequences

The increased sensitivity of FASTDB is demonstrated by its ability to detect sequence similarity between remotely related proteins. Mammalian hemoglobins and plant leghemoglobins represent two proteins at opposite ends of the taxonomic spectrum that have a similar function: to bind oxygen at very low oxygen concentration (Appleby, 1984; Long, 1989). Using the human hemoglobin α-chain as a query to search the entire Protein Identification Resource (PIR) database, FASTDB found that the first 310 sequences sorted by their optimized scores were hemoglobins (157 α-chains followed by 136 β-chains and then 17 other hemoglobin chains). Next, the bulk of the myoglobin sequences were found between 294 and 373 in the sorted list, overlapping the more remote hemoglobin chains. Immediately following the hemoglobins and myoglobins, a number of plant leghemoglobins and more distantly related myoglobins were found. Table I shows the positions of these leghemoglobins and myoglobins in the sequence list, sorted by optimized scores. All but three of the leghemoglobins are > 4 SDs from the mean. Ten of the 13 leghemoglobins present in the PIR database were found between positions 376 and 463 in the list. The three leghemoglobin sequences that were not found were fragments of the complete protein and hence had low similarity scores. The FastA program, using *k*-tuple = 1, a PAM scoring factor (PAMFACT on) with the same query sequence and database, found only one leghemoglobin sequence in its top 500 sequences.

Detection of the remotely related leghemoglobins requires both the matching matrix and the optimization of the top scores. Table II shows a systematic study of the number of leghemoglobins detected in the top 500 scores as a function of the degree of sequence divergence (number of acceptable point mutations or PAMs) and the threshold of replacement. Without optimization of the scores, only three leghemoglobins were detected and their highest level of significance, 3.14 SDs from the mean, was much below the levels of significance obtained with optimization of the alignments which were up to 6.5 SDs above the mean (Table I). FastA, which uses a unitary matrix for the initial scores and a PAM matrix for optimization, only detects a single plant leghemoglobin sequence in the top 500 sequences and its score was not significant.

Detecting local similarities

The similarity of the human hemoglobin α-chain to related heme-binding proteins represents global similarities that span the entire length of the protein. FASTDB is also capable of finding short local similarities between a query sequence and proteins in the database. For example, the rat tissue-specific transcription factor PIT-1, responsible for expression of prolactin and growth hormone in the rat pituitary gland, shares several homeo- and POU DNA binding domains with other transcription regulatory proteins (Ingraham *et al.*, 1988). When this protein is used as a query sequence with a more stringent PAM matrix, 18 of the first 24 proteins were proteins known to possess homeo- or POU DNA binding domains (Table III). All of these scores are > 5.0 SDs from the mean similarity score. From these results, the homeo-box domains appear to be more highly conserved than hemoglobins (see below). Four of the other proteins in the top 24 sequences were also known to be DNA binding proteins, including one prokaryotic enzyme, which may share a helix−turn−helix motif with the POU and homeo-box domain. A similar search of the same database using FastA detected only 11 of the more closely related homeotic proteins and no prokaryotic DNA binding proteins in the top 24 sequences.

If one uses the 61 amino acid homeo-box sequences extracted from the 291 amino acid PIT-1 protein as the query, then both FASTDB and FastA find all homeotic sequences present in the database (64 in PIR 19 and 84 in Swiss-Prot 21). These results have two important implications. First, one can make a database search more selective by limiting the focus to a single functional domain in a query sequence, even with a search protocol designed to find the best local homology. Second, the increased sensitivity offered by FASTDB can increase the score of remotely related sequences, even if the similarity with the query is restricted to a small region.

