Flexible sequence similarity searching with the **FASTA3** program package

William R. Pearson

Department of Biochemistry,
University of Virginia,
Charlottesville, VA 22908

August 28, 1998

Phone: 804-924-2818; FAX: 804-924-5069; wrp@virginia.edu

1. INTRODUCTION

Since the publication of the first rapid method for comparing biological sequences 15 years ago (1), DNA and protein sequence comparison have become routine steps in biochemical characterization, from newly cloned proteins to entire genomes. As the DNA and protein sequence databases become more complete, a sequence similarity search is more likely to reveal a database sequence with statistically significant similarity, and thus inferred homology, to a query sequence. Indeed, even in the archaebacterium *M. jannaschii*, more than 40% of the open reading frames could be assigned a function based on significant sequence similarity to a protein of known function (2).

This chapter provides a "hands on" overview of the programs in the FASTA package. Rather than discuss in depth the theory and practice of protein and DNA sequence comparison, I focus on more practical questions, such as: "Which FASTA program should I use?", "What threshold should I use for statistical significance?", "Which databases should I search?", "When should I use FASTA and when should I use BLAST?", and "When should I change the scoring matrix and gap penalties?" For an excellent review of similarity searching with BLAST and FASTA and of local similarity statistics, see ref. 3. For more specific information on how to use the FASTA programs to identify distantly related sequences, see refs. 4 and 5. A detailed explanation of the statistical estimates in the `fasta3` package is provided in ref. 6.

2. SIMILARITY SEARCHING WITH THE FASTA3 PROGRAMS

The FASTA program package has evolved significantly since its introduction ten years ago (7). The original package offered four programs: `fasta`, `tfasta`, `lfasta`, and `rdf` (`rdf` was introduced with the first `fastp` program in 1985; ref. 8). Today, programs are available for rigorous Smith- Waterman searches (`ssearch3`) and for searches with mixed peptide sequences (`fastf3` and `tfastf3`); the programs for translated DNA:protein sequence comparison have been improved substantially with the introduction of `fastx3`, `fasty3`, `tfastx3`, and `tfasty3`, and the program for estimating statistical significance from shuffled-sequence similarity scores (`prss3`) produces accurate statistical estimates. The FASTA3 programs for database searching are summarized in Table 1; the programs for evaluating statistical significance are shown in Table 2.

Table 1: Comparison programs in the FASTA3 package

`fasta3`	Compare a protein sequence to a protein sequence database or a DNA sequence to a DNA sequence database using the FASTA algorithm (4, 7). Search speed and selectivity are controlled with the ktup (wordsize) parameter. For protein comparisons, *ktup= 2* by default; *ktup=1* is more sensitive but slower. For DNA comparisons, *ktup= 6* by default; *ktup=3* or *ktup= 4* provides higher sensitivity; *ktup=1* should be used for oligonucleotides (DNA query lengths <20).
`ssearch3`	Compare a protein sequence to a protein sequence database or a DNA sequence to a DNA sequence database using the Smith-Waterman (22) algorithm. `ssearch3` is about 10-times slower than `FASTA3`, but is more

	sensitive for full-length protein sequence comparison.
`fastx3`/ `fasty3`	Compare a DNA sequence to a protein sequence database, by comparing the translated DNA sequence in three frames and allowing gaps and frameshifts. `fastx3` uses a simpler, faster algorithm for alignments that allows frameshifts only between codons; `fasty3` is slower but produces better alignments with poor quality sequences because frameshifts are allowed within codons.
`tfastx3`/ `tfasty3`	Compare a protein sequence to a DNA sequence database, calculating similarities with frameshifts to the forward and reverse orientations.
`tfasta3`	Compare a protein sequence to a DNA sequence database, calculating similarities (without frameshifts) to the 3 forward and three reverse reading frames. `tfastx3` and `tfasty3` are preferred because they calculate similarity over frameshifts.
`fastf3`	Compare a mixed peptide sequence to a protein sequence database. A mixture of peptides, typically obtained by Edman degradation after cyanogen bromide cleavage without further separation, is compared with protein sequences in a database to identify those sequences that are most likely to produce the peptide mixture.
`tfastf3`	Compare a mixed peptide sequence to a translated DNA sequence database.

Table 2: Statistics programs in the `FASTA3` package

`prss3`	Evaluate the significance of a protein or DNA sequence similarity score by comparing two sequences and calculating optimal similarity scores, and then repeatedly shuffling the second sequence, and calculating optimal similarity scores using the Smith-Waterman algorithm. The characteristic parameters of the extreme value distribution are estimated from the shuffled sequence scores and used to calculate the statistical significance of the unshuffled sequence similarity score.
`sc_to_e`	Calculate the statistical significance of a similarity score from the raw score, the length of the sequence, the statistical parameters estimated from a search, and the size of the database.
`randseq`	Produce a random sequence with the same length and amino acid composition as a query sequence. Random sequences are useful in evaluating the accuracy of statistical estimates. In general in a database search, the highest scoring match to a random query sequence should have an expectation value E() ~1.

In addition, several programs in the FASTA2 package are not yet included with the FASTA3 programs (Table 3). As this chapter is written (summer, 1998), `lalign` is the most important program in the FASTA2 package that is not in the `fasta3` package. `lalign` (and the related graphical programs `plalign` and `flalign`) can produce multiple local alignments from the same pair of protein sequences, while `fasta3` and `fasta` produce only one alignment. Multiple local alignments can highlight domains with proteins; i.e. a protein may contain several domains

that share strong similarity with a library sequence. When multiple similar domains are present, `fasta3` only shows the most similar alignment; `lalign` is required to detect the alternative alignments.

In general, programs in the FASTA3 package are preferred over the older FASTA2 programs if FASTA3 has the function you need. Programs in the FASTA3 package have more robust statistical estimates and error handling, a larger variety of scoring matrices (`fasta3` has MDM10, MDM20, PAM120, and BLOSUM80 in addition to PAM250, BLOSUM50, and BLOSUM62 in `fasta2`), and a broader array of comparison functions (`fasty3`, `fastf3`, `tfasty3`, and `tfastf3`).

Table 3: Programs available only with FASTA2

`lalign/ plalign/ flalign`	Find multiple local alignments between two protein or DNA sequences using the `sim` implementation (23) of the Waterman-Eggert (24) algorithm. `lalign` shows traditional alignments; `plalign` produces graphics, while `flalign` produces graphics commands for the GCG figure program. This program performs successive full Smith-Waterman alignments, and is best used for protein alignments. For DNA, try `lfasta` (below).
`lfasta/ plfasta/ flfasta`	Find multiple local alignments between two protein or DNA sequences using the `fasta` algorithm. `lalign` uses the heuristic `fasta` algorithm with a local band-alignment. `lalign` is preferred for protein alignment, but `lfasta` is much faster for very long DNA sequences. `plfasta` and `flfasta` produce graphical output.
`prdf`	Like `prss3`, but uses the `fasta` algorithm instead of Smith-Waterman. `prss3` is preferred.
`align`	Global sequence alignment between two protein or DNA sequences using linear space (25).
`aacomp`	Reports amino acid composition and molecular weight of a protein sequence.
`grease/ tgrease`	Calculates the hydropathy plot of a protein sequence using the Kyte-Doolittle method (26). `tgrease` produces tektronix graphics.

2.1 Which Program Should I Use?

Many investigators who use the `fasta` program for protein and DNA database searches are unfamiliar with other programs in the package, or are unclear as to when they should be used. Table 4 suggests some strategies for using the programs in the FASTA3 package.

The suggestions in Table 4 are based on two rules-of-thumb: (1) use the program that is designed for your problem; and (2) whenever possible, search protein sequence databases before DNA sequence databases. Protein sequence comparison routinely reveals homologous sequences that diverged 2-3 billion years ago; it is difficult for DNA sequence comparison to “look-back”

more than 200-500 million years. Thus, protein sequence comparison, or translated DNA sequence comparison, allows one to identify homologs that diverged 5-10-times farther back in evolutionary time (Table 5).

Table 4: Which Program When?

Problem	Program	Explanation	Alternative
Identify unknown protein	(1) `fasta3`	General protein comparison. Use *ktup=2* (the unknown default) for speed; *ktup=1* for a more sensitive search. Search first against the smallest library likely to contain a homolog (i.e. SwissProt rather than Genpept).	blastp/
	(2) `ssearch3`	10-50-fold slower than `fasta3`, but provides maximum sensitivity. No advantage for DNA comparisons.	`fasta3/ blastp`
	(3) `tfastx3/ tfasty3`	If a homolog cannot be found in the protein databases, check the DNA databases with `tfastx3` or `tfasty3`. `tfasty3` provides more accurate alignments, but is about 33% slower.	`tblastn/ tfasta`[a]
Identify structural DNA sequence	`fasta3`	If the DNA sequence encodes a protein, use protein sequence comparison first, then try translated protein sequence comparison (`fastx3`/`fasty3`). For repeated DNA sequences or structural RNAs, search first with *ktup=6* (the default), then *ktup=3*. Search with *ktup< 3* only for very short sequences (PCR primers).	`blastn`
Identify EST sequence	`fastx3/ fasty3`	Protein sequence comparison is far more sensitive than DNA comparison, so check first to see if the EST encodes a product homologous to a known protein. Current version searches forward strand only, so use `fastx3 -i` as well.	`fasta3/ blastx/ tblastx`
Identify new orthologs	`tfastx3/ tfasty3`	If possible, search EST sequences from the same species. Use low/close MDM20 scoring matrices to detect close relationships and avoid distant relationships. Confirm statistical significance	`tblastn/ tblastx`
Confirm statistical significance	`prss3`	Use 500-2000 shuffles, and remember to normalize the statistical significance to the size of the database originally searched (typically 10,000 - 100,000 sequences).	
Confirm statistical estimates	`randseq`	Use to generate random sequences; then search using `fasta3` (or `blastp` or `ssearch3`) and look for E() ~1.0.	

[a]No longer recommended.

In addition, low complexity regions are relatively easily removed from protein sequence databases and recognized in protein sequence alignments, but they are much more difficult to recognize in DNA sequence alignments. These regions can produce statistically significant similarity scores for non-homologous sequences because of their unusual amino-acid

composition. Thus, when seeking to identify a newly sequenced EST (Expressed Sequence Tag) sequence, you should first use `fastx3` or `fasty3` to search a comprehensive protein database like SwissProt or PIR, then search a larger but more redundant database like the BLAST/NCBI `nr` or OWL (9) "non-redundant" protein databases, or Genpept, and, only after these searches have failed to turn up statistically significant matches should you look for DNA sequence matches.

