Sequence Analysis Spring 2000 Dr. Richard Friedman
(212)305-6901 (76901) friedman@cuccfa.ccc.columbia.edu 130BB

Lesson 6 Database Searching by Sequence

In this lab you are going to learn to use the World-Wide Web BLAST server at, the National Center for Biotechnology Information (NCBI), the Seg program in both its implementation in the GCG package and in its original form, and several programs in the Fasta family, both in the GCG and Fasta3 packages.

Reading:

GCG Tutorial Chapter 4.

Theory:

A. Usual terminology:

1. Homology - Sequence identity.

B. Politically correct terminology:

1. Similarity - Sequence identity.
2. Homology - Related by evolution.

C. My suggested terminology:

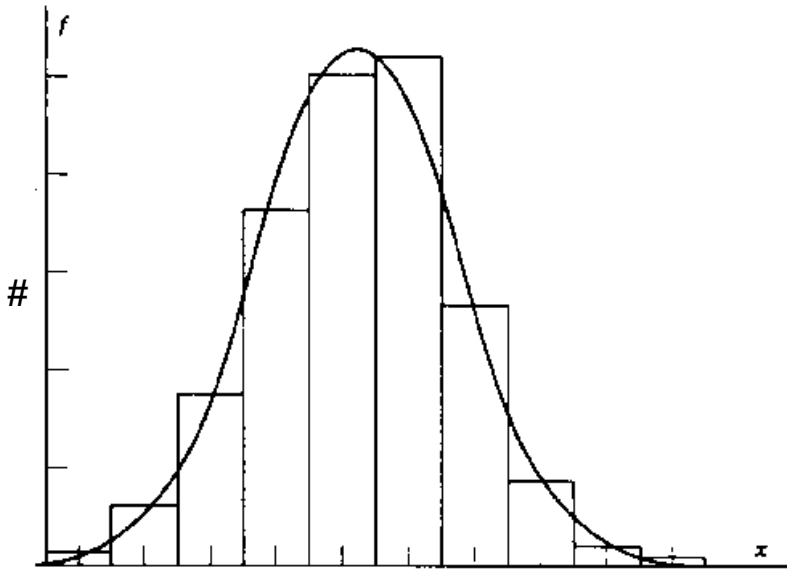
1. Homologous or Sequence Homology - Sequence identity (‘ομο -same λογοσ - word).
2. Homogeneous - Related by evolution (‘ομο – same γενουσ -family).
3. Homomorphic - Similar structure (‘ομο – same μορφοσ - shape).
4. Homopractic - Similar function (‘ομο – same πρακτικοσ - action).

homologous -> homogenous -> homomorphic (?-> homopractic) (-> metapractic)

5. Metapractic - Related function (μετα – near πρακτικοσ - action).

D. 2 sequences may be believed to be homologous (and therefore homogeneous, homomorphic, and metapractic) if their alignment score is greater than that which can occur by chance. A good measure of this is E, the expected number of sequences that by chance can have a score greater than or equal to the score of the alignment. The lower the E value the more likely the two sequences are homogenous. A good criterion for homogeneity is: $E \leq .001$ or .01 for proteins searched against a protein database . The E values for the comparison of two sequences is given by the extreme value distribution, to be defined below.

E. Random variables are often distributed according to the Gaussian (Normal) distribution (bell curve). For example, the distribution of heights in a class of fourth graders is approximated by the bell curve.

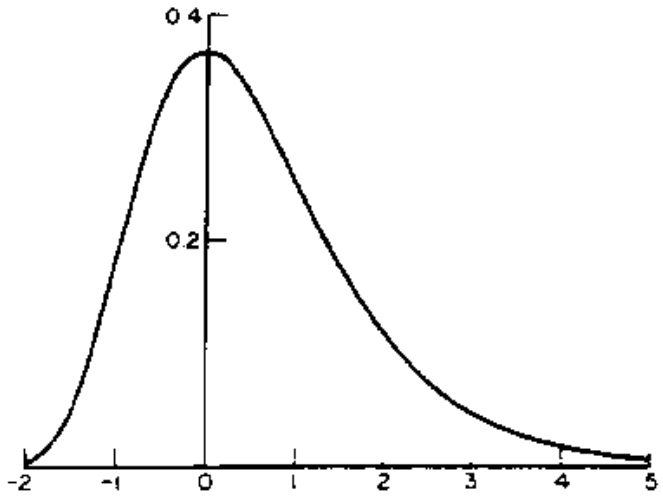


Height

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

e= 2.71828183...
x= random variable. For example, height of fourth graders in a class.
μ= mean
σ= standard deviation
f(x)= distribution of random variables, for example, distribution of heights.

F. Extreme value distribution: An extreme value distribution is the distribution of the extreme values of a set of variables. For example, the distribution of the heights of the tallest students in a class of fourth graders is given by the extreme value distribution.



General form of the extreme value distribution:

$$p(x \geq s) = 1 - e^{-ae^{-\lambda s}}$$

x= random variable, for example height s of tallest child in a number of fourth grade classes.
s= particular value of random variable, for example, a height.

$p(x \geq s)$ = probability that x is greater or equal to s, for example, the probability that the height of the tallest fourth grader in a class is greater than height =s.
a and λ are constants.

$$E = Dp(x \geq s)$$

E= expected number with value of variable greater than or equal to s. For example the number of fourth graders who are the tallest in their classes whose height is greater than s.

D= number of samples in distribution. In our example D is the number of fourth grade classes.