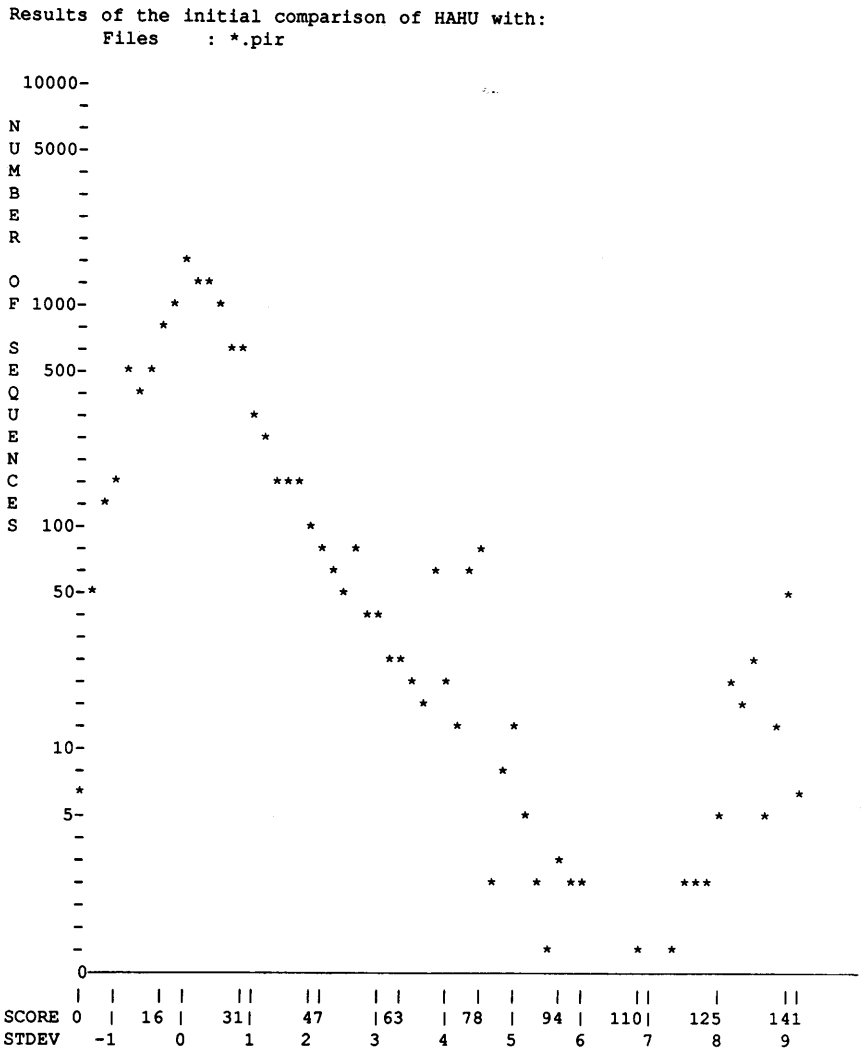

Fig. 4. The distribution of initial scores comparing the human α-globin chain versus the PIR Release 19 database. The number of sequences at each score interval is plotted on a logarithmic scale that exaggerates deviations from a normal distribution at the extremes of the curve. Standard deviations from the mean are also plotted on the abscissa. This plot comes from the experiment described in Table I and the rest of the parameters are listed there.

Adjusting the sensitivity

FASTDB allows a great degree of flexibility in detecting related proteins. One can alter both the hypothetical degree of amino acid replacement using any PAM matrix, from 1 replacement per 100 amino acids to over 250 replacements per 100 amino acids, and an arbitrary threshold to declare two residues or *k*-tuples as matching (Table II). In any particular case, it is often best to run a number of searches at varying degrees of amino acid replacement, as originally suggested by Dayhoff (Schwartz and Dayhoff, 1979). For instance, the maximum similarity between the PIT-1 protein and the engrailed homeotic protein (PIR sequence WJFFEN) as measured by the standard deviation of their similarity score from the mean is obtained with a PAM-150 matrix. The statistical significance of the similarity score of these two sequences decreases if either a PAM 100 or PAM 200 matrix is used. On the other hand, the maximum similarity score between the human hemoglobin α-chain and the leghemoglobin c1 from soybean (PIR sequence GPSYC1) is obtained with a PAM 200 matrix. This evidence suggests that the pair of homeotic proteins are more highly conserved or more closely related than the pair of hemoglobins.