Table 5: DNA vs. protein sequence comparison

The best scores are:		DNA E(188,018)	`tfastx3` E(187,524)	prot. E(331,956)
DMGST	D.melanogaster GST1-1	1.3e-164	4.1e-109	1.0e-109
MDGST1	M.domestica GST-1 gene	2e-77	3.0e-95	1.9e-76
LUCGLTR	Lucilia cuprina GST	1.5e-72	5.2e-91	3.3e-73
MDGST2A	M.domesticus GST-2 mRNA	9.3e-53	1.4e-77	1.6e-62
MDNF1	M.domestica nf1 gene. 10	4.6e-51	2.8e-77	2.2e-62
MDNF6	M.domestica nf6 gene. 10	2.8e-51	4.2e-77	3.1e-62
MDNF7	M.domestica nf7 gene. 10	6.1e-47	9.2e-77	6.7e-62
AGGST15	A.gambiae GST mRNA	3.1e-58	4.2e-76	4.3e-61
CVU87958	Culicoides GST	1.8e-41	4.0e-73	3.6e-58
AGG3GST11	A.gambiae GST1-1 mRNA	1.5e-46	2.8e-55	1.1e-43
BMO6502	Bombyx mori GST mRNA	1.1e-23	8.8e-50	5.7e-40
AGSUGST12	A.gambiae GST1-1 gene	2.3e-16	4.5e-46	5.1e-37
MOTGLUSTRA	Manduca sexta GST	5.7e-07	2.5e-30	8.0e-25
RLGSTARGN	R.legominosarum *gstA* and *gstR*	0.0029	3.2e-13	1.4e-10
HUMGSTT2A	H. sapiens GSTT2	0.32	3.3e-10	2.0e-09
HSGSTT1	H.sapiens GSTT1 mRNA	7.2	8.4e-13	3.6e-10
ECAE000319	E. coli hypothet. prot.	—	4.7e-10	1.1e-09
MYMDCMA	Methylophilus dichlorometh. DH	—	1.1e-09	6.9e-07
BCU19883	Burkholderia maleylacetate red.	—	1.2e-09	1.1e-08
NFU43126	Naegleria fowleri GST	—	3.2e-07	0.0056
SP505GST	Sphingomonas paucim	—	1.8e-06	0.0002
EN1838	H. sapiens maleylacetoacetate iso.	—	2.1e-06	5.9e-06
HSU86529	Human GSTZ1	—	3.0e-06	8.0e-06
SYCCPNC	Synechocystis GST	—	1.2e-05	9.5e-06
HSEF1GMR	H.sapiens EF1g mRNA	—	9.0e-05	0.00065

The primate, other mammal, invertebrate, and bacterial sections of Genbank were searched using a *Drososphila* glutathione transferase cDNA (`DMGST`) and protein (`gtt1_drome`) sequence using `fasta3` (DNA, *ktup=4*), `tfastx3`, and `fasta3` (protein, *ktup=2*). Expectation values for selected high scoring sequences are shown. DNA comparisons with "—" had expectation values E()>100 .With this query, DNA sequence comparison detects homologs only in other insects, while protein and translated DNA comparison finds statistically significant similarity with homologs from humans and bacteria.

2.2 FASTA vs. BLAST

The BLAST family of sequence comparison programs (10, 11) offers many of the same search capabilities as the FASTA programs (Table 6). In general, the BLAST programs are faster, but the FASTA programs can provide more accurate alignments. For most protein sequence database searching, the current `blastp2.0` (gapped blast, ref. 11) will identify an unknown protein as effectively as `fasta3` and even the more rigorous `ssearch3`. `fasta3` and `ssearch3` use different scoring matrices (BLOSUM50) and gap penalties (-12 for the first residue in a gap, -2 for each additional residue) from `blastp2.0` (BLOSUM62, -12 for the first residue in a gap, -1 for each additional residue). The previous `blastp1.4` produced very poor sequence alignments (because of the restriction on gaps); but the current `blastp2.0` version produces protein alignments that are very similar to those obtained with a rigorous Smith-Waterman search.

Table 6: Comparison of BLAST2 and FASTA3 Programs

Program		
BLAST	FASTA	Function
`blastp`	`fasta3`	General protein sequence similarity searches. blastp is faster and can show alignments between several domains in the same sequence. `fasta3` displays a Smith-Waterman final alignment and produces more accurate statistical estimates in some cases.
`blastn`	`fasta3`	DNA sequence comparison. `blastn` is highly optimized for speed; it uses a fixed word size (11 nucleotides) and scoring matrix that are inappropriate for some problems (e.g. searching for PCR primer matches). `blastn` searches with both strands of a DNA sequence. `fasta3` does not; two searches (`fasta3` and `fasta3 -i`) are required.[a]
`blastx`	`fastx3`/ `fasty3`	Compare a translated DNA to a protein sequence database. While `blastx` does six independent searches (one for each of the six frames), `fastx3` and `fasty3` effectively does a single forward (or backward) search, which allows frameshifts in computing the similarity score and alignments. As a result, `fastx3` and `fasty3` are more sensitive and can produce much better alignments than blastx when the DNA sequence has frameshift errors. blastx searches in the forward and reverse frames; `fastx3`/`fasty3` searches only in the forward or the reverse (`fasty3 –i`) frame.
`tblastn`	`tfastx3`/ `tfasty3`/ `tfasta`	Compare a protein sequence to a DNA sequence database, translating in the three forward and reverse frames. Again, `tfastx3` and `tfasty3` provide more accurate alignments than `tblastn` or `tfasta` when the DNA sequences have frameshift errors.
	`tblastx`	Compare a DNA query sequence to a DNA library, translating both sequences in all six frames and scoring using a protein substitution matrix (BLOSUM62). `fasta3` with *ktup=6* (the default) provides a similar function, but does not use a protein scoring matrix.

[a]The GCG implementation of `fasta` searches with both strands.

For translated DNA-protein comparison and DNA database searches, the FASTA programs are much better than their BLAST counterparts. Although the gapped `blastp2.0` performs very well in protein comparisons, `blastx` performs the three forward-frame searches separately, while `fastx3` and `fasty3` calculate a single alignment that allows frameshifts. Treating the all three forward reading frames as a single sequence makes it much easier to produce high quality alignments that extend across the length of the matched protein sequence and allows similarity from the different reading frames to be combined in a natural way to improve sensitivity. For example, a `blastx` search with a class-mu mouse glutathione transferase cDNA sequence with insertion and deletion errors at 5% of the positions detected only other class-mu glutathione transferases, while a search with the same sequence using `fasty3` detected more class-mu protein sequences with $10^{-20} < E() < 10^{-17}$ and an additional 8 more distantly related class-pi glutathione transferase sequences ($10^{-5} < E() < 0.01$).

The FASTA programs also provide additional flexibility for DNA sequence searches. Searches can be done with any "wordsize" (*ktup*) from 1-6; small *ktup*'s are particularly appropriate for searches with short sequences, such as PCR primers. In addition the FASTA programs can use a variety of scoring matrices, including matrices with very high mismatch penalties that can be used to identify long identities in sequences.

3. INTERPRETING FASTA STATISTICS

When rapid sequence comparison programs were first introduced in 1983 (1), it became possible to find similar DNA and protein sequences by searching sequence databases, but there was no formal basis for deciding whether a weak similarity was likely to be biologically significant. A Monte-Carlo shuffling method for evaluating similarity scores (`rdf`) was provided with the FASTP program (8), but the recommended guidelines for significant similarity ($Z > 5$) were not based on the correct statistical model for local similarity scores and did not account for database size. A sequence with a score that is 10 standard deviations ($Z > 10$) above the mean is expected 0.015 times by chance in a search of a 10,000 entry database; the same score would be expected 0.11 times by chance in a search of SwissProt (70,000 entries), and thus would not be statistically significant, even at the 0.05 level.

Accurate statistical estimates were introduced into similarity searching with the `blastp` program (10), based on the recognition that local similarity scores can be described accurately by the extreme value distribution (12, 13). The Monte-Carlo shuffling program introduced with `fastp` now uses the extreme value distribution to calculate the probability of an alignment score, and the library searching programs in the FASTA2 and FASTA3 packages provide a value that can be used to infer homology from statistically significant similarity the expectation (E()) value (6).

The E() value is the first number that you should look at when deciding whether to analyze further a high-ranking sequence alignment. Investigators often wonder what E() value they should use. This is discussed in detail in the next section, but in most cases, and E() value between 0.001 and 0.01 can be used to infer homology reliably, but lower (more conservative) values are required when hundreds or thousands of searches are performed (as when characterizing all the genes in a bacterial genome).

The E()-value calculated by the `fasta3` programs and `blast` programs is a statistical measure of the likelihood that the observed similarity score could have occurred by chance. Like any statistical measure, its usefulness depends on: (1) whether the assumptions of the underlying statistical model are correct, and (2) the kinds of errors that one is willing to accept when using the measure to draw a conclusion. For similarity searching, we infer homology (common ancestry) from "statistically significant" similarity. However, the threshold for "statistical significance" will vary, depending on whether we are more concerned about occasionally mis-identifying a non-homolog (labeling a sequence as related when it is not, a false positive or type I error) or missing a likely homolog (labeling a sequence as non-homologous when a high-scoring homolog has been found, a false-negative or type II error).

3.1 What threshold should I use to infer sequence homology?

For most molecular biologists, the greatest concern in similarity searching is a false-positive error; we don't want to send a letter to Nature identifying a yeast homolog of `p53_human` when no evolutionary relationship exists.[1] While incorrect assertion of homology was relatively common before accurate similarity statistics became available, it is rare today. (Unfortunately however, once the "observation" has been published, it is difficult to remove from the literature.) The E()-value or expectation calculated by `fasta3` *et al*. is the number of times you would expect to see a score equal or greater by chance in a search of the database. In other words, E() < 0.01 says that you expect to see a score that high (or higher) once by chance in 100 searches; E() < 0.001 says once in 1000 searches, etc. E() ~1 says that you expect to see a score that high, simply by chance, every time you do a search.

Older versions of the `blast` programs used a related statistic, the *p()*-value, to characterize the significance of a similarity score. The E()-value reported by the fasta programs ranges from *0..D*, where *D* is the number of entries in the database, while the `blast` *p()*-value ranges from 0..1. The probability (*p()*-value) of an E()-value can be found with the Poisson formula: $p(E)=1-e^{-E}$. For values of E() < 0.1, *p()* ~ E(), thus *p*(E = 0.1) = 0.1; *p*(E = 1.0) = 0.63; *p*(E = 5.0) = 0.99.

While a sensible E()-value threshold (0.001 - 0.01) can ensure that researchers avoid "false positive" errors, little can be done to avoid "false negatives," i.e. labeling a sequence as unrelated to anything in the database when in fact a homolog is present. Most diverse protein families contain pairs of related sequences that do not share statistically significant sequence similarity. Fortunately, if those families are large (e.g. globins, serine proteases, glutathione transferases, G-protein coupled receptors), it is likely that newly discovered family members will share significant similarity with some known members of the family. As the sequence databases grow more complete and protein families expand, the rate of false negatives should decrease.

3.2 Choosing a database

The expectation value E(S > x) of a similarity score is calculated from the probability of the pair-wise similarity score *p*(S > x), which can be calculated using the extreme value distribution (12, 13), and the number of "tests" (i.e. sequence comparisons) that were performed to find the

[1]The gold-standard test for homology is structural similarity. If the candidate yeast homolog of P53 has a completely different three-dimensional structure, the hypothesis is wrong.

high-scoring sequence. Thus, E(S > x) = *p*(S > x)*D*, where *D* is the number of sequences in the database. (For DNA sequence comparison, *D* is not the number of sequences in the database but the length of the database in nucleotides divided by the length of the query sequence.)

