Published in final edited form as:

Curr Protoc Bioinformatics. ; 53: 3.9.1–3.925. doi:10.1002/0471250953.bi0309s53.

Finding Protein and Nucleotide Similarities with FASTA

William R. Pearson

University of Virginia School of Medicine, Charlottesville, Virginia

William R. Pearson: wrp@virginia.edu

Abstract

The FASTA programs provide a comprehensive set of rapid similarity searching tools (*fasta36*, *fastx36*, *tfastx36*, *fasty36*, *tfasty36*), similar to those provided by the BLAST package, as well as programs for slower, optimal, local and global similarity searches (*ssearch36*, *ggsearch36*) and for searching with short peptides and oligonucleotides (*fasts36*, *fastm36*). The FASTA programs use an empirical strategy for estimating statistical significance that accommodates a range of similarity scoring matrices and gap penalties, improving alignment boundary accuracy and search sensitivity (Unit 3.5). The FASTA programs can produce “BLAST-like” alignment and tabular output, for ease of integration into existing analysis pipelines, and can search small, representative databases, and then report results for a larger set of sequences, using links from the smaller dataset. The FASTA programs work with a wide variety of database formats, including MySQL and PostgreSQL databases (Unit 9.4). The programs also provide a strategy for integrating domain and active site annotations into alignments and highlighting the mutational state of functionally critical residues. These protocols describe how to use the FASTA programs to characterize protein and DNA sequences, using protein:protein, protein:DNA, and DNA:DNA comparisons.

Keywords

Similarity; homology; expectation; E()-value; alignment annotation; scoring matrices

INTRODUCTION

Similarity searching is one of the most powerful strategies for characterizing newly determined sequences. BLAST, HMMER, and the programs in the FASTA package routinely identify homologous sequences that diverged more than a billion years ago. The FASTA software package provides a comprehensive set of programs (Table 3.9.1) for protein and DNA sequence comparison. While the FASTA programs are not as fast as the BLAST programs (UNITS 3.3 & 3.4), they can be equally sensitive, and, because they calculate statistical parameters from the distribution of similarity scores calculated during the search, they can provide more accurate statistical estimates using a wide range of scoring parameters. Programs in the FASTA package offer a broad range of speed, sensitivity, and alignment and statistical accuracy for similarity searches and statistical analysis. FASTA can be run the command line (see Basic Protocol 1) with options to customize the scoring matrix, gap penalty and output format. For large-scale analyses, scripted alignment

annotation (Basic Protocol 2) provides a powerful facility for integrating functional information from other resources.

STRATEGIC PLANNING

In planning a FASTA or BLAST search—choosing a program, a database, and the search parameters—it is important to remember the central goal of a sequence similarity search: identifying homologous sequences (Unit 3.1). Homologous sequences share a common ancestor, have similar three-dimensional structures, and often (but not always) have similar functions. When two sequences share statistically significant similarity, i.e., much more similarity than would be expected by chance, we infer that they are homologous. Similarity searches are most sensitive when: (1) protein or translated protein sequences are compared, and (2) small, comprehensive databases are searched (Unit 3.1). Most of the protocols described below can also be used for DNA similarity searching, but we focus on protein sequence examples because alignment to proteins, or translated DNA, is dramatically more sensitive (Unit 3.1).

BASIC PROTOCOL 1 — USING THE FASTA PROGRAMS

While many researchers run the FASTA program interactively through web servers, like the similarity searching tools at the European Bioinformatics Institute (<http://www.ebi.ac.uk/Tools/sss>), the FASTA programs can be downloaded (see Support Protocol 1) and run from a Unix, MacOS terminal, or Windows “console”. The FASTA programs can be run in the older “interactive” mode as well, but the “command line” mode is more convenient, because one need not wait for the results of the search, and it is possible to write scripts that perform and analyze the results from large numbers of searches. In this example, we compare an uncharacterized protein from *E. histolytica*, a protozoan that causes amoebic dysentery, to the UniProt human reference proteome, demonstrating the power of protein similarity searching (humans and *E. histolytica* last shared a common ancestor more than a billion years ago) and allowing us to examine results from a smaller set of proteins to show how statistical estimates can be validated (see Guidelines for Understanding Results).

Necessary Resources

Hardware—A modern Windows (32-bit, 64-bit), or Mac OSX (32-bit, 64-bit), or Unix/Linux computer with at least 50 MB of free disk space for the programs and 100 GB of disk space for protein sequence databases. The FASTA programs require very little memory over that required by the computer’s operating system.

Software—The FASTA programs, installed and configured as described in Support Protocol 1.

Files—Appropriate sequence databases, downloaded from the NCBI (<ftp://ftp.ncbi.nih.gov/pub/blast>) or EBI (<ftp://ftp.ebi.ac.uk/databases>), as described in Support Protocol 2.

A query protein sequence in FASTA format (APPENDIX 1B); this example uses the UniProt sequence C4M1E7_ENTHI, an uncharacterized protein from *E. histolytica*. In the examples below, the ‘\$’ character indicates a shell or command line prompt, and should not be typed. C4M1E7_ENTHI can be downloaded with the command:

```
$ curl http://www.uniprot.org/uniprot/C4M1E7.fasta > c4m1e7.fa
```

1. After downloading and installing the FASTA programs (see Support Protocol 1), run the program by typing the command (do not type the ‘\$’):

```
$ fasta36 -h
```

The following response is returned:

```
USAGE
  fasta36 [-options] query_file library_file [ktup]
  fasta36 -help for a complete option list
DESCRIPTION
  FASTA searches a protein or DNA sequence data bank
  version: 36.3.8 Jul, 2015
COMMON OPTIONS (options must precede query_file library_file)
  -s: [BL50] scoring matrix;
  -f: [-10] gap-open penalty;
  -g: [-2] gap-extension penalty;
  -S filter lowercase (seg) residues;
  -b: high scores reported (limited by -E by default);
  -d: number of alignments shown (limited by -E by default);
  -I interactive mode;
```

2. To run the program (non-interactively, the default), you must specify both a query sequence (c4m1e7.fa) and a library file (/genomes/up_human.lseg). In this example (from Linux), the alignments with *E*-values (expectation values) <2.0 are saved to the file c4_v_hum.k2.

```
$ fasta36 -s BP62 -S -E 2.0 c4m1e7.fa /genomes/up_human.lseg >
c4_v_hum.k2
```

The general form of a FASTA command-line run is:

```
fasta36 -option1 -option2 -option3 query library ktup
```

Every FASTA command-line search will include the query and library information. The *ktup* argument is often left out, and is not used by the optimal *ssearch36*, *ggsearch36*, and *glsearch36* programs. Other options are discussed in Critical Parameters.

The standard *fasta* command line differs from a *blast* command in two ways. First, the FASTA programs have two (or three) positional command line arguments, the query sequence filename, the library sequence file name, and, the *ktup* value (for the *fasta36*, *fastx36*, *fasty36*, *tfastx36* and *tfasty36* programs, the *ktup* is the number of identities that must be matched to begin identifying a similar region). Second, unlike BLAST, FASTA command-line options *must* precede the query and library file names.

On Unix, MacOSX, and Windows systems, the results of a FASTA search can be saved to a file with “output redirection” (including `> results.file` at the end of the command). Output redirection to a file is not part of the FASTA program, but can be used with any Unix command-line program and at the Windows console.

3. Examine the results (see Guidelines for Understanding Results). Figure 3.9.1 shows three parts of the results file (`c4_v_hum.k2`). The first part (A) shows a summary of the search—the query sequence, the sequence library, the scoring matrix and gap parameters, and the statistical estimates, which are recorded so that the search can be reproduced at a later time. The summary of high scoring sequences (B) illustrates the ability of protein sequence similarity searching to identify homologs in proteins that diverged more than billion years ago. In this example, the E()-value (BLAST expect) provides the most reliable evidence of homology based on excess similarity. There are three α/β -hydrolase homologs with very significant similarity ($E() < 10^{-16}$), and two more with weaker, but still significant, similarity. Even in a search with millions of query sequences, the top three homologs would be clearly significant, despite the fact that they share less than 30% amino-acid sequence identity. The alignment of the *E. histolytica* protein to ABHD1_HUMAN (C) differs slightly from BLAST alignment output formatting (BLAST alignment output is available with the ‘`-mBB`’ option). In addition to using different symbols to highlight identities and similarities, the default FASTA output provides sequence context outside the optimal locally aligned region. For a more complete discussion of the output, see Guidelines for Understanding Results.

SUPPORT PROTOCOL 1 — DOWNLOADING AND INSTALLING THE FASTA PROGRAMS

Versions of the FASTA programs are available on multiple web servers. A search for “fasta similarity search” provides links to many resources. The European Bioinformatics Institute (EBI) provides access to all the FASTA programs together with comprehensive databases: <http://www.ebi.ac.uk/Tools/sss> both as interactive web pages and as programmable web services (see Unit 3.12). Web-based FASTA resources provide up-to-date versions of protein and DNA sequence databases. The EBI provides the UniProt protein databases and ENA (European Nucleotide Archive, formerly EMBL-Bank) DNA sequences. However, to search local DNA or protein resources, to use customized scripts for alignment annotation, or to access the full range of alignment formats, the FASTA programs must be installed locally on your computer. Users who already have access to a local copy of the FASTA programs at their institution should skip to Basic Protocol 1.