Decreasing the threshold of similarity can also increase the sensitivity of the search. For example, if the threshold is decreased from 20 to 11 in the search for proteins homologous to the human hemoglobin α-chain, the number of plant leghemogobins found in the top 500 sequences increases from 4 to 10 (Table II). However, raising the threshold has a beneficial effect on lowering the background from other sequences. For example, using a PAM 250 matrix and a threshold of 11, lysine matches eight other amino acids, glycine matches seven other amino acids and serine matches six other

Table I. Leghemoglobin sequence similarity to human α-chain hemoglobin

No.	Name	Description	Length	Init.	Opt.	Sig.
376	MYRKJ	myoglobin—Port Jackson shark	148	69	103	6.50
377	GPFBA	leghemoglobin a—kidney bean	145	57	103	6.50
378	MYTUY	myoglobin—yellowfin tuna	146	61	103	6.50
379	MYCA	myoglobin—carp	146	60	103	6.50
386	A25331	myoglobin—*Dolabella auricula*	146	51	102	5.78
388	GPSYC3	leghemoglobin c3—soybean	144	62	102	5.78
390	GPSYC1	leghemoglobin c1—soybean	143	51	101	5.06
391	GPYL2	leghemoglobin II—yellow lupi	153	66	101	5.06
393	GPSYS	leghemoglobin a—soybean	143	50	101	5.06
396	GPSYC2	leghemoglobin c2—soybean	143	65	101	5.06
398	A20801	hypothetical leghemoglobin—S	151	41	100	4.34
399	S00560	non-legume hemoglobin I—casua	151	30	100	4.34
419	GPPMI	leghemoglobin I—garden pea	147	37	99	3.61
420	A29282	leghemoglobin III—alfalfa	146	43	99	3.61
444	HBSHBC	hemoglobin beta C(NA) chain	141	65	98	2.89
463	GPVF	leghemoglobin I—broad bean	143	40	98	2.89
471	GGZLB	bacterial hemoglobin—vitreos	146	34	98	2.89

The human α-chain hemoglobin was used as a query sequence versus the PIR Release 19 database, with PAM 250 and a threshold of 11. The following additional parameters were used in the search: *k*-tuple = 1; mismatch penalty = 1; joining penalty = 20; gap penalty = 5; gap-size penalty = 0.05; and window size = 32. The list of sequences sorted by their optimized scores contained 293 hemoglobins followed by 83 myoglobins. Immediately after those follow the plant leghemoglobins and the more distantly related myoglobins and hemoglobins. The table gives the position of each sequence in the list of sequences sorted by their optimized scores, the PIR name of the sequence, the first 30 characters of the description line in the PIR database, the length of the protein, its initial score and its optimized similarity score compared to human hemoglobin α-chain and the standard deviation of the optimized score from the mean of the optimized scores.

Table II. Detection of distant protein similarities as a function of PAM matrix and threshold

No. PAMs	Threshold	No. leghemoglobins
Unitary matrix		2
100	10	3
100	15	2
100	20	2
150	11	3
200	12	7
200	15	5
250	11	10
250	15	3
250	20	4

The number of plant leghemoglobins among the top 500 similarity scores from a search of PIR (Release 19) using human hemoglobin α-chain as a query. All searches were performed as described in Table I except for the number of acceptable point mutations per 100 amino acids (No. PAMs) and the threshold given above. Columns 1 and 2 give the number of acceptable point mutations used in calculating the PAM matrix and the threshold applied to this table to determine which amino acids will match each other (see System and methods).

amino acids. This high degree of matching causes proteins rich in these three amino acids to be scored more highly than other proteins. Histone H1 is one such protein rich in lysine, glycine and serine, and a number of the histone H1 proteins in the database have a relatively high similarity score, using these stringencies, when compared with nearly any query sequence. As the threshold is increased from 11 to 15 to 20, the number of histone H1 proteins in the top 500 sequences in the list of sequences sorted by their optimized scores falls from four to two to zero. With a PAM replacement matrix of 150 replacements per 100 amino acids and thresholds between 11 and 16, no histone H1 proteins are found in the top 500 sequences. This suggests that a delicate balance between sensitivity and noise must be determined when searching for very distantly related sequences.

Speed of the search

The speed of search is a function of the number of matching *k*-tuples and hence is a function of the PAM matrix and the threshold. If one uses a unitary matrix, allowing only identical amino acids to match, FASTDB takes about twice as long to search a protein database as does FastA (Table IV). This is due, in part, to the more complicated structure of the database FASTDB searches compared with FastA. A future version of FASTDB will use a simpler database, resulting in a 40% increase in search speed. When one increases the sensitivity by using a PAM matrix with a low threshold, the time required to search the database increases in proportion to the number of *k*-tuples that match under the defined conditions (Table IV and Figure 5). This increase in search time is to be expected, since more sensitive searches require more matching pairs, and the time of a search should be proportional to the number of matching pairs.