Because E() increases linearly with the number of database entries, a similarity found in a search of a bacterial genome with 1,000-5,000 entries will be 50-250-fold more significant than an alignment with exactly the same score found in the OWL non-redundant protein database (ref. 9; 250,000 entries). Thus, when searching for very distant relationships, one should always use the smallest database that is likely to contain the homolog of interest. If the goal is to find the *E. coli* homolog of the *B. subtilis* DAHP synthase (`arog_bacsu`), one should search the *E. coli* proteome (which finds the E. coli *kdsA* homolog with E(4,283) < 0.00015) rather than SwissProt (`kdsa_ecoli` E(74,417) < 0.0017) or OWL (`kdsa_ecoli` E(260,784) < 0.0085). Here, the same alignment, with the same similarity score, is 50-fold less significant against the largest database than with the smallest.

Likewise, a search of SwissProt (~70,000 entries) will be 3-5 fold more sensitive than either OWL (261,000 sequences) or the BLAST `nr` protein database (332,000 sequences), simply because Swissprot is smaller. Thus, an efficient strategy for identifying protein homologs should: (1) search smaller databases first; then (2) re-search a smaller database (like SwissProt) with a more sensitive algorithm (`fasta3` with *ktup=1* or `ssearch3`), and then, if no significant matches are found, (3) search larger databases (OWL or `nr`).

While their size reduces search sensitivity, larger databases can be effective when they provide more diverse members of a protein family. For example, the most distant `p53_human` homolog in SwissProt is a flounder sequence. OWL contains about twice as many novel p53 homologs, including one from squid.

3.3 Thresholds for large-scale sequence analysis

Genome sequencing centers and other groups that do thousands of similarity searches each day must use more conservative thresholds of statistical significance to avoid false positive errors. A threshold of E() = 0.001, which is conservative for someone who does a few searches a day, should produce 10 scores below the threshold between non-homologous sequences by chance after 10,000 searches. Indeed, if you do 100 searches with random sequences against the PIR or Swissprot databases, one of those 100 sequences will find a "homolog" with E() < 0.01, ten will have E() < 0.1, etc. (6). Genome sequencing centers typically use thresholds of E()<10^{-6}, or even lower, when characterizing thousands of sequences.

However, using a more conservative threshold of statistical significance ensures that you will make more false negative (type II) errors when looking at distant relationships. For example, in a comparison of 2608 human proteins from SwissProt against the E. coli proteome (4289 sequences), 417 obtained E() < 0.02, 373 had E() < 0.01, 301 had E() < 0.001, 256 had E() < 0.0001. Of the 72 with 0.001 < E() < 0.01, we would expect that about 26 (0.01 2608) shared similarity this high by chance, while the other 45 are truly homologous. (Unfortunately, we cannot identify which 45 sequences are homologs without additional information.) In the human/E. coli search, 209 sequences had E()<10^{-6}; we would expect all of these matches are genuine homologies. However, using the conservative 10^{-6} threshold would misidentify as "unrelated" almost 200 probable homologs. Thus, estimates of the number of "novel" or

"unidentified" proteins in newly sequenced bacterial genomes are generally overestimates, since many of these "novel" proteins may share significant similarity when searched individually, but not when searched in a group of 2,000-4,000 sequences.

3.4 Statistical estimates–what can you trust?

If the statistical estimates are accurate, the guidelines in the previous section provide a reliable strategy for identifying related sequences based on sequence similarity. However, with biological sequences (as opposed to "fair" coins), the assumptions underlying the statistical model may not be met. When the assumptions fail, the highest scoring unrelated sequence may have an expectation value that is much too low (e.g. E()<10^{-3}) or much too high (E() > 100). If the E()-value is too low, unrelated sequences will be mistakenly labeled as related (false positives). If the E()-values are too high, it is likely that the E()-values of related sequences are too high as well, and related sequences will be missed (false negatives).

In general, inaccurate statistical estimates are caused by either (1) incorrect gap penalties or (2) low complexity regions (runs of simple amino acid composition, e.g. `ggqgppgdaggpg` from a *C. elegans* collagen or `ssggvtfsvss` from a *Drosophia* trypsin) in the query sequence (3, 14). In the first case, the statistical model has failed. The statistical theory behind the estimates for BLASTP, FASTA and Smith-Waterman (`ssearch3` scores assumes that the scores are "local," i.e. on average, non-identical amino acids will have similarity scores $s_{ij} < 0$. If the gap penalties are too low, then the alignment algorithm will choose to insert a gap, rather than to end the alignment, and the alignment will tend to become "global," aligning the sequences from end to end. The statistical properties of "global" alignment scores are different from those of "local" scores. "Local" scores follow the extreme-value distribution; the distribution of "global" alignment scores is not well understood.

The reliability of the sequence statistics can be confirmed quickly by looking at the histogram of observed and expected similarity scores that is displayed after a `fasta3` search,[2] and by checking the expectation (E()) value of the highest scoring unrelated sequence.[3] If there is good agreement between the observed and expected distribution of scores and the E() value of the highest scoring unrelated sequence is ~1, the statistical estimates should be accurate.

---

[2]These examples show results from running the `fasta3` and `ssearch3` programs, which are distributed from `ftp://ftp.virginia.edu/pub/fasta/`. The programs available from this site run on most UNIX platforms (Digital UNIX, IBM AIX, Linux, SGI Irix, and Sun Solaris) as well as Windows (Windows95 and NT) and Macintosh. The output shown here may differ slightly from the FASTA program distributed with the Genetics Computer Group, but similar information is available from all modern FASTA implementations.

[3]Although identifying the highest scoring unrelated sequence seems to presume knowledge of the protein family, additional searches with candidate unrelated sequences (E() ~1) can often separate low scoring related from high scoring unrelated sequences (5).

Figure 1: Histogram of `fasta3` similarity scores

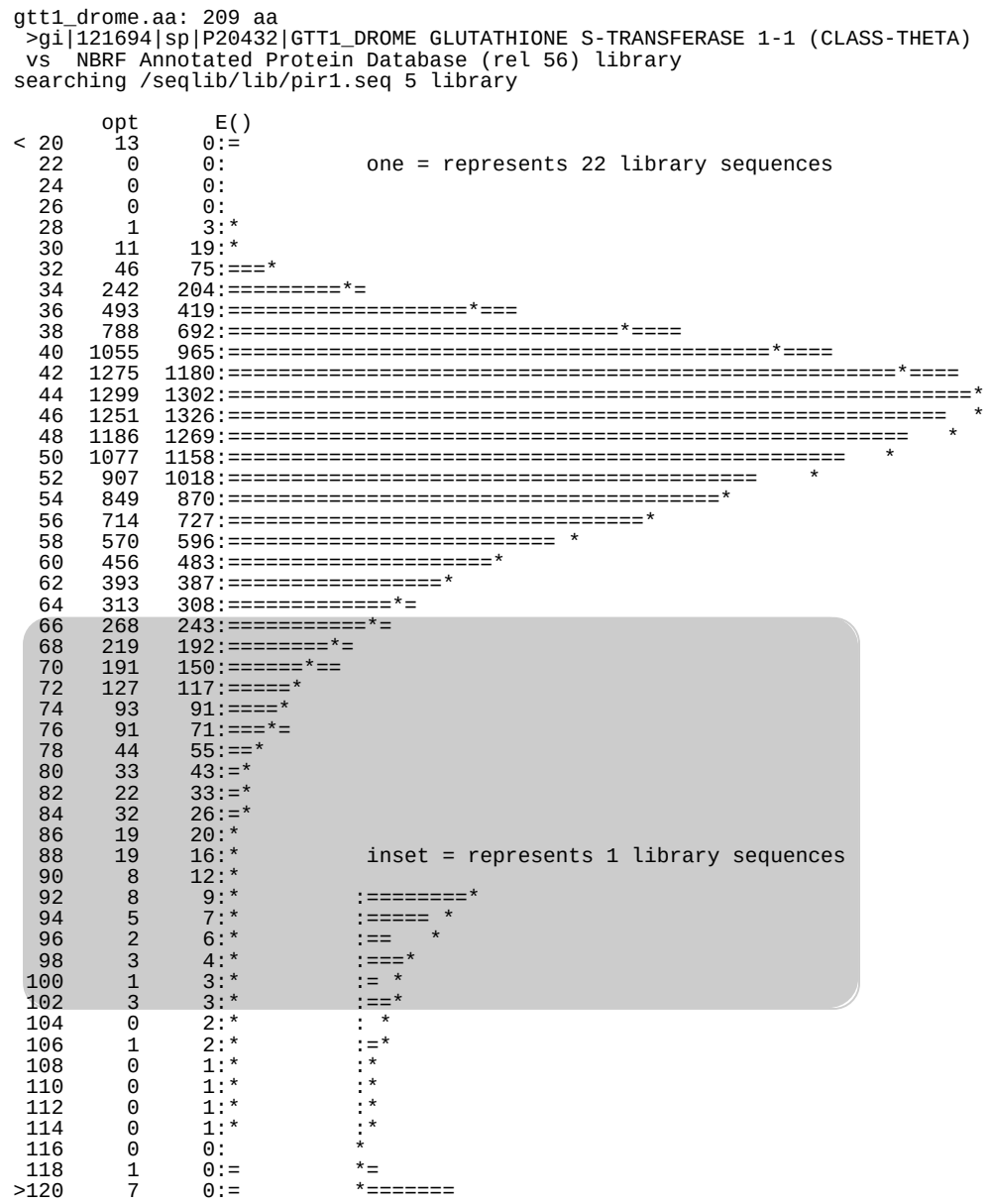

Fig. 1: Histogram of `fasta3` similarity scores—Results of a search of a *Drosophila* class-theta glutathione transferase (`gtt1_drome`) against the annotated PIR1 protein sequence database. The initial histogram output is shown. The shaded section indicates the region that is most likely to show discrepancies between observed and expected numbers of scores when the statistical model fails.

3.4.1 Low gap penalties cause inaccurate estimates

For most protein and DNA sequence searches, there is excellent agreement between the observed and expected distribution of scores (Fig. 1) and the E()-value of the highest scoring unrelated sequence is ~1.0 (Table 7; ref. 6). The FASTA programs provide a histogram summarizing the distribution of observed and expected scores after every search (Figs. 1-3). Fig. 1, reports that for this search, 788 sequences ("`opt`" column) in the database obtained scores of 38-39 (left-most column), while 692 sequences ("`E()`" column) are expected to have scores in that range for a database of 14,000 sequences. Agreement between observed ("`===`" graph) and expected ("`*`" in histogram) is especially important in the shaded area in Fig. 1. For many searches, it is also possible to confirm the accuracy of the estimates by looking for the highest scoring unrelated sequence in the list of high scoring sequences. In Table 7 the highest scoring unrelated sequences are `S30223` and `NOBY2`, with expectation values ~8. (Ideally, these scores would be a bit closer to 1; the highest scoring unrelated sequence in the same search with `ssearch3` has E() < 3.)