Necessary Resources

Hardware—A modern Windows (32-bit, 64-bit), or Mac OSX (32-bit, 64-bit), or Unix/Linux computer with at least 50 MB of free disk space for the programs and 100 GB of disk space for protein sequence databases. The FASTA programs require very little memory over that required by the computer’s operating system.

Software—The latest FASTA program distributions for Windows, Mac OSX, and Unix/Linux are available from several sources: (1) Github: <http://github.com/wrpearson/fasta36>, (2) the author’s software repository: <http://faculty.virginia.edu/wrpearson/fasta>, and (3) the EBI’s software repository: <ftp://ftp.ebi.ac.uk/pub/software/unix/fasta>.

Unix/Linux/MacOSX versions of the programs are provided as compressed “tar files,” e.g., `fasta36.tar.gz`. Be sure to transfer the `fasta36.tar.gz` file in binary format. Windows versions of the programs are available as compressed .zip files (`fasta36-win32.zip`) in the CURRENT directory. The programs come with complete source code and executable binaries for 64/32-bit Linux, 64/32-bit Windows and Mac OSX machines. The FASTA distribution file should be copied to a new directory for installation.

Files—To verify that the program is installed correctly, this protocol uses the `mgstml.aa` and `prot_test.lseg` files included in the FASTA distribution file.

1. Unpack files and compile if necessary
 - a. For Unix/Linux/MacOSX — Download and unpack the FASTA distribution file using shell commands:

```
$ curl -O \
\ http://faculty.virginia.edu/wrpearson/fasta/
executables/fasta36-macosxuniv.tar.gz
$ tar zxvf fasta36-macosxuniv.tar.gz
```

The `curl` command ends with a `'\'` to indicate that the command is continued onto a second line. This file contains the Mac OSX executable binaries. Linux-32/64 bit users should download the `-linux32` or `-linux64` files.

- b.** Windows — Use a web browser to download <http://faculty.virginia.edu/wrpearson/fasta/executables/fasta36-win32.zip> and then unpack the file in your `Documents` directory.

2. Install the programs.

On Windows and Macintosh computers, it is generally easiest to run the programs from the directory (folder) where they were unpacked. On Unix/Linux computers, particularly when several people use the computer, one usually copies the FASTA programs to a common directory for programs, e.g., `/usr/local/bin/` or perhaps `/seqprg/bin/`. This directory should be in the executable search path.

- a.** For Unix/Linux: Once the programs have been unpacked into the `fasta36` directory, test them by typing (from a Unix/Linux shell prompt):

```
$ bin/fasta36 -help
$ bin/fasta36 seq/mgstml.aa seq/prot_test.lseg >
mgstml_test.out
$ more mgstml_test.out
```

This is an example of running the program from the command line, with `mgstml.aa` as the query sequence and `prot_test.lseg` as the database, saving the resulting output to a file (`mgstml_test.out`). The `mgstml.aa` and `prot_test.lseg` files are included in the FASTA distribution file in the `seq/` directory. Every FASTA command-line search must include the query and library information.

- b.** For Windows:
 - i.** First open a DOS “console” window
 - ii.** `cd` to the directory where you installed the FASTA package
 - iii.** Then type the commands:

```
$ bin\fasta36 -help
$ bin\fasta36 seq\mgstml.aa seq
\prot_test.lseg > mgstml_test.out
```

SUPPORT PROTOCOL 2: DOWNLOADING AND PREPARING SEQUENCE DATABASES

The FASTA programs currently work with eight different “flatfile” sequence database formats, NCBI BLAST binary formats, and a MySQL SQL query format (UNIT 9.2). The default database format is FASTA format (APPENDIX 1B). FASTA format sequence databases can be downloaded from the National Center for Biotechnology Information (<ftp://ftp.ncbi.nih.gov/blast/db/>) and the European Bioinformatics Institute (<ftp://ftp.ebi.ac.uk/pub/databases/>). For example on Linux/Unix/MacOS machines, the Swiss-Prot database can be downloaded in FASTA format with:

```
$ curl -O ftp://ftp.ncbi.nih.gov/blast/db/FASTA/swissprot.gz
$ gunzip swissprot.gz
```

Alternatively, the database can be downloaded from:

```
$ curl -O \
\ ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/
uniprot_sprot.fasta.gz
$ gunzip uniprot_sprot.gz
```

(Again, the `curl` command above has been continued on to a second line by ending the first line with a ‘\’.)

Sequence database formats

Because many sequence databases are quite large, most researchers prefer not to keep multiple copies of a database. The current version of FASTA supports the BLAST formatdb and makeblastdb databases, the default format used by BLAST.

Removing low-complexity regions with PSEG

While FASTA statistical estimates are, in general, very accurate, they can be confused by query sequences that contain runs with reduced amino acid complexity, for example, proline-rich regions. The SEG and PSEG programs (Wootton and Federhen, 1993) can be used to remove these low-complexity regions, and the PSEG program can be used to convert the low-complexity regions to lowercase, so that, with the FASTA -S option (see Critical Parameters), they are ignored during the initial similarity scan. The pseg program can be downloaded from <ftp://ftp.ncbi.nih.gov/pub/seg/pseg>. Copy all the program source files into a new pseg directory, compile the program with make, and move it to the

program directory (/seqprg/bin). Finally, convert low-complexity regions to lowercase with the command:

```
$ /seqprg/bin/pseg ./swissprot -z 1 -q > swissprot.pseg
```

Here, ./swissprot is the original database file that was downloaded and uncompressed and “ -z 1 -q” indicates that the results should be written in FASTA format, with lowercase letters for low-complexity regions, to the file swissprot.lseg.

BASIC PROTOCOL 2—LARGE-SCALE SEQUENCE ANALYSIS WITH ALIGNMENT ANNOTATION

In addition to basic similarity searching and alignment display, the FASTA programs offer a flexible option for integrating functional site and domain information into the alignment summary: scripted alignment annotation. With alignment annotation, the FASTA programs run a script that retrieves active site and variant site information, as well as domain or exon boundaries. For example, Figure 3.9.2 shows the integration of UniProt (UniProt Consortium, 2015) variation and active site annotations together with Pfam (Finn et al., 2014) domain annotations.

While Figures 3.9.1B and C present convincing biological evidence that the putative uncharacterized protein from *E. histolytica* is homologous to the human α/β -hydrolase—very significant sequence similarity and a sequence alignment that includes almost the entire protein—much more is known about α/β -hydrolase that can be used to support the inference of homology and *functional* similarity. Information on the functional residues in the human protein can be mapped to the *E. histolytica* protein by sequence alignment, revealing that the *E. histolytica* protein has identical functional residues, and a full-length Pfam AB-hydrolase domain (Figure 3.9.2A)—strong support for inference of α/β -hydrolase activity in *E. histolytica*. While the information in Figure 3.9.2A is readily available from UniProt and Pfam, manually scanning and mapping sequence coordinates for tens of thousands of predicted proteins from a newly sequenced genome is impractical. To simplify large-scale sequence annotation, the FASTA programs offer a compact output, similar to BLAST-tabular output, that provides the same similarity and alignment information, but also provides CIGAR encoded alignment information and encoded annotation information.

Necessary Resources

Hardware—A modern Windows (32-bit, 64-bit), or Mac OSX (32-bit, 64-bit), or Unix/Linux computer with at least of free disk space for the programs and 100 GB of disk space for protein sequence databases.

Software—The FASTA programs, installed and configured as described in Support Protocol 1. Included with the FASTA distribution is the program `scripts/ann_upfeats_pfam_www.pl`, which extracts annotation information from the UniProt/EBI and Pfam web servers (See Support protocol 1).

Files—Appropriate sequence databases, downloaded from the NCBI (<ftp://ftp.ncbi.nih.gov/pub/blast>), EBI (<ftp://ftp.ebi.ac.uk/databases>), as described in Support Protocol 2. For the search below, the UniProt Reference human proteome was downloaded using the commands:

```
$ curl -O ftp://ftp.ebi.ac.uk/pub/databases/uniprot/current_release/\
knowledgebase/reference_proteomes/Eukaryota/UP000005640_9606.fasta.gz
$ gunzip -c UP000005640_9606.fasta.gz > up_human.fa
$ pseg ./up_human.fa -z 1 -q > up_human.lseg
```

Again, the `curl` command is shown on two lines; this is possible because of the ‘\’ at the end of the line.