Searches of DNA databases

The concept of matching non-identical *k*-tuples is very useful for searching DNA databases as well as protein databases. One example is using a protein sequence as a query to search a DNA database. FASTDB can carry out this kind of search in two ways. First, it can translate the entire database in three or six frames and can report and sort scores for protein sequences in all frames simultaneously. Alternatively, FASTDB can use a reverse translated form of a protein query sequence. This procedure of reverse translation results in a DNA query sequence that is necessarily ambiguous because of the redundancy of the genetic code. By properly matching all of the ambiguous base codes in the reverse translated DNA sequence, FASTDB allows all possible codons of a protein query to match appropriate codons in the database. The handling of ambiguous base sequences has been implemented in a fully symmetric fashion, so that ambiguities in the database are also properly matched. The only exception to this is that the fully ambiguous character N is not allowed to match any base. If it were, many runs of Ns that exist in both the GenBank and EMBL databases would match every query.

When the reverse translated sequence of the human

Table III. Similarity between the tissue-specific transcription factor PIT1 from rat pituitary glands and known homeotic proteins

No.	Name	Description	Length	Init.	Opt.	Sig.
1	A26066	segmentation protein eve—fru	376	49	83	7.61
2	A26636	segmentation protein eve—fru	376	49	83	7.61
3	A27662	mec-3 homeotic protein—caeno	313	47	144	7.20
4	B25682	engrailed homeotic protein—F	584	47	155	7.20
5	WJFFEN	engrailed homeotic protein—F	552	47	156	7.20
6	S00987	Hox 6.1 homeotic protein—mou	153	45	75	6.79
7	A25872	acid phosphatase synthesis reg	559	45	86	6.79
8	S00835	bicoid homeotic protein—frui	494	44	92	6.58
9	A29585	Hox 2.3 homeotic protein precu	87	43	48	6.38
10	A26846	Hox-2.3 homeotic protein—mou	217	43	112	6.38
11	WJXLMM	MM3 homeotic protein—African	88	43	49	6.38
12	WJMSM6	m6 homeotic protein—mouse (f	119	42	55	6.17
13	A25399	Antennapedia homeotic protein	378	41	145	5.97
14	A25400	Antennapedia homeotic protein	378	41	145	5.97
15	A23450	Antennapedia homeotic protein	378	41	145	5.97
16	HBCQ	hemoglobin beta chain—spectra	146	40	76	5.76
17	A26062	segmentation protein prd—fru	613	40	150	5.76
18	C29585	Hox 4 homeotic protein precurs	120	40	51	5.76
19	TVVPTB	large T antigen—polyomavirus	695	40	145	5.76
20	NDBSR1	type II restriction enzyme, Bs	576	39	87	5.56
21	B26332	BSH9 homeotic protein—fruit	80	39	49	5.56
22	TVVPT4	large T antigen—*Rhesus maca*	708	39	147	5.56
23	A05280	myosin heavy chain, skeletal m	170	38	73	5.35
24	A26332	BSH4 homeotic protein—fruit	80	38	48	5.35

The PIT-1 protein sequence was used as a query in a search of the PIR Release 19 database using a PAM 150 amino acid replacement matrix and a threshold of 20. All other parameters were as mentioned in Table I. The sequences are ranked by sorting their initial scores.

Table IV. Initial search times for comparing the human hemoglobin α-chains with the PIR database

Program and matrix employed	Threshold	No. of metching 1-tuples	CPU search time (min:s)
FASTDB			
PAM 100	20	22	8:57
PAM 100	15	24	9:46
PAM 100	10	36	18:40
PAM 250	20	19	10:17
PAM 250	15	32	17:46
PAM 250	11	56	31:09
Unitary matrix		20	10:41
FastA (PAMFACT ON)		20	4:27

The PIR sequence HAHU (146 residues) was used as a query to search the PIR Release 19 database (2, 802, 055 residues) using parameters given in Table I, except for the PAM matrix and threshold shown here. The number of matching 1-tuples in column 3 is the total number of matching amino acids (both identical and non-identical) that one obtains by applying the matching threshold in column 2 to the PAM matrix described in column 1 (see System and methods). The time for an identical search using FastA is also shown. All searches were performed on a Sun 3/280 computer.

hemoglobin α-chain is used as a query to search the EMBL database (Release 18, 22,994 sequences and 27,249,782 residues), the top 105 sequences were all globin sequences. Most of these sequences were either cDNA or pseudogenes, because the genomic versions of the globins are divided by two intervening sequences and hence have lower optimized scores. Of the 127 globin cDNA and pseudogenes in the EMBL database, 118 were found among the top 133 sequences in the list of optimized scores. The nine remaining globin DNA sequences were located among the top 200 scores.