Table 7: FASTA search - high scoring sequences

Name	description	len	initn	opt	z-score	E()
XUFF11	glutathione transferase	209	1399	1399	1626.5	1.2e-84
XUZM32	glutathione transferase	222	133	173	210.9	8.6e-06
XUZM31	glutathione transferase	220	107	164	200.6	3.2e-05
XUZM1	glutathione transferase	213	123	144	177.7	0.00061
RGECSS	string. starv. prot. - E. coli	212	106	140	173.1	0.0011
XURTG	glutathione transferase	222	58	139	171.7	0.0013
XURT8C	glutathione transferase	222	39	115	144.0	0.046
XURTG4	glutathione transferase	218	40	93	118.7	1.2
A37378	glutathione transferase	210	40	82	106.2	5.8
S30223	elongation factor eEF-1g	227	34	80	103.5	8.3
NOBY2	*phosphopyruvate hydratase*	437	53	83	103.1	8.8
PWBYD	*H+-transporting ATP synthase*	212	53	79	102.7	9.2

High scoring sequences from searches of `gtt1_drome` against the annotated PIR1 database (27) with `fasta3` (*ktup=2*. High-scoring unrelated sequences are highlighted in *italics*.

Tables 8 and 9, and Fig. 2 show two examples of searches where the statistical model has failed. In the first case (Table 8), a DNA search was performed with gap penalties of -12 and -2, rather than the default -16, -4. While the histogram (not shown) shows good agreement between the observed and expected distribution of scores, the E()-value of the highest scoring unrelated sequence is 0.01. (That the high-scoring unrelated sequence does not contain a homolog was confirmed by scanning it with `tfasty3`). Moreover, the E()-values for homologous alignments increase by 10^7 (e.g. from 1.2×10^{-12} to 0.0008 for AC002520; Table 8) when the gap penalties are reduced from –16/-4 to –12/-2. DNA sequence searches with even lower gap penalties do show sizeable differences between the observed and expected distribution of scores, but the E()-value of the highest unrelated sequence is usually the most sensitive measure of the accuracy of the statistical estimates.

Figure 2: Poor statistics: low complexity regions

```
grou_drome.aa: 719 aa
 >GROU_DROME GROUCHO PROTEIN (ENHANCER OF SPLIT M9/10). - DROSOPHILA MELANOGAS
 vs  NBRF Annotated Protein Database (rel 56) library
searching /seqlib/lib/pir1.seq 5 library

       opt      E()
< 20    13     0:=
  22     0     0:           one = represents 28 library sequences
  24     1     0:=
  26     0     0:
  28     1     3:*
  30    10    20:*
  32    21    76:= *
  34   105   205:====   *
  36   272   422:==========     *
  38   540   697:====================    *
  40   937   972:==================================*
  42  1269  1188:=========================================*===
  44  1645  1311:================================================*============
  46  1666  1335:================================================*============
  48  1577  1278:============================================*===========
  50  1310  1166:=======================================*=====
  52  1056  1025:===================================*=
  54   851   876:============================*
  56   669   732:========================  *
  58   423   601:================     *
  60   419   487:===============  *
  62   255   390:==========   *
  64   196   310:=======    *
  66   181   245:======= *
  68   154   193:======*
  70    99   151:==== *
  72    74   118:=== *
  74    63    92:===*
  76    60    72:==*
  78    47    56:=*
  80    48    43:=*
  82    36    33:=*
  84    33    26:*=
  86    27    20:*
  88    21    16:*          inset = represents 2 library sequences
  90    18    12:*
  92    20     9:*         :====*=====
  94    20     7:*         :===*======
  96    17     6:*         :==*======
  98     7     4:*         :=*==
 100    10     3:*         :=*===
 102    11     3:*         :=*====
 104    10     2:*         :*====
 106    11     2:*         :*=====
 108     7     1:*         :*===
 110    10     1:*         :*====
 112     6     1:*         :*==
 114     4     1:*         :*=
 116    11     0:=         *======
 118    10     0:=         *=====
>120    70     0:===       *==================================
5446221 residues in 14321 sequences
 Expectation_n fit: rho(ln(x))= 8.0964+/-0.00108; mu= 4.7475+/- 0.061;
 mean_var=157.6967+/-31.622, 0's: 13 Z-trim: 96  B-trim: 33 in 1/62
 Kolmogorov-Smirnov  statistic: 0.0497 (N=29) at  52
```

Fig. 2: Poor statistics: low complexity regions—A fasta3 search (*ktup=2*) of the PIR1 database using grou_drome. The histogram of sequence similarity scores is shown. In this case, there are clear discrepancies between the observed and expected numbers of sequences with scores in the central part of the distribution and in the tails, and there is an excess of high scoring sequences. Table 9 shows that all of these excess high-scoring sequences are unrelated.

Table 8: FASTA search - low gap penalties

The best scores are:		(length)	initn	opt	z-sc	E(-12/-2)	E(-16/-4)
AC002520	Human Chr. 1p13	(11901)	1507	404	173.1	0.0008	1.2e-12
AC000031	Human Chr. 1p13.3	(39043)	1396	394	161.0	0.0011	6.5e-12
HSU47924	*Human chr. 12p13*	(78864)	235	352	138.3	0.01	2.0
AC000032	Human Chr. 1p13	(29867)	1354	345	141.6	0.018	6.6e-09
CACD42	*C.atys CD4 mRNA*	(1189)	69	307	146.1	0.26	—
HUMDXS455A	*Human cosmid*	(38409)	126	274	109.2	0.89	—
HSHS12ENH	*Homo sapiens DNA*	(3735)	151	278	126.1	1.1	0.038
HSV411C11	*Human DNA*	(5637)	165	276	122.5	1.1	—
HUMHSLA	*Human hormone-sens.*	(3255)	63	275	125.7	1.3	—
AF031078	*Human chr. X*	(78864)	188	264	100.2	1.4	0.078
AF035180	*Human chr. 4q35*	(4638)	67	271	121.7	1.5	0.08

High-scoring sequences from a fasta3 search (*ktup=6*) of the Primate division of Genbank 106 (~80,0000 sequences) using the reverse complement of a m*Gstm1* cDNA sequence (MUSGLUTA) using the default substitution matrix (+5/-4) and low (-12/-2) or default (-16/-4) gap penalties. Unrelated sequences are highlighted with *italics*. The low gap penalties improve the E()-value of the unrelated HSU47924 sequence to E() < 0.01 and reduce the significance of the homologous AC002520, AC000031, and AC000032 sequences by 10^7 .

3.4.2 Low E-values from low-complexity regions

Low E()-values between non-homologous sequences are usually caused by low complexity regions (3,14). The *Drosophila* "groucho" protein sequence (grou_drome) contains only 5 low complexity regions (83 of 719 residues as determined by seg, ref. 14), but as comparison of Fig. 2 and Fig. 3 shows, matches in these regions significantly distort the distribution of the high-scoring unrelated sequences. In contrast, a search with the 5 low-complexity regions masked (Fig. 3) shows the expected distribution of scores. Examination of the list of high-scoring sequences in the low- complexity search (Table 9) shows a large number of "significant" matches (0.00013 < E() < 0.02) to unrelated proteins with biased amino-acid compositions, while the highest scoring unrelated sequence in the "seg-ed" search has E() < 0.047 Perhaps surprisingly, the significance of the related GTP-binding regulatory protein similarity scores improve almost 1000-fold as well (Table 9).

Table 9: FASTA search – low complexity regions

Search with complete grou_drome:		length	initn	init1	opt	z-sc	E(14,212)
RGHUB1	GTP-binding reg. prot.	(340)	161	147	237	197.4	4.9e-05
RGHUB3	GTP-binding reg. prot.	(340)	163	152	233	194.2	7.4e-05
RGBOB2	GTP-binding reg. prot.	(326)	181	149	228	190.5	0.00012
PIHUB6	*salivary proline-rich prot*	(392)	142	142	229	190.1	0.00013
RGKWB	GTP-binding reg. prot.	(340)	159	154	222	185.4	0.00023
RGFFBH	GTP-binding reg. prot.	(340)	169	144	219	183.0	0.00031
PIHUSD	*proline-rich glycoprot.*	(310)	141	141	217	182.0	0.00035
PIRT3	*acidic proline-rich protein*	(206)	138	138	212	180.7	0.00042
WMBEW6	*capsid protein - herpes*	(635)	101	101	206	168.7	0.002
S23447	*annexin XI form B-bovine*	(505)	84	84	202	166.9	0.0024
PIHUPF	*salproline-rich glycoprot.*	(251)	147	147	193	164.3	0.0034
PIHUSC	*proline-rich phosphoprot.*	(166)	88	88	180	156.6	0.0092

Search with complete grou_drome:		length	initn	init1	opt	z-sc	E(14,212)
CGHU6C	*collagen alpha 1 (II)*	(1487)	104	104	197	156.0	0.0099
RGOOBE	GTP-binding reg. prot.	(341)	156	125	181	152.8	0.015
FOLJSP	*gag polyprotein - foamy vir*	(811)	121	121	187	151.9	0.017
CGBO1S	*collagen alpha 1 (I)-bovine*	(779	88	88	185	150.6	0.02
LUDO7	*annexin VII - slime mold*	(462)	88	88	179	149.2	0.024
CGHU2S	*collagen alpha 2 (I)*	(1366)	88	88	187	148.6	0.026
LUBO11	*annexin XI form A-bovine*	(503)	84	84	177	147.1	0.031
S09257	*Hox A4 - chicken*	(309)	116	116	172	146.2	0.035
OZZQMY	*circumsporozoite prot pre.*	(367)	146	146	172	145.1	0.04
Search with seg-ed grou_drome: (low complexity regions removed)							
RGHUB1	GTP-binding reg. prot.	(340)	161	147	237	247.5	8e-08
RGHUB3	GTP-binding reg. prot.	(340)	163	152	233	243.3	1.4e-07
RGHUB2	GTP-binding reg. prot.	(340)	181	149	228	238.1	2.7e-07
RGKWB	GTP-binding reg. prot.	(340)	159	154	222	231.9	5.9e-07
RGFFBH	GTP-binding reg. prot.	(340)	169	144	219	228.7	8.9e-07
RGOOBE	GTP-binding reg. prot.	(341)	156	125	181	189.1	0.00014
BVBYMS	*MSI1 protein - yeast*	(422)	116	74	139	143.9	0.047
ERHUAH	*coatomer complex alpha*	(1224)	109	109	134	131.7	0.23
I37062	*involucrin S - gorilla*	(495)	129	81	115	117.8	1.3

Unrelated sequences are highlighted in *italics.*

For protein-protein database searches, removal of low-complexity sequences is equally effective for either the query sequence or the protein database. However, it is more difficult to remove low-complexity regions from DNA query sequences, such as EST sequences. Unfortunately, high-scoring alignments between low-complexity protein sequences and out-of-frame DNA translations are common (15). A simple strategy for improving the sensitivity of translated DNA searches (fastx3, fasty3, or blastx) is to search against a "seg-ed" protein database (14).