A query protein sequence in FASTA format (APPENDIX 1B); this example uses the UniProt sequence C4M1E7_ENTHI, a putative uncharacterized protein from *E. histolytica*, which can be downloaded with the command:

```
$ curl http://www.uniprot.org/uniprot/C4M1E7.fasta > c4m1e7_enthi.fa
```

1. After downloading and installing the FASTA programs and scripts (See Support Protocol 1), run the program by typing the command:

```
$ fasta36 -s BP62 -m 8CC -E 1e-3 -V \!scripts/
ann_upfeats_pfam_www.pl\
c4m1e7_enthi.fa /slib/up_human.lseg > c4m1e7_v_human.k2_tab
```

This command sends the results to the `c4m1e7_v_human.k2_tab` file.

The `fasta36` command above illustrates several command line options (Table 3.9.2):

```
-s BP62 # use the BLOSUM62 matrix with -11/-1 gap penalties
-S      # soft-mask lower-case characters (low complexity found
by pseg)
-m 8CC  # show results using commented blast tabular format(-
m 8C),
        # CIGAR alignments, and annotations (the second ‘C’ in -m
8CC)
-E 1e-3 # only show results with expect < 0.001
-V \!script/... # annotate the library/subject sequences
```

2. View the results file using

```
$ more c4mle7_v_human.k2_tab
```

The compact alignment line in Figure 3.9.2B provides all the annotation information in Figure 3.9.2A, but in a compact, easily parsed, form. In particular, the annotation encoding (the last tab-delimited field in the output) shows the location and substitution state for both the variant and active site residues. In addition, the boundaries and statistical significance of the C.AB-hydrolase domain is shown. In this example, the compact annotation summary concisely highlights the identity of the three active site residues that are part of the charge relay system annotated by UniProt.

Figure 3.9.2 also illustrates the power of sub-alignment scoring—the ability to divide an aligned region into different parts and calculate the contribution of each part of the alignment to the overall score Mills and Pearson (2013). In this example, Pfam annotates an α/β hydrolase domain between residues 123–363 of the human ABHD1_HUMAN protein includes 255 of the 256 states of the α/β hydrolase model (PF00561). However, the sequence alignment extends beyond the α/β -hydrolase domain, by 86 additional aligned residues in the human protein, almost 25% of the alignment. Sub-alignment scoring breaks the overall alignment score into two parts, the α/β -hydrolase domain between residues 123–363 of the human protein, and an unannotated region from residues 1-122 (the NODOM or no domain region). While the NODOM region includes 86 aligned residues, it contributes only about 13% of the score, suggesting that the apparent homology implied by the alignment may actually be alignment overextension (Gonzalez and Pearson, 2010; Mills and Pearson, 2013). Alternatively, the “overextension” may reflect a conservative α/β -hydrolase domain model, and the additional aligned sequence might support a longer domain. This latter alternative could be tested by searching Swiss-Prot with the 86 residue NODOM region, and looking for other proteins that annotate an homologous α/β -hydrolase domain.

ALTERNATE PROTOCOL 2 – Using annotation files

BASIC PROTOCOL 2 used a perl script to dynamically capture feature and domain annotation information from UniProt and Pfam web resources. For each sequence accession reported in the similarity search, the script queries Pfam and UniProt web services to obtain domain and functional site information. For example, running the script on one of the high scoring sequences, ABHD_HUMAN, produces the result:

```
$ scripts/ann_upfeats_pfam_www.pl 'sp|Q96SE0|ABHD1_HUMAN'
==:Active site
==*:Modified
=#:Substrate binding
^=:Metal binding
=@:Site
>sp|Q96SE0|ABHD1_HUMAN
```

```
54 V Q dbSNP:rs34127901
```

```

123 - 365 C.AB_hydrolase :1
137 V E dbSNP:rs6715286
203 = - Active site: Charge relay system
329 = - Active site: Charge relay system
358 = - Active site: Charge relay system
371 V C dbSNP:rs2304678

```

In this example, the lines beginning with '=' indicate the symbols used to highlight different types of sites (only active sites are annotated for this protein), while 'V' in the second column indicates a variant site and '-' indicates the boundaries of domain. The information produced by the `ann_upfeats_pfam_www.pl` script is then used to annotate sites and partition the alignment scores in the search.

Annotation and domain information can also simply be provided in a file. For example, this information can be used to characterize the conservation of the exons of the `sp|P09488|GSTM1_HUMAN` protein:

```

>sp|P09488|GSTM1_HUMAN
1      -      12 exon_1
13     -      37 exon_2
38     -      59 exon_3
60     -      86 exon_4
87     -     120 exon_5
121    -     152 exon_6
153    -     189 exon_7
190    -     218 exon_8

```

Necessary Resources

Hardware—A modern Windows (32-bit, 64-bit), or Mac OSX (32-bit, 64-bit), or Unix/Linux computer with at least 50 MB of free disk space for the programs and 100 GB of disk space for protein sequence databases.

Software—The FASTA programs, installed and configured as described in Support Protocol 1. Included with the FASTA distribution is the program `scripts/ann_upfeats_pfam_www.pl`, which extracts annotation information from the UniProt/EBI and Pfam web servers (See Support protocol 1).

The `scripts/summ_domain_ident.pl` perl script, included in the FASTA distribution.

Files—The Swiss-Prot database downloaded from NCBI (<ftp://ftp.ncbi.nih.gov/pub/blast>), EBI (<ftp://ftp.ebi.ac.uk/databases>), as described in Support Protocol 2, and “soft-masked” using `pseg` to produce `/slib/swissprot.lseg` (Support protocol 1).

The `scripts/gstml_hum_exons.ann` file, included in the FASTA distribution.

The `seq/gstml_human.aa` query sequence file, included in the FASTA distribution.

1. After downloading and installing the FASTA programs and scripts (See Support Protocol 1), run the `ggsearch36` program by typing the command:

```
$ ggsearch36 -E 0.001 -m 8CC -q -s BP62 -S -V q
\<gstml_hum_exons.ann \
  ../seq/gstml_human.aa /slib/swissprot.lseg >
gstml_hum_v_sp.gg_exons
```

(Again, a single command has been broken into two lines using the ‘\’ at the end of the line. In addition, a ‘\’ precedes the ‘<’ before the `gstml_hum_exons.ann` file name.) This command sends the results to the `gstml_v_sp.gg_exons` file.

The `ggsearch36` program calculates a global-global sequence alignment (alignments extend from the beginning to the end of both the query and subject/library sequence). As a result, in this example the entire query sequence, and thus all eight exons, are aligned. In contrast to the first example, the ‘<’ in the annotation option `-V q` \<gstml_hum_exons.ann specifies that the file be read, rather than executed, to obtain the annotation information. (The ‘q’ indicates the annotation information applies to the query sequence.) Just as in the previous example (Figure 3.9.2), the query annotation directs `ggsearch36` to subdivide the alignment, and report the fraction identical for each of the exon regions.

2. View the results file using: `$ more gstml_human_v_sp.gg_exons`

After several lines beginning with #, you should see a line of the form:

```
sp|P09488|GSTM1_HUMAN gi|121735|sp|P09488.3|GSTM1_HUMAN
100.00 218 0 0\
  1 218 1 218 0 113.6 218M |RX:
1-12:1-12:s=64;b=6.9;I=1.000;Q=120.2;C=exon_1 ...
```

All but the last two fields are the standard blast tab-delimited summary format. The second to the last field is the CIGAR alignment string (`218M` indicates 218 matches for the 100% identical alignment) and the domain annotation for each of the eight exons (only the first is shown).

3. The `scripts/summ_domain_ident.pl` script can then be used to summarize the identity across each exon for the homologs found by `ggsearch36`:

```
$ summ_domain_ident.pl gstm1_human_v_sp.gg_exons
```

This program produces output of the form:

subj_acc	ident	exon_1	exon_2	exon_3	exon_4	exon_5	exon_6	exon_7	exon_8
sp GSTM1_HUMAN	100.0	1.000+	1.000+	1.000+	1.000+	1.000+	1.000+	1.000+	1.000+
sp QGSTM2_PONAB	85.78	0.818-	0.920+	1.000+	0.926+	0.735-	0.781-	0.784-	0.966+
sp GSTM1_MACFA	85.78	0.909+	1.000+	1.000+	0.926+	0.735-	0.656-	0.946+	0.793-
sp GSTM2_MACFA	85.78	0.818-	0.920+	1.000+	0.926+	0.735-	0.781-	0.784-	0.966+
sp GSTM1_BOVIN	83.94	1.000+	0.760-	1.000+	0.889+	0.676-	0.781-	0.919+	0.828-
sp GSTM2_MOUSE	83.94	0.909+	0.920+	0.955+	0.889+	0.735-	0.844+	0.865+	0.690-
sp GSTM2_HUMAN	84.40	0.818-	0.960+	1.000+	0.926+	0.647-	0.750-	0.784-	0.966+
sp GSTM2_RAT	81.65	0.909+	0.840+	0.955+	0.852+	0.706-	0.812-	0.892+	0.655-
sp GSTM4_RAT	82.57	0.909+	0.960+	0.955+	0.852+	0.765-	0.781-	0.730-	0.793-
sp GSTMU_MESAU	81.65	0.818+	0.960+	0.909+	0.889+	0.706-	0.812-	0.838+	0.655-
sp GSTM7_MOUSE	81.19	0.909+	0.920+	0.955+	0.852+	0.735-	0.781-	0.757-	0.759-

This table summarizes the average alignment identity in each of the 8 exons, adding a '+' if the exon sequence identity is greater than or equal to the average across the protein, and a '-' if it is lower. A quick scan of the '+' and '-' symbols suggests that the parts of the protein encoded by exons 2–4 are more conserved than average (more '+' symbols), while the parts encoded by exons 5–8 diverge more rapidly (more '-' symbols). The two groups of exons correspond to the two domains that Pfam (Finn *et al.*, 2014), and other domain databases, annotate on GSTM1_HUMAN. Exons 1 – 4 encode an N-terminal domain, while exons 5 – 8 encode a C-terminal domain, which may evolve more rapidly.