The implementation of ambiguous matching in FASTDB does not sacrifice speed. The previous search was carried out using a *k*-tuple size of 4 and took 33 min 34 s to search the entire EMBL database. This compares favorably with 29 min for an unambiguous query sequence of similar length and 15 min for FastA which does not match ambiguous bases.

As mentioned, FASTDB also allows one to search a DNA database using a protein sequence by directly translating the database in either three or all six frames during the search itself. This method gives comparable sensitivity to the method using a reverse translation with ambiguous bases; it also suffers from the same limitation of not scoring genomic sequences with intervening sequences as highly as contiguous cDNA and messenger RNA sequences. The direct translation method also suffers from a 6-fold increase in the database search time that is required for each of the frames.

One final method of searching DNA databases for protein coding regions, using a DNA query sequence, allows non-identical *k*-tuples to match. By using a matching matrix that allows two 3-tuples to match if they encode the same amino acid, FASTDB locates all similar coding regions between the DNA query and the DNA database, independent of the actual codons used in either. In a search using the chimpanzee cDNA sequence for its α-globin protein as a query against the EMBL database, the first 145 sequences found included exclusively globin coding regions from a wide variety of organisms (data not shown). This search is 8 times slower than the search using ambiguous codons, primarily because of the use of 3-tuples rather than 4-tuples for performing the database search.

Discussion

Two properties of FASTDB are responsible for its increased sensitivity in database searches. First, the ability of FASTDB to allow non-identical residues or *k*-tuples to match in the initial search detects similar regions that are ignored by algorithms that only allow exact matching. Second, FASTDB does a secondary scoring of the top 4000−5000 sequences using a PAM matrix and the method of Smith and Waterman (1981). This secondary scoring and subsequent sorting raise the similarity scores and statistical significance of very distantly related sequences much higher than otherwise possible. It is only worth performing this rescoring and secondary sorting if one can be assured that the distantly related sequences are, in fact, contained in the saved sequences from the initial pass. It is the matching of non-identical residues that ensures that remotely related proteins are detected in the first pass.

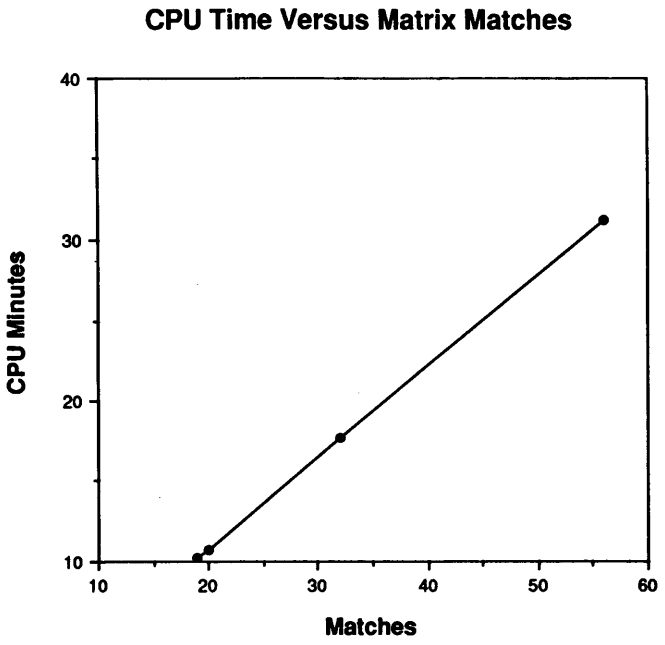

Fig. 5. Search time as a function of number of matching amino acids in matching matrix. This plot shows the data from Table IV for FASTDB search time for PAM 250 data as a function of the number of matching amino pairs in the matching matrix. The point with exactly 20 pairs of matching amino acids represents the data for FASTDB with a unitary matrix.