Low-gap penalties and low-complexity regions produce unreliable statistical estimates because the underlying assumptions of the statistical model do not apply. Low gap penalties cause alignments to shift from local to global; extreme-value alignment statistics apply only to local alignments. Low-complexity regions violate implicit assumptions about higher-order structure in the "unrelated" sequences. With low-complexity sequences the matches are statistically significant but not biologically significant, because the statistical model assumed that each position of a random (unrelated sequence) is independent of all the others.

When the statistical model is valid—local alignments and truly "random" unrelated sequences—statistically significant similarity scores can be used to infer homology reliably. And one can usually check that the statistical model is correct by looking at the histogram of observed and expected similarity scores, and by checking the expectation value of the highest scoring unrelated sequence.

Figure 3: Accurate statistics with “seg-ed” query

```
grou_drome.seg: 719 aa
 >GROU_DROME GROUCHO PROTEIN (ENHANCER OF SPLIT M9/10). - DROSOPHILA MELANOGAS
 vs  NBRF Annotated Protein Database (rel 56) library
searching /seqlib/lib/pir1.seq 5 library

       opt      E()
< 20    48     0:==
  22    14     0:=          one = represents 24 library sequences
  24    21     0:=
  26    37     0:==
  28    39     3:*=
  30    65    20:*==
  32    95    76:===*
  34   175   206:========*
  36   348   424:===============  *
  38   591   700:========================    *
  40   891   977:====================================  *
  42  1141  1194:================================================ *
  44  1328  1317:=========================================================*=
  46  1373  1342:==========================================================*==
  48  1395  1285:============================================================*=====
  50  1227  1172:=============================================*===
  52  1107  1031:=========================================*====
  54   888   880:====================================*
  56   723   735:==============================*
  58   602   604:=========================*
  60   490   489:====================*
  62   357   392:=============== *
  64   284   312:============*
  66   246   246:==========*
  68   177   194:========*
  70   131   152:======*
  72   110   119:====*
  74    64    93:===*
  76    76    72:==*=
  78    53    56:==*
  80    41    43:=*
  82    44    33:=*
  84    22    26:=*
  86    26    20:*=
  88    17    16:*          inset = represents 1 library sequences
  90    11    12:*
  92    14     9:*         :========*=====
  94     5     7:*         :===== *
  96     7     6:*         :=====*=
  98    11     4:*         :===*=======
 100     2     3:*         :==*
 102     5     3:*         :==*==
 104     3     2:*         :=*=
 106     1     2:*         :=*
 108     1     1:*         :*
 110     0     1:*         :*
 112     1     1:*         :*
 114     0     1:*         :*
 116     0     0:          *
 118     1     0:=         *=
>120    13     0:=         *==============
5446221 residues in 14321 sequences
 Expectation_n fit: rho(ln(x))= 6.3481+/-0.00105; mu= 10.5411+/- 0.059;
 mean_var=92.0111+/-17.844, 0's: 13 Z-trim: 24  B-trim: 593 in 1/62
 Kolmogorov-Smirnov  statistic: 0.0129 (N=29) at  42
```

Fig. 3: Accurate statistics with “seg-ed” query—The search in Fig. 3 was performed using the grou_drome sequence with low-complexity sequences masked using the “seg” program (14). With low complexity sequences removed, the numbers of observed and expected similarity scores agree closely. Identical results are obtained when low-complexity regions are removed from the PIR1 database instead of grou_drome.

Table 10: FASTA3 general options

-a	show full sequences rather than only overlapping region (fastx/y3 and tfastx/y3 do not provide this feature)
-b #	number of best scores to show (must be < -E cutoff)
-d #	number of best alignments to show (must be < -E cutoff)
-E #	Expectation value limit for displaying scores and alignments. (By default, 10.0 for protein sequence comparisons; 5.0 for fastx/y3, and 2.0 for DNA sequence comparisons.)
-H	turn off histogram display
-I	(DNA only) reverse complement the query sequence; by default `fasta3`, `fastx3`, and `ssearch3` search only with the forward sequence. (tfastx/y3) compare against only the reverse complement of the library sequences.
-L	report long sequence description in alignments
-m 1-6,10	alignment display options (Table 14)
-n	force query to nucleotide sequence (default: autodetect)
-N #	read database in chunks of # residues. # should be > 2-times the query sequence length, as the chunks overlap by the length of the query. (default: 80,000 query-length)
-O file	send output to file
-q/-Q	quiet option; do not prompt for input
-r file	save all scores to statistics file
-S #	offset substitution matrix values
-s name	scoring matrix. BLOSUM50 is used by default for proteins, PAM120, PAM250, and BLOSUM62 can be specified by setting -s P120, P250, or BL62. Additional matrices include: BLOSUM80 (BL80), and MDM_10, MDM_20, MDM_40 (M10, M20, M40, 19). Alternatively, BLASTP1.4 for- mat scoring matrix files can be specified.
-w #	line width for similarity score and sequence alignment output
-W #	amount of sequence context around the alignment. Default is 30 residues (not used by fastx/y3, tfastx/y3).
-x "#,#"	offsets query and library sequence for numbering alignments
-z #	specify statistics calculation. Default is -z 1. Table 13.
-Z #	specify the size of the library to be used for statistical significance estimates.

4. FASTA3 PROGRAM OPTIONS

The behavior of the programs in the FASTA package can be modified with a variety of command line options; options are available to change the scoring matrix and gap penalties, use alternate statistical estimation methods, and change the format of the alignment output. Many of the options apply to all of the programs in the package (Table 10); other options are specific to `fasta3` or tfastx/y3 (Table 11). When using the FASTA programs distributed from the U. of Virginia, command line options must precede other program arguments. The standard invocation of a FASTA program is:

```
program -opt1 -opt2 arg2 -opt3 query_file library ktup-opt
```

specifically:

```
fasta3 -q -f -14 -w 75 -L -m 1 mgstm1.aa /slib/swissprot 1
```

In the latter case, the `fasta3` program is run in “quiet” (-q) mode with a penalty for the first residue in a gap of -14 (`-f -14` rather than the default -12), alignments are printed at 75 residues per line (`-w 75`), a long description of the library sequence is shown with the alignment (`-L`), and the alignment symbol highlights the differences rather than similarities (`-m 1`). Fig. 4

shows the difference between a conventional alignment (Fig. 4A) and one produced with the command line options shown above (B).

Table 11:

`fasta3`, `fastx/y3`, `tfastx/y3`, `tfasta3` options	
`-1`	sort by "init1" score
`-3`	(`tfasta3`, `tfastx3`, `tfasty3` only) use only forward frame translations
`-A`	force Smith-Waterman alignment for output. Smith-Waterman is the de fault for protein sequences, `fastx/y3`, and `tfastx/y3`, but not for `tfasta3` or DNA comparisons with `fasta3`.
`-c #`	threshold for band optimization
`-f #`	penalty for the first residue in a gap
`-g #`	penalty for additional residues in a gap
`-h #`	fastx/y3, tfastx/y3 only - penalty for a frameshift between codons
`-j #`	`fasty3`, `tfasty3` only - penalty for a frameshift against a codon
`-t #`	translation table - `fastx/y3`, `tfastx/y3`, and `tfasta3` now support the BLAST translation tables. See `http://www.ncbi.nlm.nih.gov/htbin-post/Taxonomy/wprintgc?mode=c/`
`-y #`	Width for band optimization; by default 16 for DNA and protein *ktup= 2*; 32 for protein *ktup= 1*
`ssearch3` command line options	
`-f #`	penalty for first residue in a gap
`-g #`	penalty for additional residues in a gap

Fig. 4 goes near here.

Command line options can be divided into five general categories: (1) scoring parameter options, (2) statistics options, (3) algorithm-specific options, (4) file specification options, and (5) output options.

4.1 Changing the scoring parameters

All the programs in the FASTA3 package calculate sequence alignments using two types of scoring parameters: a substitution matrix and gap penalties. The default scoring matrix, gap penalties, E() value cutoff, and comparison algorithm are shown in Table 12. The `fasta3`, `ssearch3`, `fastx/y3` and `tfastx/y3` programs use the BLOSUM50 scoring matrix (16) for protein sequence (and translated protein sequence) comparisons. Alternate protein scoring matrices can be specified with the -s option. Available protein matrices include BLOSUM62 (`-s BL62`) and BLOSUM80 (`-s BL80`), PAM250 (`-s P250`) and PAM120 (`-s P120`) (17, 18), and low evolutionary distance matrices MDM10 (`-s M10`) and MDM20 (`-s M20`) (19). In addition, any scoring matrix can be used by providing a file name for the file containing the substitution values (`-s matrix.file`). Version 3 of the FASTA programs uses the same substitution matrix format as the `blastp` programs, and the `pam` program distributed with the BLAST package can be used to generate appropriately formatted matrices.

Table 12: FASTA Program Defaults

program	query	library	scoring(-s)	gap (-f, -g)	frameshift	-E()	alignment

			matrix	penalties	(-h,-j)	cutoff	
`fasta3`	protein	protein	BLOSUM50	-12/-2		10.0	Smith-Waterman
	DNA (1 strand)	DNA	+5/-4	-16/-4		2.0	band Smith-Waterman[a]
`ssearch3`	protein	protein	BLOSUM50	-12/-2		10.0	Smith-Waterman
	DNA (1 strand)	DNA	+5/-4	-16/-4		2.0	Smith-Waterman
`fastx3`	DNA (1 strand)	protein	BLOSUM50	-15/-2	-20	5.0	Smith-Waterman[b]
`fasty3`	DNA (1 strand)	protein	BLOSUM50	-15/-2	-20/-20	5.0	Smith-Waterman[b]
`tfastx3`	protein	DNA	BLOSUM50	-15/-2	-20	5.0	Smith-Waterman[b]
`tfasty3`	protein	DNA	BLOSUM50	-15/-2	-20/-20	5.0	Smith-Waterman[b]
`fastf3`	mixed peptides	protein	MDM20			5.0	
`tfastf3`	mixed peptides	DNA	MDM10			5.0	

[a] ref. 28; [b] ref. 15

For DNA sequence comparisons, the substitution matrix scores +5 for a match and -4 for a mismatch (+2 for match to an ambiguous nucleotide, -1 for a mismatch to an ambiguous residue). Alternate DNA substitution matrices can be specified using the -s dna-matrix.file option.

The BLOSUM50 matrix works well for recognizing very distant relationships (and works well for long, closely related sequences as well). Searches with short sequences (18) or for closely related sequences (e.g. mouse proteins against mouse ESTs) will be more effective with "shallower" scoring matrices—matrices like MDM10 and MDM20 that are optimum for small amounts of change in very short sequences.

Gap penalties in the FASTA programs can be changed with the -f and -g options; -f specifies the cost of the first residue in a gap and -g specifies the cost of each additional residue. An alternate representation of gap penalties takes the form: $q + rk$, where q is the penalty for opening a gap and r is the penalty for each residue in the gap (k is the length of the gap). Thus, `-f -12`, `-g -2` (the default for protein searches) is equivalent to: $q = 10$, $r = 2$. Protein substitution matrices like BLOSUM50 and PAM250, which are scaled in 1/3-bit units (18), work well with gap penalties of `-12/-2` or `-14/-2` (20), while scoring matrices like BLOSUM62 and PAM120, which are scaled in 1/2-bit units, work well with a lower initial residue penalty, (`-f -8`).