Summary—Alignment annotation, combined with BLAST tabular output format (`-m 8CC`) provides a compact and efficient strategy for integrating evolutionary, functional, and structural information in a format that is easily parsed for large scale analysis. The annotation integration and sub-alignment scoring process is very general. Annotation scripts can access data from multiple resources (e.g. both UniProt and Pfam are used by the `ann_upfeats_pfam_www.pl` script used in BASIC PROTOCOL 2), and different kinds of information, e.g. Pfam domains and exon domains, can be combined in the same search. The `scripts/` directory provides annotation scripts that reference the UniProt and Pfam web servers (`'_www_'`) as well as scripts that query a local database (see Unit 9.4). For large-scale analyses, the scripts that use local MySQL versions of the annotation databases are considerably faster.

As more functional annotation becomes available, it will become increasingly easy to identify the conservation state of functionally critical residues in homologs. Identifying homologs is a first step in characterizing a sequence; site annotation and sub-alignment scoring allows biologists to focus on functional sites and regions in homologs.

GUIDELINES FOR UNDERSTANDING RESULTS

The natural question that every similarity search is designed to answer is: “Which proteins are homologous to my query sequence?” The answer can be deduced from the statistical estimates in the summary of the high-scoring alignments from the database search (Figure 3.9.1B). Both the FASTA and BLAST programs provide several numbers for evaluating the quality of an alignment; in Figure 3.9.1B, FASTA provides the raw BLOSUM50 gapped similarity score (in the opt column), the bit score (which is comparable to a BLAST bit score and can be used to calculate the probability of a score using the equation above), and the expectation, or E()-value. The alignment provides additional measures of alignment quality: the percent identity (gapped or ungapped) and the alignment length.

Annotation scripts (the `-v !script.pl` option) can help address the related question, “is this homolog likely to have a similar function?” While the relationship between structure and function is complex (new function prediction Unit), homologous proteins that differ at critical functional residues are less likely to have similar functions. Annotation integration allows the FASTA programs to highlight the conservation states of specific functional residues.

Using the E()-Value (expect) to Identify Homologs

In evaluating the search results, the expectation or E()-value is the most reliable and sensitive indicator of likely sequence homology. For protein:protein alignments, if the E()-value is less than 10^{-6} , the sequences are almost certainly homologous (Unit 3.1). Sequences with E()-values $<10^{-3}$ are almost always homologous as well, but in these cases, one must ensure that the statistical estimates are accurate (see below). Indeed, in most cases, sequences with $E() < 0.01$ are homologous. It is important to remember that the E()-value simply reports the number of times a similarity score is expected by chance, or the number of expected false positives (non-homologs) per search. Since there will be a highest-scoring unrelated sequence in every search of a comprehensive database, the E()-value for the highest-scoring unrelated sequence (the highest-scoring potential false positive) will be approximately equal to 1 (see Critical Parameters, Selecting the Database). Of course, distantly related homologous sequences may also have $E() \sim 1$, or even higher. A similarity score with $E() < 0.01$ or $E() < 0.001$ simply says that this score should occur by chance once in 100 or once in 1000 database searches. As noted in Critical Parameters, the E() value depends on the database size; thus, in Figure 3.9.1B, E(21,039) is shown, because 21,039 sequence alignment scores were examined to find the best alignments.

Evaluating Statistical Estimates

The observation that the highest-scoring unrelated sequence should have an E() value near 1.0 provides a strategy for evaluating the accuracy of the statistical estimates provided by

FASTA (or BLAST) by examining the $E()$ -value of the highest-scoring candidate unrelated sequence. While it is impossible to know that a sequence is unrelated simply by looking at the alignment score, it is possible to do additional searches, or to look at domain content, to infer non-homology. The five human proteins with statistically significant similarity to the *E. histolytica* protein in Figure 3.9.1B all contain α/β -hydrolase domains (this is confirmed by the search performed in BASIC PROTOCOL 2). In addition, several α/β -hydrolase proteins do not share significant similarity. To confirm the accuracy of the statistical estimates, we seek the highest-scoring non-homologous protein—a protein that does not share significant similarity with α/β -hydrolases and does not contain a α/β -hydrolase domain.

In Figure 3.9.1B, the strongest candidate non-homolog is CLTR2_HUMAN, a cysteinyl leukotriene receptor, with an $E()$ -value < 0.11 , and ADCYA_HUMAN, an adenylate cyclase, with $E() < 0.33$. To test whether either of these is an α/β -hydrolase homolog, we can search a large comprehensive database, e.g., Swiss-Prot, with the candidate non-homologs to see whether CLTR2 or ADCYA share significant similarity with any α/β -hydrolases. In fact, both are clear non-homologs. When CLTR_HUMAN is compared to Swiss-Prot proteins using `-s BP62`, it is clear that it belongs to the G-protein coupled receptor (GPCR) family. CLTR_HUMAN finds 1495 GPCR homologs with $E()$ -values < 0.001 , the highest scoring non-GPCR has an $E()$ -value < 0.85 , and there are no α/β -hydrolases with $E()$ -values < 10 . This is the expected behavior for a non-homolog. The C4M1E7_ENTHI : CLTR2_HUMAN alignment appears in the search by chance, not because of homology.

Likewise, a more comprehensive search with ADCYA_HUMAN against Swiss-Prot finds four mammalian type 10 homologs with two adenylate cyclase domains, another set of marginally significant bacterial proteins with adenylate cyclase domains, and many more distantly related adenylate cyclase proteins, but no sequences with $E() < 10$ that contain the α/β -hydrolase domain. Thus, by identifying the two highest scoring non-homologs, both with scores near 1.0, (0.11, 0.33), we can be confident that the statistical estimates are accurate. By demonstrating that the statistics are accurate, we have more confidence that the very significant alignments with α/β -hydrolases reflect excess similarity produced by common ancestry (homology).

Statistics from Shuffled Sequences

The FASTA programs estimate the statistical significance of an alignment by examining the distribution of alignment scores from the “unrelated” sequences in the sequence database. While the FASTA program does not know a priori which sequences are “unrelated,” it assumes that, in a comprehensive database search, fewer than 5% of the sequences in the database will be related, and FASTA uses several strategies to exclude those sequences from the statistics calculation. Thus, one can interpret the $E()$ -value as a measure of how often the query sequence would match a sequence like those in the database by chance.

Sometimes, however, the query sequence is different from most of the sequences in the database due to sequence composition, or some other sequence ordering peculiarity, rather than homology, and the sequence’s alignment score is high, not because of homology, but because it shares the property. For example, a membrane protein with strongly biased amino

acid composition might have marginally significant alignment scores with other membrane proteins, not because they are homologous, but because they have a high fraction of hydrophobic amino acids (in practice, statistically significant matches to non-homologous membrane proteins rarely occur, but the high-scoring non-significant matches are often other membrane proteins). When the statistical estimates are suspect, the FASTA programs can be run in “pairwise comparison” mode, where “library” becomes a single sequence. When a query sequence is compared to a single sequence or a small number of sequences, the FASTA program (typically `ssearch36` for protein:protein or DNA:DNA comparison, and `fastx36` for DNA:protein comparison) aligns the query sequence to a single “library” (or “subject”) sequence, calculating an optimal alignment score.

The program then shuffles the “library” sequence 200 to 1000 times, producing 200 to 1000 new random sequences with the same length and sequence composition, and uses the distribution of these scores to estimate the statistical significance of the original un-shuffled sequence. The programs can also use a “window” shuffling mode (`-v 10` for a 10 residue window) that preserves the local sequence composition within a local region while producing the random sequences. `ssearch36` can be used for either protein:protein or DNA:DNA comparison, though the shuffling strategy does not preserve the higher-order statistical properties of DNA sequences, and is thus less reliable for DNA (so lower significance thresholds should be used).