FASTDB can utilize many different matching matrices customized for detecting a variety of types of similarities. Several matching matrices are coded within the FASTDB program. For searching DNA databases, FASTDB provides a genetic code matching matrix of 3-tuples that calls any two codons encoding the same amino acid matching and all other pairs non-matching. Similarly, one can manually enter matrices that allow specific dinucleotide matches, tetranucleotide matches, etc. For manually entered matrices there is no basis for residue scoring and no optimization is performed. Thus, FASTDB can perform similarity searches using stringencies from any pattern of *k*-tuple exact matching to any pattern of inexact matching.

Matching matrices can also be calculated indirectly from tables of amino acid similarities or substitution frequencies combined with a threshold to determine matching *k*-tuples. In this paper we have emphasized the use of amino acid substitution matrices (also known as acceptable point mutation matrices or PAM matrices). FASTDB calculates PAM matrices to any degree of amino acid substitution from the original Dayhoff data on frequencies of amino acid replacements. By starting with the original substitution data and using Markov methods to calculate the PAM matrix, FASTDB eliminates both rounding errors and errors in improper normalization of the transition matrices described by Wilbur (1985). This permits the maximum flexibility of the sensitivity of the search to range from one acceptable point mutation per 100 amino acids (nearly a unitary matrix) to over 700 mutations per 100 amino acids (close to the steady-state amino acid composition for a highly mutated protein sequence).

As noted by Dayhoff (Schwartz and Dayhoff, 1979), matrices of acceptable point mutations naturally divide the amino acids into groups of interchangeable residues which correspond quite well with classes of amino acids based on chemical properties. This grouping can be seen in Figures 1 and 2. However, when a user applies a specific threshold to such matrices, a degree of subtlety in matching is achieved that cannot be obtained by merely performing a simple transformation of the amino acid alphabet. For example, in the matching matrix derived from a PAM 250 replacement matrix filtered with a threshold of 11 (Figure 2), all of the amino acid pairs in the group (serine, threonine, proline, glycine and alanine) match each other except for threonine versus proline. It is this ability to specify precisely which pairs of residues match and which do not that gives FASTDB its power of discrimination.

With this increased sensitivity come important concerns about arbitrary sequences which match in amino acid composition but not in amino acid sequence. As mentioned, PAM 250 matrices with a threshold of 11 allow lysine, glycine and alanine to match a large number of other amino acids. This matching causes the score of histone H1 proteins, rich in these three amino acids, to rise to levels of significance with nearly any query sequence. A higher threshold would be the natural solution to this artefact. However, raising the threshold can eliminate some biologically significant matches and so should be done with caution. This problem of sensitivity versus background is not unique to the implementation of matching matrices in FASTDB. Even with unitary matrices, the sequences near the top of the distribution of similarity scores have an amino acid composition more closely related to the query than to sequences with lower scores. It is often difficult on a statistical basis to separate sequences showing amino acid composition biases from true sequence homology based on a common ancestral sequence.

FASTDB provides many tools with which to judge the statistical significance of the similarity scores. First, distributions of both the initial and the optimized scores are plotted (Figure 4). The abscissa of the plot shows both the absolute range of scores as well as the standard deviation of each range from the mean. If the database is of sufficient size, then the standard deviation of a sequence's score from the mean, coupled with the size of the database, allows one to calculate an expectation frequency that gives a good estimate of the significance of a protein's score (Collins *et al.*, 1988). If there are an insufficient number of sequences in the database being searched, then FASTDB can randomly permute the query sequences up to 30 times and compare each of the permuted sequences against the same database. The mean and standard deviations are then calculated using these random scores. In addition, FASTDB allows one to permute either the individual residues in the query or to permute *k*-tuples of any size. These calculations of

significance can help overcome the amino acid composition bias mentioned above.
tions of significance can help overcome the amino acid composition bias mentioned above.

In addition to the PAM matrices, several other amino acid similarity matrices are available in FASTDB, including a genetic code matrix and McLachlan's structure-genetic code matrix (McLachlan, 1972). With the demonstrated increased sensitivity of FASTDB and the use of matrices based on structural similarities, one might expect to be able to search for proteins that are similar in structure to other proteins (for instance β-barrels or coiled-coils), but share neither functional nor evolutionary relationships. One might expect that in order to detect structural aspects of protein sequences, one would have to tailor the amino acid replacement matrices to the various structural aspects of the query sequence. For example, if one knew that a particular region of the query was an α-helix and another region was a β-sheet, then it would be most appropriate to have different substitution matrices to represent the kinds of replacements seen in these different structures. This concept is similar to the profile idea described by Gribskov *et al.* (1987) in which a different matrix of similarity was assigned to every position in the query. The rapid FASTDB implementation could be readily converted to use different matrices for different regions and even for different residues in a query sequence. Experiments to see if FASTDB can be used to detect similarities as remote as structural homologies are currently under way.