Just as "shallower" substitution matrices may be appropriate for comparisons between closely related sequences (e.g. mammals), higher gap penalties may be appropriate as well.

Using a MDM20 scoring matrix with gap penalties of -20/-4 will cause the program to recognize, with very high expectation values, sequences that have diverged by about 20-40%, but the program will probably miss clear homologs that share less than 30% protein sequence identity.

The `fastx3`/`tfastx3` and `fasty3`/`tfasty3` programs provide additional gap parameters. `fastx3`/`tfastx3` uses `-h` to specify the cost of a frameshift (which must, because of the nature of the `fastx3` algorithm, fall between two codons). `fasty3`/`tfasty3` uses `-h` to set the cost of a between-codon frameshift and `-j` to specify the cost of a frameshift that within a codon. When searching with EST sequences that contain approximately 5% errors, the default values `-h -20` and `-j -20` work well (15). However, if the DNA sequences are known to be relatively error free, searches with higher frameshift penalties are appropriate, as they will reduce the noise from out-of-frame alignments.

In general, the default gap parameters provided by the FASTA programs are at the lower end of the useful range. Reducing the gap penalties more will often cause alignments to shift from local to global, and thus violate the assumptions underlying the statistical estimates. Small increases in the initial residue (`-f`) penalty will sometimes slightly improve the expectation value of an alignment, but researchers should be suspicious of borderline scores that change dramatically with different gap penalties. Changes in substitution matrices usually have a greater effect than small changes in gap penalties; the expectation values from searches with the PAM250 matrix are often 10^{-3}–10^{-10} lower than when BLOSUM50 is used. For example, for the scores shown in Table 7, the E()-values for the alignments of `gtt1_drome` and `xuzm32`, `xuzm31`, and `xuzm1` drop from 8.5×10^{-8}, 2.5×10^{-6}, and 8.8×10^{-5} to 7.1×10^{-5}, 0.001, and 0.15 when the PAM250 matrix is used. When evaluating the significance of an alignment using the Monte-Carlo `prss3` program, one should be certain to use the same substitution matrix and gap penalties.

4.2 Alternate statistical estimates

One of the strengths of the FASTA3 package is its ability to estimate accurately the statistical significance of a local similarity score, regardless of whether it was calculated from a protein:protein, DNA:DNA, or protein:translated-DNA alignment. The programs in the FASTA3 package calculate expectation values based on parameters estimated from the distribution of scores from "unrelated" sequences. Thus, the statistical estimates are accurate for the typical case of a search against a database containing tens of thousands of unrelated sequences, but they will not be accurate if the database does not contain unrelated sequences. The FASTA3 programs provide six statistical estimation options (Table 13; ref. 6). The `-z 3` option is of particular interest, as it can be used when searching databases that do not contain unrelated sequences, or even when comparing a pair of sequences.

Table 13: Statistics options

`-z –1`	No statistical estimates. Sometimes necessary when there are no unrelated sequences in the database.
`-z 0`	Unscaled statistical estimates. Estimates are calculated from the mean and and variance of the sequence similarity scores. Typically used when all of the library sequences have about the same length.

`-z 1`	Regression-scaled estimates. Mean and variance of the similarity scores are calculated after correcting the scores for a log(n) effect.
`-z 2`	Log-corrected estimates. Provided for historical purposes only; this method is out of date and should not be used.
`-z 3`	Altschul-Gish estimates (protein only). Instead of estimating the parameters from the data, pre-calculated parameters published by Altschul and Gish (29) are used. `-z 3` is the only option for estimating the significance of an alignment when unrelated sequences are not the majority of the searched library.
`-z 4`	An alternative to `-z 1` that uses a different method for removing high scoring, potentially related sequences during the parameter estimating process.
`-z 5`	An alternative the `-z 1` that also uses regression of the score variance with log(n) (library sequence length). While `-z 5` is likely to provide somewhat more accurate estimates than `-z 1`, it is also more sensitive to problems with the data, particularly when relatively small libraries (< 500 entries) are searched.

The dependence of statistical significance on database size can complicate comparisons of searches on different databases. The "`-Z number`" option can be used to force the program to pretend that a database of size "number" was searched, e.g. "`-Z 100000`" might be used to reflect the consensus that there are ~100,000 mammalian genes. ("number" should never be smaller than the actual size of the database searched.) This option is particularly important in combination with –z 3 when searching a small set of pre-selected sequences.

4.3 Input options

The FASTA programs provide a number of options that change how the query sequence is used and how the database is selected (Table 14). The most commonly used input option is `-i`, which causes a DNA search to use the reverse complement of the query sequence. (Unlike BLASTN and the GCG version of FASTA, the U. of Virginia FASTA programs do not automatically search with both the forward and reverse DNA strands when a DNA query is used.)

Table 14: Input options

`@`	In addition to using file names, the FASTA3 programs can accept query sequences from the stdin file stream on Unix and Windows computers. In this case, all information must be given on the command line, e.g.: `fasta3 -q @ /slib/swiss.seq 1 < query.aa` indicates that the input will come from `stdin (< query.aa)` and that the `swiss.seq` library will be searched with *ktup=1*. The @ option is most commonly used with `perl` scripts on WWW servers.
`:#-#`	Specify a sub-sequence. Query sequence file names can be followed by a ":" and a range of numbers to specify a portion of a sequence. If the first number is not given, 1 is assumed. If the last number is not given, the subsequence extends to the end of the sequence. Thus, `gtt1_drome.aa:51-150` specifies the 100 residues beginning at residue 51. Subsequence ranges can be given when the query sequence is entered on the command line or when prompted by the program. They can also be entered

	after an "@" (stdin) symbol. Subsequence ranges can only be used for the first (query sequence).
`-i`	(DNA queries only) Search with the reverse complement of the query sequence.
`-l file`	Identify the FASTLIBS file used to locate sequence databases.
`-n`	Force the input (query) sequence to be read as DNA (`fasta3` and `ssearch3` only).
`-N #`	Read long library sequences (such as bacterial genomes) in chunks of "#" residues; e.g. -N 5000 would read long sequences in 5000 residue portions.
`-q/Q`	Quiet. Do not prompt for input.

The FASTA programs make it easy to specify a search with only part of the query sequence with the ":" modifier to the query sequence file name. The command:

```
fasta3 gtt1_drome.aa:1-100 s
```

searches the database specified by the "`s`" abbreviation with the first 100 residues of the query sequence `gtt1_drome`.

`fasta3` and `ssearch3` use a simple algorithm to decide if a query sequence is likely to be protein or DNA. If the sequence is more than 85% A+C+G+T, it is assumed to be DNA; otherwise it is treated as a protein sequence. The `-n` option forces a query sequence to be treated as DNA; the `-n` option is required for DNA sequences provided through the `stdin` (`@`) option (Table 14). Unlike the BLAST programs, the FASTA programs currently report only the best alignment between the query sequence and the library sequence, even when the library sequence is very long and may contain hundreds of genes. By default, FASTA breaks up long DNA sequences into ~80,000 nucleotide pieces, but this size is too large for gene dense bacterial, yeast, and C. elegans genomes. The `-N 5000` option tells `fasta3` and `tfastx/y3` to read long DNA sequences in chunks of 5000 nucleotides. This is essential when scanning large, gene dense DNA sequences.

Table 15: Output options

`-a`	(`fasta3` and `ssearch3` only) show the query and library sequences in their entirety, not just the portion that aligns.
`-A`	(`fasta3` DNA only) `fasta3` does a full Smith-Waterman (22) alignment for protein sequences (and translated `fastx/y3` and `tfastx/y3` alignments) but only a band-limited alignment for DNA:DNA alignments. The -A option forces `fasta3` to do a full Smith- Waterman alignment for DNA sequences. This can slow the program down substantially if one of the sequences is quite long.
`-b #`	The number of high-scoring library sequences scores to be shown.
`-d #`	The number of high-scoring alignments to be shown.
`-E #`	The expectation (E()) value cutoff for showing scores and alignments. By default, `-E 10` for protein:protein comparisons, `-E 5` for translated DNA:protein comparisons, and `-E 2` for DNA:DNA comparisons. The -E cutoff overrides the `-b` and `-d` options; to ensure that at least 20 scores and 5 alignments are shown, the options: `-E 1000.0 -b 20 -d 10` would be used.
`-F #`	A lower-bound expectation value cutoff that prevents very closely related sequences from being shown. `-F 1e-4` will prevent the programs from

	showing library sequences with E() <10^{-4} . This option is useful for focussing on distant homologues in large protein families with many close homologues.
`-H`	Do not show the histogram.
`-L`	Provide long sequence descriptions with the alignment. Some sequence library formats (particularly reformatted GCG libraries) include a lot of uninformative text before the actual sequence description. With the `-L` option, all the sequence description available is displayed with the alignment.
`-m #`	See Table 16.
`-O file`	Send results to "file". Unix and Windows users should use the "> file" method for output redirection.
`-r file`	Send intermediate results for all sequences to "file".
`-w #`	Width of alignment output. The FASTA programs display alignments with 60 residues per line by default; this width can be increased to 200 residues with the -w option.
`-W #`	Amount of sequence context. `fasta3` and `ssearch3` provide neighboring sequence context in the alignment (translated fastx/y3 and tfastx/y3 do not). The amount of context is typically one half of an output line, but this amount can be increased or reduced with the `-W` option.
`-x "# #"`	Sequence coordinates. Normally, the FASTA programs assume that each sequence begins at residue 1. On occasion, it is useful to use a different initial coordinate, such as when comparing a cDNA to the encoding gene or when working with only a portion of a sequence. -x “1 -751” would tell `fasta3` to begin the numbering of the library sequence at “-751” rather than “1”. On Unix, DOS, and Macintosh systems, the two numbers must be surrounded by double quotation ("...") marks.

4.4 Changing the output appearance

Many of the FASTA command line options change the appearance of the alignment output (Table 15). Options are available to change the number of residues displayed on an alignment line, to change the numbering of the residues, and to change the format of the alignment. Two options are of particular interest: `-m 5` provides both the sequence alignment and a crude graphical mapping of the aligned region against the query sequence. This graph makes it much easier to see quickly the parts of the query that align with the different library sequences, and thus can highlight query sequences with separable domains. The `–m 6` option is identical to `-m 5`, but provides `html` mark up commands and links to Entrez and other sites for re-searching to confirm relationships with the library sequence.