For example, to test whether the apparent similarity between the *E. histolytica* C4M1E7_ENTHI putative protein and the human ABD12_HUMAN α/β -hydrolase is supported by a shuffled sequence analysis, we can download the ABD12_HUMAN sequence and compare it to C4M1E7:

```
$ curl http://www.uniprot.org/uniprot/Q8N2K0.fasta > abd12_human.fa
$ ssearch36 -k 1000 -Z 21039 -s BP62 c4m1e7_enthi.fa abd12_human.fa
```

`ssearch36` can then calculate the Smith-Waterman score for an alignment of C4M1E7_ENTHI with ABD12_HUMAN, and then shuffle the ABD12_HUMAN sequence 1000 times and calculate the statistical significance of the un-shuffled score based on the distribution of scores from the shuffled sequence comparisons. In this example, the original (un-shuffled) sequence alignment has an alignment score with an E()-value $<8.5 \times 10^{-5}$, very similar to the significance estimated in the initial search.

In the command above, the `-Z 21039` option is included to ensure that the shuffled statistical significance is calculated based on the number of sequences compared in the original database search. Omitting the `-Z` option would improve the statistical significance 20,000-fold, but since the candidate homolog was found in a search of 21,039 sequences, the statistical effect of those additional alignments must be included in subsequent estimates based on shuffles.

Statistical estimates for protein:protein alignments are much more reliable than DNA:DNA statistics. DNA:protein alignment statistical estimates, such as those produced by `fastx36`

and fasty36 (Table 1) are intermediate in accuracy. DNA:protein alignment statistics can be misleading because out-of-frame translations sometimes produce low-complexity regions that match low-complexity domains in the sequence databases. While low-complexity regions can often be masked by pseg, occasionally a low-complexity region slips through, producing an apparently significant match. fastx36/fasty36 shuffling can identify these pathological cases and substantially improves statistical reliability some cases.

For “normal” soluble proteins, the statistical estimates from the original similarity search will match quite closely those from ssearch36 or fastx36/fasty36. But if there is some concern about the reliability of the significance estimate, the shuffling programs can provide an alternative estimate.

COMMENTARY

Background Information

There are three widely used sets of programs for searching protein and DNA sequences, the BLAST package (UNITS 3.3 & 3.4), HMMER (Eddy, 2011; Finn et al., 2011) and the FASTA package. These programs do similar things; they compare a query sequence to a library of sequences, calculating tens of thousands to millions of similarity scores, and report back the library sequences that are most similar to the query. Most importantly, BLAST, HMMER, and FASTA calculate the statistical significance of the alignment scores, so that investigators can judge whether an alignment score is likely to have occurred by chance. Without accurate statistical significance estimates, it is impossible to evaluate the scientific importance of an alignment score. Because there are so many sequences, matches that seem intuitively unlikely will often occur by chance—for example, a search of the nr database, containing 400 million residues, with a 300-residue query sequence, is expected to match 9 identical residues more than 50% of the time. In contrast, relatively low-identity alignments (<20% identical over 300 residues) can be very statistically significant, with $E() < 10^{-6}$. One should always focus on the statistical significance, or expectation [E()] value, when evaluating whether two sequences are likely to be homologous.

The BLAST and FASTA packages have programs that perform many of the same functions (Table 1) for protein:protein, DNA:DNA, and translated protein:DNA comparison (currently, HMMER does not offer translated searches). FASTA has several programs for searching with short, ordered or unordered, noncontiguous peptide or DNA sequences (fasts36, fastf36, fastm36). While the type of query sequence and target database usually determine the program, one should search with protein sequences whenever possible. Protein sequence comparison is 5- to 10-fold more sensitive than DNA sequence comparison; it is routine to identify sequences that diverged more than a billion (plants/animals) or even two billion (prokaryotes/eukaryotes) years ago with protein or translated protein sequence comparisons; searches with DNA sequences rarely find significant matches in sequences that diverged more than 250 million years ago. fastx36, fasty36, tfastx36, and tfasty36 can align DNA sequences with frame-shifts, so even if open reading frames cannot be identified unambiguously because of sequencing errors, the translated DNA sequence can be correctly aligned with an homologous protein.

Searching smaller databases also increases sensitivity. Now that a large number of fully sequenced prokaryotic, fungal, plant, and animal genomes are available, it is much more effective to search complete genomes from taxonomic neighbors than to search the comprehensive *nr* or *sp-trembl* databases, which contain 50 – 75 million entries. Many distant homology relationships can be detected by searching an individual *E. coli* or *S. cerevisiae* proteome [$E(5,000) < 10^{-3}$], but would need an alignment score 10,000 times more significant to be detected in the context of *nr* or *sp-trembl*. Proteome data sets are available for all fully sequenced genomes.

Most researchers do similarity searches on the Internet, not on their local computers. Internet searches are much more convenient; one does not have to download the BLAST or FASTA packages, download sequence databases, reformat sequence databases, or keep the programs and databases up to date, because the Internet site does all this work. But Internet site searches are often slower and less flexible; by searching on one's own computer, it is possible to tailor the search parameters, database, and output formats to meet one's own research needs.

Critical Parameters

Selecting the Correct Program: The FASTA package provides programs for searching protein, DNA, or translated DNA sequence databases, using proteins, DNA, translated DNA, or short peptides as queries. Table 1 summarizes the programs that should be used for various analysis problems, and corresponding programs, if any, in the BLAST package (UNITS 3.3 & 3.4). In general, if one has a protein sequence, one should use *fasta36* or *ssearch36* (a Smith-Waterman implementation; Smith and Waterman, 1981; UNIT 3.10). If one has a DNA query sequence that codes for protein, one should use *fastx36* (Pearson et al., 1997), which compares a DNA query to a protein database. For most researchers, *fasta36* (protein) and *fastx36* (DNA) will meet 80% or more of search needs. In both cases a protein sequence database is searched. Sometimes, it may be desirable to check whether a particular protein sequence is present in an unfinished (or incompletely annotated) genome. Here, *tfastx36*, which compares a protein sequence to a DNA sequence database, can be used.

The FASTA package also provides some more specialized programs, particularly *fasts36* (Mackey et al., 2002) which is designed to search with a set of unordered oligopeptide (or DNA) sequences, and *fastm36*, which does the same search with an ordered set of oligopeptides (or nucleotides). *fasts36* is designed to identify proteins from de novo tandem mass spectroscopy (MS/MS) sequence data. Three or four oligopeptides of length 4 to 6 are typically sufficient to identify, by similarity, sequences that diverged in the past 400 million years. Thus, MS/MS peptides from a hamster or rabbit can be reliably identified by searching against the human proteome.

Selecting the Database: The statistical significance, or expectation value, reported by a FASTA or BLAST search is the product of two terms: (1) the probability of the pairwise sequence similarity score, typically calculated using the extreme-value distribution, and (2) the size of the database searched. Both BLAST and FASTA (version 36 and later) report

similarity scores in terms of bits; if an alignment of two 300-residue sequences has a bit score of 40, the probability of that pairwise alignment occurring by chance is:

$$P(40 \text{ bits}) = mn2^{(-bit)} = 300 \times 300 \times 2^{-40} = 8.2 \times 10^{-8} \sim 10^{-7}$$

While a probability of 10^{-7} may seem significant, it must be corrected for the number of sequences that were examined to find the alignment. If the 40-bit alignment was found after a search of the NCBI *nr* database, which contains more than 50 million protein sequences, then the expected number of times a 40-bit score would be seen by chance is:

$$E(40 \text{ bits}) = P(40 \text{ bits}) \times D = 8.2 \times 10^{-8} \times 50 \times 10^6 = 4.1$$

(*D* is the database size) and is thus not statistically significant; a similarity as good or better is expected by chance four times in every database search. In contrast, if one were searching for a homolog in *E. coli*, or many other bacteria, one could either search the bacterial proteome or the proteome of related bacteria, in which case the expectation [*E*()] value for exactly the same alignment score would be:

$$E(40 \text{ bits}) = P(40 \text{ bits}) \times D = 8.2 \times 10^{-8} \times 4000 = 3 \times 10^{-4}$$

Thus, an alignment score that would be clearly statistically significant in a search of a bacterial proteome would not be significant when searching the *nr* database.

This relationship between database size, statistical significance, and the ability to infer homology is disconcerting to many researchers. If something is homologous, it is argued, it should be homologous regardless of the database in which it is found. This is true of course, but misses a fundamental asymmetry in similarity searching: sequences that share statistically significant similarity can be inferred to be homologous, but the inverse is not true. Non-significant similarity does not imply non-homology; there are many examples of homologous proteins that do not share significant pairwise sequence similarity. The problem with searching large databases is the “noise” associated with the many additional opportunities to obtain a high score by chance. In general, one should search the smallest comprehensive database that is likely to contain homologs to the protein of interest. For vertebrate sequences, this would be the human genome; for invertebrate sequences, *Drosophila* and *C. elegans* are available. By searching a group of eukaryotic genomes—e.g., human (20,000 proteins), *Drosophila* (14,000 proteins), *C. elegans* (20,000 proteins), *S. cerevisiae* (6,700 proteins), and *Arabidopsis* (27,000 proteins)—the database will be about the same size as the Swiss-Prot database, but will be both more comprehensive and less redundant for eukaryotic proteins.