Availability

FASTDB is currently available on the GenBank Online Service and directly from IntelliGenetics, Inc.

Acknowledgements

We would like to acknowledge the expert help of Nancy Bigham, Cindy Brehmer and Kenna Mawk in the design and testing of FASTDB. We would like to thank Cindy Cohen for help in preparing the manuscript and especially for the tables and figures. We would also like to thank Drs Harry Mangalam and Michael Rosenberg for encouraging us to use their PIT-1 sequence for finding other homeotic proteins. This work was supported entirely by IntelliGenetics Inc.

References

Appleby,C.A. (1984) Leghemoglobin and *Rhizobium* respiration. *Annu. Rev. Plant. Physiol.*, **35**, 443–478.

Collins,J.F., Coulson,A.F. and Lyall,A. (1988) The significance of protein sequence similarities. *Comput. Applic. Biosci.*, **4**, 67–71.

Dayhoff,M.O., Schwartz,R.M. and Orcutt,B.C. (1979) A model of evolutionary change in proteins. *Atlas of Protein Structure*, **5**, Suppl. 3, 345–352.

Dumas,J.P. and Ninio,J. (1982) Efficient algorithms for folding and comparing nucleic acid sequences. *Nucleic Acids Res.*, **10**, 197–206.

Feng,D.F., Johnson,M.S. and Doolittle,R.F. (1985) Aligning amino acid sequences: comparison of commonly used methods. *J. Mol. Evol.*, **21**, 112–125.

Fitch,W.M. (1966) An improved method of testing for evolutionary homology. *J. Mol. Biol.*, **16**, 9–16.

Fitch,W.M. and Margoliash,E. (1967) Construction of phylogenetic trees. *Science*, **155**, 279–284.

Gribskov,M., McLachlan,A.D. and Eisenberg,D. (1987) Profile analysis: detection of distantly related proteins. *Proc. Natl. Acad. Sci. USA*, **84**, 4355–4358.

Gribskov,M., Homyak,M., Edenfield,J. and Eisenberg,D. (1988) Profile scanning for three-dimensional structural patterns in protein sequences. *Comput. Applic. Biosci.*, **4**, 61–66.

Ingraham,H.A., Chen,R., Mangalam,H.J., Elsholtz,H.P., Flynn,S.E., Lin,C.R., Simmons,D.M., Swanson,L. and Rosenfeld,M.G. (1988) A tissue-specific transcription factor containing a homeodomain specifies a pituitary phenotype. *Cell*, **55**, 519–529.

Lipman,D.J. and Pearson,W.R. (1985) Rapid and sensitive protein similarity searches. *Science*, **227**, 1435–1441.

Long,S.R. (1989) Rhizobium-legume nodulation: life together in the underground. *Cell*, **56**, 203–214.

McLachlan,A.D. (1972) Repeating sequences and gene duplication in proteins. *J. Mol. Biol.*, **64**, 417–437.

Pearson,W.J. (1986) Sensitivity and selectivity in protein sequence comparison. In *Methods in Protein Sequence Analysis*. Humana Press, Clifton, NJ, pp. 521–434.

Pearson,W.R. and Lipman,D.J. (1988) Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. USA*, **85**, 2444–2448.

Schwartz,R.M. and Dayhoff,M.O. (1979) Matrices for detecting distant relationships. *Atlas of Protein Structure*, **5**, Suppl. 3, 353–358.

Smith,T.F. and Waterman,M. (1981) Identification of common molecular subsequences. *J. Mol. Biol.*, **147**, 195–197.

Wilbur,W.J. (1985) On the PAM matrix model of protein evolution. *Mol. Biol. Evol.* **2**, 434–447.

Wilbur,W.J. and Lipman,D.J. (1983) Rapid similarity searches of nucleic acid and protein data banks. *Proc. Natl. Acad. Sci. USA*, **80**, 726–730.

Received October 10, 1989; accepted May 1, 1990