Figure 4: Alternative output formats

A.

```
>>GTT1_MUSDO GLUTATHIONE S-TRANSFERASE 1 (EC 2.5.1.18) (C (208 aa)
 initn: 1229 init1: 1229 opt: 1230 Z-score: 1472.4 expect() 2.3e-75
Smith-Waterman score: 1230;  85.024% identity in 207 aa overlap

               10        20        30        40        50        60
gi|121 MVDFYYLPGSSPCRSVIMTAKAVGVELNKKLLNLQAGEHLKPEFLKINPQHTIPTLVDNG
       .:::::::.:::::.:::::.::::::::::::::::::::::::::::::::::::.
GTT1_M  MDFYYLPGSAPCRSVLMTAKALGIELNKKLLNLQAGEHLKPEFLKINPQHTIPTLVDGD
                10        20        30        40        50
```

B.

```
>>GTT1_MUSDO GLUTATHIONE S-TRANSFERASE 1 (EC 2.5.1.18) (CLASS-THETA).    (208 aa)
 initn: 1229 init1: 1229 opt: 1230 Z-score: 1615.1 expect() 2.6e-83
Smith-Waterman score: 1230;  85.024% identity in 207 aa overlap

               10        20        30        40        50        60        70
gi|121 MVDFYYLPGSSPCRSVIMTAKAVGVELNKKLLNLQAGEHLKPEFLKINPQHTIPTLVDNGFALWESRAIQVYLVE
       x        x     x     x x                                   xX        x
GTT1_M  MDFYYLPGSAPCRSVLMTAKALGIELNKKLLNLQAGEHLKPEFLKINPQHTIPTLVDGDFALWESRAIMVYLVE
                10        20        30        40        50        60        70
```

Fig. 4: Alternative output formats—Alignments of `gtt1_drome` with `gtt1_musdo` are shown using the default (A) program parameters and (B) the command line options:

```
-f –14 -w 75 –L –m 1
```

(see text for details).

Table 16: Alignment options

`-m 0`	Highlight identical aligned residues with ":", conservative replacements with "."
`-m 1`	Identities are not highlighted. Highlight conservative replacements with "x", non-conservative replacements with "X".
`-m 2`	Highlight identities with ".", non-identical residues with the residue.
`-m 3`	The alignments are printed as two fasta format sequence entries with "-" indicating gaps. These files are sometimes useful as input to other programs.
`-m 4`	Do not show an alignment; show a graph (------) of where the aligned region maps onto the query sequence. Useful for highlighting different domains in proteins.
`-m 5`	A combination of -m 0 and -m 4 that shows both the mapping and the alignment.
`-m 6`	Similar to -m 5, but includes html commands for a WWW browser like Netscape or Internet Explorer and links to simplify looking up the library sequence and re-searching the database.
`-m 10`	Parseable output designed to be read by other computer programs. Each alignment is a series of labeled tags that specify the beginning, end, score, search parameters, and other information.

5. BEYOND SEQUENCE HOMLOGY—IDENTIFYING NEW PARALOGS

The use of the FASTA and BLAST programs for identifying distantly related sequences has been extensively reviewed (3-5), so in this last section we will consider a slightly different problem that exploits the flexibility of the FASTA programs and the high quality of their alignments.

Here, we seek to identify new paralogs of known human or mouse families from EST databases. For example, two human prostaglandin synthase enzymes are known, COX1 (`pgh1_human`) and COX2 (`pgh2_human`), in humans, mice, rats, and other mammals. Prostaglandin synthases are targets of non-steroidal anti-inflamatory drugs, including aspirin and ibuprofen. Thus, there is great interest in finding additional members of this family and it is certainly possible that additional prostaglandin synthases have been sequenced, either by large scale EST sequencing or by genomic sequencing.

5.1 Overall strategy

Paralogs are members of a gene family (and are thus related or homologous) that differ from other sequences in the family because of gene duplication events. (Orthologous genes differ because they are found in different species.) A search of the SwissProt database (Table 17) shows the two prostaglandin synthase (PGH) subfamilies, but also shows distantly related peroxidases. The human PGH1 and PGH2 isoenzymes share about 65% sequence identity (E() $< 10^{-165}$). (In contrast, orthologous human and mouse PGH1 sequences share 89.3% identity.) We expect a new human PGH synthase to share very strong similarity to PGH1 and PGH2 (E() $<10^{-20}$) but to share less than 80% identity to either PGH1 or PGH2. Since we will be scanning EST databases to find the new paralogs, we expect that sequences with > 90-95% identity are probably from mRNAs for known proteins that have sequencing errors, but that sequences that are 50-90% identical are candidate paralogs.

Table 17: Prostaglandin synthase search results

The best scores are:		len	E(74357)
PGH1_HUMAN	prostaglandin G/H synthase 1	599	3.9e-264
PGH1_SHEEP	prostaglandin G/H synthase 1	600	2.3e-244
PGH1_MOUSE	prostaglandin G/H synthase 1	602	9.5e-237
PGH2_CHICK	prostaglandin G/H synthase 2	603	1.2e-168
PGH2_HUMAN	prostaglandin G/H synthase 2	604	1.9e-165
PGH2_MOUSE	prostaglandin G/H synthase 2	604	2.4e-164
PGH2_CAVPO	prostaglandin G/H synthase 2	604	1.7e-163
PGH2_RAT	prostaglandin G/H synthase 2	604	1.4e-162
PERM_MOUSE	myeloperoxidase prec.	718	0.0001
PERO_DROME	peroxidase prec.	690	0.00024
PERT_HUMAN	thyroid peroxidase prec.	933	0.0003
PERM_HUMAN	myeloperoxidase prec.	745	0.00034
PERT_PIG	thyroid peroxidase prec.	926	0.0029
PERL_BOVIN	lactoperoxidase prec.	712	0.016
PERT_MOUSE	thyroid peroxidase prec.	914	0.02
PERL_HUMAN	lactoperoxidase LPO	324	0.027
PERT_RAT	thyroid peroxidase prec.	914	0.089
FBP1_STRPU	fibropellin I prec.	1064	0.16

PGCN_RAT	neurocan core prot. prec.	1257	0.21
FBP3_STRPU	fibropellin C prec.	570	0.31
PGCN_MOUSE	neurocan core prot. prec.	1268	0.33
PERE_MOUSE	eosinophil peroxidase prec.	716	0.51
NOTC_DROME	neurogenic locus notch prot.	2703	0.74
DLK_MOUSE	delta-like prot. prec.	385	0.86
PERE_HUMAN	eosinophil peroxidase prec.	715	0.92
NTC1_MOUSE	neurogenic locus notch homolog	2531	0.94

Results of a `fasta3` (*ktup=2*) search with `pgh1_human` against the Swissprot protein sequence database.

To identify new `pgh1_human` paralogs, we will search the human EST database (obtained from ftp://ncbi.nlm.nih.gov/blast/db/) with the `pgh1_human` and `pgh2_human` protein sequences using the `tfasty3` program. `tfasty3` is used because: (1) we wish to compare a protein query to a DNA (EST) database; and (2) we will use both the expectation value E() and the percent identity to characterize matches, so a high-quality protein:DNA alignment is required (`tfastx3` is faster but produces a lower quality alignment, ref. 15). We will then examine the EST sequences that share significant similarity and categorize them as orthologous to `pgh1_human`, `pgh2_human`, or a new paralog.

5.2 Statistical significance and percent identity

While our goal is to identify sequences that are similar to, but not identical with, known prostaglandin synthases, conventional similarity criteria (E()-value and percent identity) do not fully capture the information we seek. As the results of the `pgh1_human` and `pgh2_human` `tfasty3` searches demonstrate (Table 18), EST sequences that share higher sequence identity do not necessarily have better E()-values.

The discrepancy between E()-value and percent identity reflects the dependence of E()-value on alignment length. EST sequences tend to be partial, so that an orthologous 100% match to the C-terminal 30 amino acids in `gb|N79146` can have a worse expectation value (2.9×10^{-6}) than a 59% identity to a paralogous gene ($E() < 6.7 \times 10^{-19}$). However, percent identity is a poor criterion for similarity, because unrelated sequences (e.g. `gb|AA485017`) can share high identity (66.1% over 62 codons) that does not produce a statistically significant similarity score. Nevertheless, for sequences that share significant similarity, percent identity is a useful measure of sequence difference. Thus, among the statistically significant matches in Table 18, orthologous matches always had percent identities > 90%, with one possible exception (`gb|AA223896`, see below).

Table 18: Prostaglandin synthase ESTs

pgh1_human:		len	[f/r]	opt	E(10^6)	%ident.	I/II
gb|R96180	Pineal_gland_N3HPG	355	[f]	654	3e-38	98.0	I
gb|AA296431	Umbilical vein endothelial	279	[f]	380	6.7e-19	59.1	II
gb|T29235	Human Bone	257	[f]	358	2.2e-17	63.3	II
gb|AA037294	Senescent_fibroblasts_NbHSF	471	[f]	304	3.1e-13	98.0	I
gb|AI022012	Senescent_fibroblasts_NbHSF	537	[r]	248	3.5e-09	64.5	II
gb|N79146	Multiple_sclerosis_2NbHMSP	544	[f]	207	2.9e-06	100.0	I
gb|AA223896	NT2 neuronal precursor	97	[f]	185	1.3e-05	80.0	??

gb|AA485017	NCI_CGAP_GCB1	208	[f]	124	0.72	66.1	
pgh2_human:		len	[f/r]	opt	E(10^6)	%ident.	I/II
gb|AA296431	Umbilical vein endothelial	279	[f]	574	1.4e-35	96.8	II
gb|T29235	Human Bone	257	[f]	536	1e-32	92.9	II
gb|AI022012	Senescent_fibroblasts_NbHSF	537	[r]	541	1.1e-32	95.8	II
gb|R96180	Pineal_gland_N3HPG	355	[f]	410	6.3e-23	65.8	I
gb|AA223896	NT2 neuronal precursor	97	[f]	136	0.01	50.0	??
gb|AA885610	NCI_CGAP_Lu5	320	[f]	141	0.018	46.3	
gb|AA911293	NCI_CGAP_Lu5	172	[f]	131	0.049	43.6	

Results from searches with pgh1_human and pgh2_human against the BLAST est_human database using tfasty3 and with the default BLOSUM50 scoring matrix. pgh1 (COXI) or pgh2 (COXII) orthologs are labeled in the right column.

5.3 Shifting evolutionary horizons with scoring matrices

Examination of the high scoring ESTs found with pgh1_human and pgh2_human in Table 18 suggests that all but one of the ESTs share > 90% identity with either pgh1_human or pgh2_human. The exception, gb|AA223896, shares only 80% identity with pgh1_human and 50% identity with pgh2_human, and thus is a candidate novel paralog prostaglandin synthase.

However, the gb|AA223896 EST sequence is very short (97 nucleotides), and there are only 6 mismatches, half of which are within 20 nucleotides of one end of the sequence. Thus, we must consider whether this is truly a novel paralog, or simply a short, poor-quality sequence of a pgh1_human mRNA that has several errors at one end (as is expected with high-throughput EST sequencing). While the end-sequence error problem could be reduced by ad hoc changes to the alignment code that down-weighted end-mismatches, a simpler approach is to use shallower scoring matrices.