Thus, *nr* should be the last, rather than the first, database to search. The most effective strategy would be to search individual complete proteomes from organisms that are close to the query sequence, then a taxonomically deeper set, then Swiss-Prot, then *nr* (Unit 3.1).

Changing Search Parameters—Scoring Matrices: The FASTA programs provide program options to modify the scoring matrix (-s matrix, see Table 3 and Unit 3.5) and gap penalties (-f, -g) in the search, to exclude low-complexity regions from the initial alignment scores (-S), to select sequences in a molecular weight range (-M), and to modify the output formats (-m, Table 4) and default statistical methods (-z, Table 5). Tables 2–5 show the scoring, output, and statistics options. The options in Table 2 must be specified on the command line when the FASTA program is started. All FASTA options begin with a minus sign and must come before the query file, library file, and ktup parameters. Thus:

```
$ fasta36 -S -s BP62 query.aa /slib/swissprot > results.file # correct
```

is correct, while

```
$ fasta36 query.aa /slib/swissprot -S -s BP62 > results.file
# wrong; options must be first
```

will fail. When the FASTA programs are run from the command line, the results should be saved to a file, e.g.,

```
$ fasta36 query.aa /slib/swissprot > results.file
```

The most commonly used option should be -S, which causes the program to ignore low-complexity regions when searching suitably formatted databases (see Support Protocol 2). The next most common parameter change should be the scoring matrix. To maximize sensitivity, the FASTA programs use the BLOSUM50 matrix, with gap penalties of -10 (gap-open) and -2 (gap-extend, a one residue gap costs -12). While this scoring matrix and gap-penalty combination is capable of identifying long homologous regions with less than 20% sequence identity, it will be less effective at identifying shorter domains. Shallower scoring matrices (e.g. the BLOSUM62 -11/-1 matrix and gap penalties used by BLASTP, and specified with -s BP62) allow shorter domains, with higher identity, to be identified. Changing to a much shallower scoring matrix (e.g., VT40 or VT20, Unit 3.5) can limit the evolutionary look-back time of a search from 2,000 million years or more (BLOSUM50, BLOSUM62) to 200 million years or less.

The -s option changes the default scoring matrix (BLOSUM50; UNIT 3.5). As shown above, -s BP62 specifies a search with the BLOSUM62 scoring matrix and gap open/extend penalties (-11/-1) used by the BLASTP (UNIT 3.4) program. The FASTA programs provide a comprehensive selection of BLOSUM matrices (Henikoff and Henikoff, 1992) and evolutionary model based matrices: PAM, (Jones et al., 1992) and VTML, (Mueller et al., 2002). Table 3 lists the built-in scoring matrices and associated default gap penalties. The FASTA programs also provide an option for dynamic scoring matrix adjustment using the -s ?matrix option. If the matrix name (Table 3) begins with a '?', e.g., ?BP62, the FASTA programs will check the length of the query sequence, and, if the query sequence is too short

to produce a 40-bit score with the specified scoring matrix, the program will scan through the matrices in Table 3 to find a matrix that could produce a 40-bit score, based on the information content of the matrix. For example, a search with a 40 residue protein with `-s ?BP62` would select the VT80 matrix, because it provides 1.39 bits per aligned residue, while the next less shallow matrix (VT120) matrix only provides 0.94 bits per residue (Unit 3.5).

In addition to the built-in matrices, any scoring matrix can be specified by providing a file of scores in the same format as the BLAST scoring matrix format. Built-in scoring matrices have default gap penalties that are effective with that matrix ((Reese and Pearson, 2002), Unit 3.5), but these penalties can be changed with the `-f` and `-g` options. Gap penalties should only be increased; decreasing gap penalties can shift alignments from local to global, invalidating the statistical model. The default scoring matrices and gap penalties provide smooth transitions in target percent identity, alignment length, and information content (Unit 3.5).

Changing Search Parameters—Statistics: Alternate statistical parameter estimation routines are specified with the `-z` option (Table 5). Again, the default `-z 1` is preferred for most cases, but there are two situations where an alternative might be more effective. The first relates to the fact that the statistical estimation routines assume that most of the sequences in the database are unrelated to the query. If searching a library of related proteins, then one must use the `-z 11` or `-z 21` option. With the `-z 11` option, the program shuffles each of the library sequences to produce a random sequence, calculates a similarity score, and uses the scores from the random sequences to estimate the statistical parameters. The `-z 21` option provides two statistical estimates (E()-values), the first based on the similarity search, and a second based on shuffling the sequences of the high scoring sequences found in the search. The second option provides a potentially more conservative estimate of statistical significance when searching a comprehensive database. Rather than shuffle every sequence in the database, `-z 21` performs a smaller number of shuffles (500 by default, set with the `-k #` option). For even more conservative statistical estimates, a “window-shuffle” can be specified. `-v 10` causes the shuffles to be done with groups of 10 residues, preserving local amino-acid composition bias, while shuffling the sequence.

Search Options for Large-Scale Comparison—Large-scale sequence comparison can generate hundreds of megabytes of output data, so it is critical that: (1) every effort is made to reduce false-positive results, and (2) only essential information be captured. False positives (unrelated sequences with apparently statistically significant scores) can be reduced dramatically by including the `-S` option and searching sequence databases that have low-complexity regions indicated by lowercase letters with `pseg`. In general the statistical significance threshold should be lowered as well. For protein sequence comparisons, by default `fasta36` reports all alignment scores with $E() < 10.0$; in a large-scale sequence comparison with thousands of queries, this would produce tens of thousands of non-significant scores. For a search with thousands of DNA queries against a protein sequence database with an expectation threshold of 0.001 (`-E 0.001`) would reduce the *per-search*

expected false positive rate to 10^{-3} , but for a set of 1,000 searches, ~1 false positive (unrelated sequence) is expected.

```
$ fastx36 -S -E 0.001 DNA_reads.fa up_human.lseg > reads_vs_human.fx
```

Very large scale searches largely preclude looking at individual alignments, so the FASTA programs offer two output formats FASTA-tabular (`-m 9`) and BLAST-tabular (`-m 8`) that provide alignment scores, expectation values, and alignment boundaries in a very compact, easily parsed, format (Figure 3.9.2B). When using the `-m 9` or `-m 9c` options (Table 2), all the alignment output is usually excluded by using the `-d 0` option. The `-m 8` (BLAST tabular) option does this automatically. For example, to identify homologs shared by the *E. coli* and human proteomes using the sensitive `ssearch36` program (an accelerated version of the Smith-Waterman algorithm, (Farrar, 2007; Smith and Waterman, 1981).

```
$ ssearch36 -S -E 0.001 -m 8CC ecoli_prot.lib up_human.lseg > eco_hum.res
```

Again, in this example, setting the E()-value threshold (`-E 0.001`) this low means that only 4 false-positives are expected, on average, from the 4,000 proteins in *E. coli*. The `-m 8CC` option provides alignment encodings and domain annotations (Fig. 3.9.3) in a format that is compatible with BLAST tabular format (Figure 3.9.2B and BASIC PROTOCOL 2).

Both FASTA-tabular and BLAST-tabular formats can also provide residue-by-residue alignment information using the CIGAR string (`-m 9C`, `-m 8CC`) encoding. (The FASTA programs also offer an older FASTA alignment encoding: `-m 9c`, `-m 8Cc`.) If annotation information is available from a `-V !script.sh` program, an additional annotation field is displayed.

Acknowledgments

WRP is supported by a grant from the National Library of Medicine, LM04969.