Table 19: Searching with “shallow” scoring matrices

pgh1_human:	len	E(BL50)	%	E(M20)	%	E(M10)	%	I/II
gb|R96180	355	3e-38	98.0	2.3e-72	99.0	6.5e-75	100.0	I
gb|AA296431	279	6.7e-19	59.1	6.8e-25	61.3	1.3e-22	62.4	II
gb|T29235	257	2.2e-17	63.3	5.3e-22	64.8	2.6e-18	66.2	II
gb|AA037294	471	3.1e-13	98.0	3e-30	98.0	3.3e-31	97.8	I
gb|AI022012	537	3.5e-09	64.5	1.2e-15	58.8	3.4e-13	60.8	II
gb|N79146	544	2.9e-06	100.0	2.6e-16	100.0	3.0e-17	100.0	I
gb|AA223896	97	1.3e-05	80.0	8.4e-13	87.1	2.8e-12	87.1	??
gb|AA485017	208	0.72	66.1	4.8e-14	84.7	4.1e-14	88.9	??
pgh2_human:								
gb|AA296431	279	1.4e-35	96.8	2.2e-69	96.8	8.0e-72	98.9	II
gb|T29235	257	1e-32	92.9	2.9e-61	94.1	9.1e-63	95.2	II
gb|AI022012	537	1.1e-32	95.8	1.6e-68	96.0	1.1e-70	97.0	II
gb|R96180	355	6.3e-23	65.8	1.0e-30	56.9	9.1e-27	60.3	I
gb|AA485017	208	__a	—	2.4e-05	75.6	3.3e-4	79.1	??
gb|AA223896	97	0.01	50.0	0.01	69.0	0.2	79.2	??
gb|AA885610	320	0.018	46.3	—	—	—	—	
gb|AA911293	172	0.049	43.6	—	—	—	—	

[a]E() values indicated as — were >5.0.

Additional searches with very shallow scoring matrices (MDM20 and MDM10, ref. 19; Table 19) show slightly different, potentially more interesting perspectives. When shallower scoring matrices are used, both orthologous and paralogous alignments become mores statistically significant, and, as expected, the percent identities increase ("shallower" scoring matrices give more positive scores to identities and more negative scores to non-conservative replacements). Of greater interest are two sequences `gb|AA223896` and `gb|AA485017`, which show significant similarity with `pgh1_human` with MDM20 and MDM10. Both sequences are tantalizing candidates for new paralogs (as orthologs consistently have percent identities higher than 90% with MDM20. However, the alignments of both sequences show a large number of frameshifts (which do not affect the percent identity calculation), suggesting that these sequences may have percent identities < 90% because of a poor quality sequence, rather than a novel gene.

The last two entries (`gb|AA885610` and `gb|AA911293`) in the `pgh2_human` search shows that shallow scoring matrices can also be used to quickly rule out high scoring unrelated sequences. The expectation values for those two sequences, which were marginally significant (0.018 and 0.049) scores with BLOSUM50 and were not significantly similar to `pgh1_human`, became very high (E() > 5) when MDM20 and MDM10 were used. Thus, shallower scoring matrices can be used to provide a more stringent test for sequence similarity when near-identity is expected for at least one of the query sequences.[4]

6. SUMMARY

The FASTA3 and FASTA2 packages provide a flexible set of sequence comparison programs that are particularly valuable because of their accurate statistical estimates and high-quality alignments. Traditionally, sequence similarity searches have sought to ask one question: "Is my query sequence homologous to anything in the database?" Both FASTA and BLAST can provide reliable answers to this question with their statistical estimates; if the expectation value E() is < 0.001-0.01 and you aren't doing hundreds of searches a day, the answer is probably yes.

In general, the most effective search strategies follow these rules:

1. Whenever possible, compare at the amino acid level, rather than the nucleotide level. Search first with protein sequences (`blastp`, `fasta3`, and `ssearch3`), then with translated DNA sequences (`fastx`, `blastx`), and only at the DNA level as a last resort (Table 5).
2. Search the smallest database that is likely to contain the sequence of interest (but it must contain many unrelated sequences for accurate statistical estimates).
3. Use sequence statistics, rather than percent identity or percent similarity, as your primary criterion for sequence homology.

---

[4]While MDM20 and MDM10 can serve to provide more stringent alignments, they are not the best matrices, because they were built assuming an evolutionary model. More accurate matrices could be derived from looking at large numbers of EST sequencing errors, and building a matrix that was based on a sequencing error model, rather than evolutionary divergence.

4. Check that the statistics are likely to be accurate by looking for the highest scoring unrelated sequence, using prss3 to confirm the expectation, and searching with shuffled copies of the query sequence (randseq, searches with shuffled sequences should have E() ~1.0).
5. Consider searches with different gap penalties and other scoring matrices. Searches with long query sequences against full-length sequence libraries will not change dramatically when BLOSUM62 is used instead of BLOSUM50 (20), or a gap penalty of -14/-2 is used in place of -12/-2. However, shallower or more stringent scoring matrices are more effective at uncovering relationships in partial sequences.(3, 18), and they can be used to sharpen dramatically the scope of the similarity search.

However, as illustrated in the last section, the E() value is only the first step in characterizing a sequence relationship. Once one has confidence that the sequences are homologous, one should look at the sequence alignments and percent identities, particularly when searching with lower quality sequences. When sequence alignments are very short, the alignment should become more significant when a shallower scoring matrix is used, e.g. BLOSUM62 rather than BLOSUM50 (remember to change the gap penalties).

Homology can be reliably inferred from statistically significant similarity. While homology implies common three-dimensional structure, homology need not imply common function. Orthologous sequences usually have similar functions, but paralogous sequences often acquire very different functional roles. Motif databases, such as PROSITE (21), can provide evidence for the conservation of critical functional residues. However, motif identity in the absence of overall sequence similarity is not a reliable indicator of homology.

Acknowledgements

W.R.P. is supported by a grant from the National Library of Medicine (LM04961).

REFERENCES

1. 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.
2. Bult, C. J., White, O., Olsen, G. J., Zhou, L., Fleischmann, R. D., Sulton, G. G., Blake J. A., Fitzgerald, L. M., Clayton, R. A., Gocayne, J. D., Kerlavage, A. R., Dougherty, B. A., Tomb, J.-F., Adams, M. D., Reisch, C. I., Overbeek, R., Kirkness, E. F., Weinstock, K. G., Merrick, J. M., Glodek, A., Scott, J. L., Geoghagen, N. S. M., Weidman, J. F., Fuhrmann, J. L., Nguyen, D., Utterback, T. R., Kelley, J. M., Peterson, J. D., Sadow, P. W., Hanna, M. C., Cotton, M. D., Roberts, K. M., Hurst, M. A., Kaine, B. P., Borodovsky, M., Klenk, H.-P., Fraser, C. M., Smith, H. O., Woese, C. R., and Venter, J. C. (1996) Complete genome sequence of the methanogenic archaeon, methanococcus jannaschii *Science* **273**, 1058-1073.
3. Altschul, S. F., Boguski, M. S., Gish, W., and Wootton, J. C. (1994) Issues in searching molecular sequence databases *Nature Genet.* **6**, 119-129.
4. Pearson, W. R. (1996) Effective protein sequence comparison *Methods Enzymol.* **266**, 227-258.
5. Pearson, W. R. (1997) Identifying distantly related protein sequences *Comput. Appl. Biosci.* **13**, 325-332.

6. Pearson, W. R. (1998) Empirical statistical estimates for sequence similarity searches *J. Mol. Biol.* **276**, 71-84.
7. Pearson, W. R. and Lipman, D. J. (1988) Improved tools for biological sequence comparison *Proc. Natl. Acad. Sci. USA* **85**, 2444-2448.
8. Lipman, D. J. and Pearson, W. R. (1985) Rapid and sensitive protein similarity searches *Science* **227**, 1435-1441.
9. Bleasby, A. J., Akrigg, D., and Attwood, T. K. (1994) Owl-a non-redundant composite protein sequence database. *Nucleic Acids Res.* **22**, 3574-3577.
10. Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990) A basic local alignment search tool *J. Mol. Biol.* **215**, 403-410.
11. Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997) Gapped blast and psi-blast: a new generation of protein database search programs *Nucleic Acids Res.* **25**, 3389-3402.
12. Arratia, R., Gordon, L., and Waterman, M. S. (1986) An extreme value theory for sequence matching *Ann. Stat.* **14**, 971-993.
13. Karlin, S. and Altschul, S. F. (1990) Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes *Proc. Natl. Acad. Sci. USA* **87**, 2264-2268.
14. Wootton, J. C. and Federhen, S. (1993) Statistics of local complexity in amino acid sequences and sequence databases *Comput. Chem.* **17**, 149-163.
15. Pearson, W. R., Wood, T., Zhang, Z., and Miller, W. (1997) Comparison of DNA sequences with protein sequences *Genomics* **46**, 24-36.
16. Henikoff, S. and Henikoff, J. G. (1992) Amino acid substitutions matrices from protein blocks *Proc. Natl. Acad. Sci. USA* **89**, 10915-10919.
17. Schwartz, R. M. and Dayhoff, M. (1978) Matrices for detecting distant relationships In M. Dayhoff, (ed.), Atlas of Protein Sequence and Structure, volume 5, supplement 3, pp. 353-358 National Biomedical Research Foundation Silver Spring, MD.
18. Altschul, S. F. (1991) Amino acid substitution matrices from an information theoretic perspective *J. Mol. Biol.* **219**, 555-565.
19. Jones, D. T., Taylor, W. R., and Thornton, J. M. (1992) The rapid generation of mutation data matrices from protein sequences *Comp. Appl. Biosci.* **8**, 275-282.
20. Pearson, W. R. (1995) Comparison of methods for searching protein sequence databases *Protein Sci.* **4**, 1145-1160.
21. Bairoch, A. (1991) PROSITE: a dictionary of sites and patterns in proteins *Nucleic Acids Res.* **19**, supplement, 2241-2245.
22. Smith, T. F. and Waterman, M. S. (1981) Identification of common molecular subsequences *J. Mol. Biol.* **147**, 195-197.
23. Huang, X. and Miller, W. (1991) A time-efficient, linear-space local similarity algorithm *Adv. Appl. Math.* **12**, 337-357.
24. Waterman, M. S. and Eggert, M. (1987) A new algorithm for best subsequences alignment with application to tRNA-rRNA comparisons *J. Mol. Biol.* **197**, 723-728.
25. Myers, E. W. and Miller, W. (1988) Optimal alignments in linear space *Comp. Appl. Biosci.* **4**, 11-17.
26. Kyte, J. and Doolittle, R. F. (1982) A simple method for displaying the hydropathic character of a protein *J. Mol. Biol.* **157**, 105-132.

27. Barker, W. C., Garavelli, J. S., Haft, D. H., Hunt, L. T., Marzec, C. R., Orcutt, B. C., Srinivasarao, G. Y., Yeh, L. S. L., Ledley, R. S., Mewes, H. W., Pfeiffer, F., and Tsugita, A. (1998) The PIR-International protein sequence database *Nucleic Acids Res.* **26**, 27-32.
28. Chao, K.-M., Pearson, W. R., and Miller, W. (1992) Aligning two sequences within a specified diagonal band *Comp. Appl. Biosci.* **8**, 481-487.
29. Altschul, S. F. and Gish, W. (1996) Local alignment statistics *Methods Enzymol.* **266**, 460-480.