Literature Cited

- Eddy SR. Accelerated profile HMM searches. PLoS Comput Biol. 2011; 7:e1002195. [PubMed: 22039361]
- Farrar M. Striped Smith-Waterman speeds database searches six times over other SIMD implementations. Bioinformatics. 2007; 23:156–161. [PubMed: 17110365]
- Finn RD, Bateman A, Clements J, Coghill P, Eberhardt RY, Eddy SR, Heger A, Hetherington K, Holm L, Mistry J, Sonnhammer ELL, Tate J, Punta M. Pfam: the protein families database. Nucleic Acids Res. 2014; 42(Database issue):D222–30. [PubMed: 24288371]
- Finn RD, Clements J, Eddy SR. HMMER web server: interactive sequence similarity searching. Nucleic Acids Res. 2011; 39:W29–W37. [PubMed: 21593126]
- Gonzalez MW, Pearson WR. RefProtDom: A protein database with improved domain boundaries and homology relationships. Bioinformatics. 2010; 26:2361–2361. [PubMed: 20693322]
- Henikoff S, Henikoff JG. Amino acid substitutions matrices from protein blocks. Proc Natl Acad Sci USA. 1992; 89:10915–10919. [PubMed: 1438297]

- Huang X, Hardison RC, Miller W. A space-efficient algorithm for local similarities. *Comp Appl Biosci*. 1990; 6:373–381. [PubMed: 2257499]
- Jones DT, Taylor WR, Thornton JM. The rapid generation of mutation data matrices from protein sequences. *Comp Appl Biosci*. 1992; 8:275–282. [PubMed: 1633570]
- Kann MG, Goldstein RA. Performance evaluation of a new algorithm for the detection of remote homologs with sequence comparison. *Proteins*. 2002; 48:367–76. [PubMed: 12112703]
- Mackey AJ, Haystead TAJ, Pearson WR. Getting more from less: Algorithms for rapid protein identification with multiple short peptide sequences. *Mol Cell Proteomics*. 2002; 1:139–147. [PubMed: 12096132]
- Mills LJ, Pearson WR. Adjusting scoring matrices to correct overextended alignments. *Bioinformatics*. 2013; 29:3007–2013. [PubMed: 23995390]
- Mott R. Maximum-likelihood estimation of the statistical distribution of smith-waterman local sequence similarity scores. *Bull Math Biol*. 1992; 54:59–75. [PubMed: 25665661]
- Mueller T, Spang R, Vingron M. Estimating amino acid substitution models: a comparison of Dayhoff's estimator, the resolvent approach and a maximum likelihood method. *Mol Biol Evol*. 2002; 19:8–13. [PubMed: 11752185]
- Pearson WR. Effective protein sequence comparison. *Methods Enzymol*. 1996; 266:227–258. [PubMed: 8743688]
- Pearson WR, Lipman DJ. Improved tools for biological sequence comparison. *Proc Natl Acad Sci USA*. 1988; 85:2444–2448. [PubMed: 3162770]
- Pearson WR, Wood TC, Zhang Z, Miller W. Comparison of DNA sequences with protein sequences. *Genomics*. 1997; 46:24–36. [PubMed: 9403055]
- Reese JT, Pearson WR. Empirical determination of effective gap penalties for sequence comparison. *Bioinformatics*. 2002; 18:1500–1507. [PubMed: 12424122]
- Schwartz, RM.; Dayhoff, M. Matrices for detecting distant relationships. In: Dayhoff, M., editor. *Atlas of Protein Sequence and Structure*. Vol. 5. National Biomedical Research Foundation; Silver Spring, MD: 1978. p. 353-358.
- Smith TF, Waterman MS. Identification of common molecular subsequences. *J Mol Biol*. 1981; 147:195–197. [PubMed: 7265238]
- UniProt Consortium. UniProt: a hub for protein information. *Nucleic Acids Res*. 2015; 43(Database issue):D204–12. [PubMed: 25348405]
- Waterman MS, Eggert M. A new algorithm for best subsequences alignment with application to tRNA-rRNA comparisons. *J Mol Biol*. 1987; 197:723–728. [PubMed: 2448477]
- Wootton JC, Federhen S. Statistics of local complexity in amino acid sequences and sequence databases. *Comput Chem*. 1993; 17:149–163.
- Zhang Z, Pearson WR, Miller W. Aligning a DNA sequence with a protein sequence. *J Computational Biology*. 1997; 4:339–349.

Curr Protoc Bioinformatics. Author manuscript; available in PMC 2017 March 24.

expectation value. (C) The highest scoring alignment between C4M1E7_ENTHI and ABD1_HUMAN.

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

A.a graphical annotated alignment

```
>>sp|Q96SE0|ABHD1_HUMAN Abhydrolase domain-containing protein 1 OS=Homo sapi (405 aa)
Variant: 88S=54Q : P54Q : dbSNP:rs34127901
Region: 71-151:37-122 : score=44; bits=19.0; Id=0.221; Q=21.0 : NODOM
Variant: 166E=137E : D137E : dbSNP:rs6715286
Site:= : 232S=203S : Active site: Charge relay system
Site:= : 365D=329D : Active site: Charge relay system
Site:= : 398H=358H : Active site: Charge relay system
Region: 152-403:123-363 : score=290; bits=117.0; Id=0.279; Q=302.4 : C.AB_hydrolase
initn: 210 initl: 128 opt: 281 Z-score: 609.0 bits: 121.7 E(21039): 8.6e-28
Smith-Waterman score: 334; 26.5% identity (63.4% similar) in 344 aa overlap (71-403:37-363)
```

B.compact alignment annotations

```
#query_id      subject_id      %id  alen  mism  gaps  qstart  qend  sstart  send  expect  bits
tr|C4M1E7|C4M1E7_ENTHI sp|Q96SE0|ABHD1_HUMAN 26.45 344 225 28 71 403 37 363 8.6e-28 121.7
<cont> 20M1I20M1I11M1D19M4I96M3D35M5I21M6D11M2D33M1D31M4D19M (alignment encoding)
<cont> |XV:88S=54Q|XR:71-151:37-122:s=44;b=19.0;I=0.221;Q=21.0;C=NODOM (annotation encoding part 1)
<cont> |XV:166E=137E|X=:232S=203S|X=:365D=329D|X=:398H=358H (annotation encoding part 2)
<cont> |XR:152-403:123-363:s=290;b=117.0;I=0.279;Q=302.4;C=C.AB_hydrolase (annotation encoding part 3)
```

Figure 3.9.2. Alignment annotation

Annotations on the alignment of a putative *E. histolytica* protein (C4M1E7_ENTHI) with human ABD1, a α/β -hydrolase protein family member (Figure 3.9.1C). The annotations on this alignment were produced by the `scripts/ann_upfeat_pfam_www.pl` script provided with the FASTA package distribution. (A) A text and graphical presentation of the annotation and domain data. UniProt annotated Variant: and functional (Site:) information is shown, as well as the conservation state, e.g. 166E=137E of the annotated site. Pfam domain boundaries Region: are also used to produce sub-alignment scores. The boundaries in the query (152-403) and subject (123-363) sequences are shown, as are the raw score, bit score, fraction identical, and Q-score ($-10\log(p)$). (B) Compact alignment information, alignment encoding, and annotation information. The scores, and alignment start and end coordinates shown in Figure 3.9.1C and schematically in part (A) are reported here as tab-delimited fields. The single line beginning `tr|C4M1E7|C4M1E7_ENTHI` has been split into five lines to fit the page. The notations `<cont>` and parenthetical comments are not included in the single line output. Each of the fields in the `-m 8CC` output is separated by a `<tab>` character. The output fields match `blast` tabular output, with the addition of a CIGAR string and an annotation string. The annotation string includes all the annotation information shown in part (A).

subj_acc	ident	exon_1	exon_2	exon_3	exon_4	exon_5	exon_6	exon_7	exon_8
sp GSTM1_HUMAN	100.0	1.000+	1.000+	1.000+	1.000+	1.000+	1.000+	1.000+	1.000+
sp QGSTM2_PONAB	85.78	0.818-	0.920+	1.000+	0.926+	0.735-	0.781-	0.784-	0.966+
sp GSTM1_MACFA	85.78	0.909+	1.000+	1.000+	0.926+	0.735-	0.656-	0.946+	0.793-
sp GSTM2_MACFA	85.78	0.818-	0.920+	1.000+	0.926+	0.735-	0.781-	0.784-	0.966+
sp GSTM1_BOVIN	83.94	1.000+	0.760-	1.000+	0.889+	0.676-	0.781-	0.919+	0.828-
sp GSTM2_MOUSE	83.94	0.909+	0.920+	0.955+	0.889+	0.735-	0.844+	0.865+	0.690-
sp GSTM2_HUMAN	84.40	0.818-	0.960+	1.000+	0.926+	0.647-	0.750-	0.784-	0.966+
sp GSTM2_RAT	81.65	0.909+	0.840+	0.955+	0.852+	0.706-	0.812-	0.892+	0.655-
sp GSTM4_RAT	82.57	0.909+	0.960+	0.955+	0.852+	0.765-	0.781-	0.730-	0.793-
sp GSTMU_MESAU	81.65	0.818+	0.960+	0.909+	0.889+	0.706-	0.812-	0.838+	0.655-
sp GSTM7_MOUSE	81.19	0.909+	0.920+	0.955+	0.852+	0.735-	0.781-	0.757-	0.759-

Figure 3.9.3.
Output from scripts/summ_domain_ident.pl.

Table 1

Comparison programs in the FASTA36 package

FASTA program	BLAST equiv.	Description
fasta36	blastp/ blastn	Compare a protein sequence to a protein sequence database or a DNA sequence to a DNA sequence database using the FASTA algorithm Pearson (1996); Pearson and Lipman (1988). Search speed and selectivity are controlled with the <i>ktup</i> (word size) parameter. For protein comparisons, <i>ktup</i> = 2 by default; <i>ktup</i> = 1 is more sensitive but slower. For DNA comparisons, <i>ktup</i> = 6 by default; <i>ktup</i> = 3 or <i>ktup</i> = 4 provides higher sensitivity.
ssearch36		Compare a protein sequence to a protein sequence database or a DNA sequence to a DNA sequence database using the Smith-Waterman algorithm (Smith and Waterman, 1981, Unit 3.10). <i>ssearch36</i> uses SSE2 acceleration Farrar (2007), and is only 2 – 5X slower than <i>fasta36</i> .
ggsearch36/ glsearch36		Compare a protein sequence to a protein sequence database or a DNA sequence to a DNA sequence database using an optimal global:global (<i>ggsearch36</i>) or global:local (<i>glsearch36</i>) algorithm.
fastx36/ fasty36	blastx	Compare a DNA sequence to a protein sequence database, by comparing the translated DNA sequence in three frames and allowing gaps and frame-shifts. <i>fastx36</i> uses a simpler, faster algorithm for alignments that allows frame-shifts only between codons; <i>fasty36</i> is slower but can produce better alignments because frame-shifts are allowed within codons (Zhang et al. 1997).
tfastx36/ tfasty36	tblastn	Compare a protein sequence to a DNA sequence database, calculating similarities with frame-shifts to the forward and reverse orientations (Zhang et al. 1997).
fastf36/ tfastf36		Compares an ordered peptide mixture, as would be obtained by Edman degradation of a CNBr cleavage of a protein, against a protein (<i>fastf</i>) or DNA (<i>tfastf</i>) database (Mackey et al. 2002).
fasts36/ tfasts36		Compares set of short peptide fragments, as would be obtained from mass-spec. analysis of a protein, against a protein (<i>fasts</i>) or DNA (<i>tfasts</i>) database (Mackey et al. 2002).
lalign36		Calculate multiple, non-intersecting alignments using the sim2 implementation of the Waterman-Eggert algorithm(Waterman and Eggert 1987) developed by Xiaoqui Huang and Web Miller (Huang et al. 1990). Statistical estimates are calculated from Smith-Waterman scores of shuffled sequences.

Table 2**fasta36 Command Line Options**

Option	Description
-a	show entire length of both sequences in alignment (<i>fasta36</i> , <i>ssearch36</i>)
-A	force full Smith-Waterman alignment for DNA (<i>fasta36</i>)
-b #	number of high scores to display
-c # , #	(<i>fasta36</i> [<i>t</i>] <i>fast</i> [<i>xy</i>] <i>36</i> only) fraction of sequences to optimize, join
-C #	length of sequence name labeling aligned sequences
-d #	number of alignments to display (must be less than -b value)
-D	provide debugging output
-e file	<i>expand_script.sh</i> – script to expand sequences included in alignments
-E # [#]	Expectation threshold for displaying scores, alignments. The second value, indicates the threshold for secondary alignments.
-f #	Gap-open penalty (-10 default for proteins with default BLOSUM50 matrix)
-F #	Expectation lower limit for displaying scores, alignments
-g #	Gap-extension penalty (-2 default for proteins with default BLOSUM50 matrix)
-h	short help message
-help	longer help message
-H	show histogram
-i	DNA queries—search with reverse complement query only (<i>fasta36</i> , <i>fastx36</i> , <i>fasty36</i>). For <i>tfastx36</i> , <i>tfasty36</i> , only search reverse-complement from library (complement of -3)
-I	use interactive mode – prompt for query sequence file, library, and <i>ktup</i>
-j #	penalty for frame-shift within or between codons ([<i>t</i>] <i>fast</i> [<i>xy</i>] <i>36</i>)
-k #	number of shuffles for statistical estimates
-K #	maximum number of repeats for <i>lalign36</i>
-l file	location of FASTLIBS file
-L	Longer sequence descriptions in alignments
-m #	Alignment display options (Table 4)
-M # - #	Library length range; only sequences within the residue range are considered
-n	force DNA query; useful with DNA sequences with many ambiguous residues
-N #	break long sequences into segments of length #.
-o # , #	coordinate offsets for query and library sequences (previously -X)
-O file	send output to file (redirection using > file is preferred)
-p	force protein query
-P file	specify PSIBLAST format PSSM (Position Specific Scoring Matrix) file (<i>ssearch36</i> , <i>ggsearch36</i> , <i>glsearch36</i>)
-q / -Q	Quiet: do not prompt for any input [default]
-r	### DNA match/mismatch scores (+5 / -4 default)
-R file	score results file (for debugging)
-s file	Scoring matrix (see Table 3). Preceding the name with a ' ? ' (-s ?BP62) allows the matrix to be adjusted for short query sequences.

Option	Description
-S	Treat lowercase as low-complexity (soft-mask)
-t #	NCBI genetic code translation table
-T #	number of threads or workers (<code>pvc</code> omp / <code>mp</code> omp)
-U	treat query as an RNA sequence. In addition to selecting a DNA/RNA alphabet, this option causes changes to the scoring matrix so that G:A, T:C or U:C are scored as G:G -3.
-v #	do window shuffles with window size #
-V file	annotation script file. =file reads annotations, !file runs the file as a script (e.g., !annot.pl)
-w #	Alignment display width (default 60)
-W #	Alignment context (unaligned residues shown around alignment, default=30, fasta36, ssearch36)
-X	extended options, see the <code>doc/fasta_guide.pdf</code> in the FASTA distribution.
-z #	Statistical estimation strategy (Table 5)
-Z #	Effective database size for expectation value
-3	DNA queries—search with forward strand only; DNA libraries, search forward strand only

Table 3

fasta36 built-in scoring matrices

name	abbrev	gap-penalties	bit-scale	notes
BLOSUM series (Henikoff and Henikoff, 1992)				
BLOSUM 50	BL50	-10/-2	bits/3	
BLOSUM 62	BP62	-11/-1	bits/2	same as BLASTP
BLOSUM 62	BL62	-7/-1	bits/2	
BLOSUM 80	BL80	-10/-2	bits/2	
Dayhoff PAM series (Schwartz and Dayhoff, 1978)				
PAM 250	P250	-10/-2	bits/3	
PAM 120	P120	-14/-3	bits/3	
Modern PAM (MDM) series (Jones et al., 1992)				
MDM 10	MD10	-23/-4	bits/3	
MDM 20	MD20	-22/-4	bits/3	
MDM 40	MD40	-21/-4	bits/3	
VTML series (Mueller et al., 2002)				
VTML 10	VT10	-16/-2	bits/2	
VTML 20	VT20	-15/-2	bits/2	
VTML 40	VT40	-15/-2	bits/2	
VTML 80	VT80	-15/-2	bits/2	
VTML 120	VT120	-15/-2	bits/2	
VTML 160	VT160	-15/-2	bits/2	
VTML 200	VT200	-15/-2	bits/2	
Optima5	OPT5	-18/-2	bits/5	(Kann and Goldstein, 2002)

-m	Output options
0	Default, conventional 3 line alignment, identities indicated with ':', conservative replacements '.' :
MWRTC GPPYT	
::..:: :::	
MWKSC GYPYT	
1	Similar to -m 0, but conservative replacements indicated with 'x', non-conservative replacements 'X' (good for highly identical alignments)
MWRTC GPPYT	
 xx X	
MWKSC GYPYT	
2	Similar to -m 0, but 2-line alignment:
MWRTC GPPYT	
..KS..Y...	
3	One line alignment:
MWKSCGYPYT	
6	-m 0 using HTML output with links
B	BLAST alignment style:
Query	1
	MSHHWGYGKHNGPEHWHKDFPIAKGERQSPVDIDHTAKYDPSLKPLSVSYDQATSLRIL 60
	M+ WGY HNGP+HWH+ FP AKGE QSPV++ T ++DPSL+P SVSYD ++ IL
Sbjct	1
	MAKEWGYASHNGPDHWHELFNPAKGENQSPVELHTKDIRHDPQLPWSVSYDGGSAKTIL 60
BB	BLAST alignment style with BLAST-like beginning, end
8	BLAST tabular output
8C	BLAST tabular output with comments
8CC	BLAST tabular output with comments and CIGAR string, annotations (Figure 3.9.2B)
8CB	BLAST tabular output with comments and Blast BTOP alignment encoding, annotations.
9	FASTA tabular output
9i	FASTA -m 1 output with identity, alignment length
9C	-m 9 output with CIGAR string, annotations
9B	-m 9 output with BTOP string, annotations
10	Parse-able alignments with label:value output

Table 5

fasta36 statistical options (-z)

- z	Statistics options
1	Default linear regression fit to average unrelated score vs. log (length) of library sequences
2	Censored (5%) maximum-likelihood estimation of λ , K for extreme-value distribution
3	Altschul-Gish pre-calculated λ , K
4	Variation of -z 1 that does additional iterations to censor high scores
5	Variation of -z 1 that also does regression of variance with log(length) of library sequences
6	Censored maximum-likelihood estimates that include a composition term Mott (1992)
11–16	Identical to 1–2, 4–6, but fits against scores from shuffled library sequences (required when libraries of related sequences are searched)
21–26	Identical to 1–2, 4–6, but calculates a second E()-value (E_2) based on shuffles of the top alignment scores
-Z#	calculate E()-value based on specified database size