Application of the extreme value distribution to sequence analysis (Karlin-Altschul theory):

$$p(x \geq s) = 1 - e^{-Kmne^{-\lambda s}}$$

s= Smith-Waterman score.
$p(x \geq s)$ = probability that x is greater than or equal to s, for the comparison of two sequences.
K, λ = constants depending on the query sequence, the database sequence, and the scoring matrix, and are obtained either by theory or by fitting the statistical distribution.
m= number of residues in query sequence.
n= number of residues in database sequence.

$$E = Dp(x \geq s) \quad (1)$$

E= expected number of sequences with Smith-Waterman score greater than or equal to s.

D= number of sequences in database.

For $Kmne^{-\lambda s} \leq .01$

$$p(x \geq s) \approx 1 - (1 - Kmne^{-\lambda s}) = Kmne^{-\lambda s} \quad (2)$$

Combining eqs. 1 and 2 yields:

$$E = DKmne^{-\lambda s}$$

This method of calculating E is used by the Blast (Version 1) and Fasta (Version 2 and above) families of programs.

In PRSS3:

D = the number of sequences in the database in which the target sequence was discovered.

In the Blast2 family of programs:

$$E = p\left(\frac{N}{n}\right) = Kmne^{-\lambda s}\left(\frac{N}{n}\right) = KmNe^{-\lambda s}$$

N= number of residues in database.

G. Complexity (entropy)

$$S = -\sum_{i=1}^{N} \frac{n_i}{L} \log_2 \left(\frac{n_i}{L} \right)$$

S= complexity of sequence.
L= number of residues over which complexity is calculated.
N= number of kinds of residues.
For proteins, N=20.
For nucleic acids, N=4.
n_i = number of residues of the ith type in window L.

Summary of Commands:

Note: In this document different fonts have different meanings:

Times is used to explain commands.

`Courier is used to indicate commands and command options.`

`Courier italics are used to indicate command parameters, for example, filenames.`

`Courier bold is used to indicate commands that are not displayed.`

`Courier bold italics are used to indicate computer-generated output.`

Helvetica is used to indicate menu items.

Blast1 yields ungapped alignments and can yield many false positives. This version is described in the tutorial.
Blast2 yields gapped alignments and fewer false positives.

`blast`	GCG version of Blast2 that searches local databases. These databases are updated bimonthly.
`netblast`	GCG version of Blast2 that searches databases at NCBI. These databases are updated nightly.
`tofasta`	GCG program that translates sequence in GCG format to fasta format.

http://www.ncbi.nlm.nih.gov/BLAST/ NCBI web site.
Then go to BLAST 2.0 (Gapped BLAST and Graphical Viewer).
Then go to Basic BLAST Search.

BLAST Programs:

blastp	Protein query sequence compared to protein database.
blastn	Nucleic acid query sequence compared to nucleic acid database.
blastx	Nucleic acid query sequence translated in all six frames and compared to protein database.
tblastn	Protein query sequence compared to nucleic acid query sequence translated in all six frames.
tblastx	Nucleic acid query sequence translated in all six frames compared to nucleic acid database translated in all six frames.
`fromgenbank`	GCG program that translates sequence file from Genbank format to GCG format.
`netfetch` *`m81385`*	Copies the sequence whose accession number is "`m81385`" from the NCBI database into your directory. Saves as an RSF file.
`reformat` *`m81385.rsf{*}`*	Extracts file in standard GCG format from RSF file.
`datalist`	Lists the databases available to the local versions of Gapped BLAST and PSIBLAST.
`tofasta`	Translates a sequence file from GCG format to Fasta format.
`fromfasta`	Translates a sequence file from Fasta format to GCG format.
`seg`	GCG program that filters out low complexity regions from protein sequences.

`dust` *`human.n.tfa`*

Filters out low complexity regions from the nucleic acid file "*`human.n.tfa`*".

Fasta Programs:
Fasta1: No probability estimates. Discussed in tutorial.
Fasta2: Probability estimates. Discussed in GCG.
Fasta3: Improvement of Fasta2. Standalone package that is installed on Cuccfa. Programs in the Fasta3 package are named with a suffix "3". The corresponding programs in the GCG package lack the suffix.

`fasta`

GCG program in which a protein query sequence is compared to a protein database or a nucleic acid query sequence is compared to a nucleic acid database.

`fasta -nomon`

Runs Fasta without outputting sequences searched on screen.

`fasta -nomon` *`-in1=sequencename`* `-in2=`*`databasename:*`*
`-out=`*`outfilename`* `-D &`

Runs Fasta in background with no further input.

`What word size (* 2 *) ?` `1`
or
`-wor=`*`1`*

Makes Fasta word size =1.

Word = 1 is recommended for sensitive proteins searches.
Word = 4 is recommended for sensitive DNA searches.

`fasta3`

Fasta3 package program in which a protein query sequence is compared to a protein database or a nucleic acid query sequence is compared to a nucleic acid database.

`ssearch`

GCG program in which a protein query sequence is compared to a protein database or a nucleic acid query sequence is compared to a nucleic acid database using **a full Smith-Waterman algorithm**. Usually not recommend for nucleic acid searches.

`ssearch3`

Fasta3 package program in which a protein query sequence is compared to a protein database or a nucleic acid query sequence is compared to a nucleic acid database using **a full Smith-Waterman algorithm**.

	Usually not recommend for nucleic acid searches.
`tfasta`	GCG program in which a protein query sequence is compared to a a nucleic acid database. Not as good as Tfasty3.
`tfasta`3	Fasta3 package program in which a protein query sequence is compared to a nucleic acid database translated in all six frames. Not as good as Tfasty3.
`tfastx`	GCG package program in which a protein query sequence is compared to a nucleic acid database translated in all six frames **with frameshifts**, **between codons** Recommended over Tfasta3, but not as good as Tfasty3.
`tfastx`3	Fasta3 package program in which a protein query sequence is compared to a nucleic acid database translated in all six frames **with frameshifts**, **between codons** Recommended over tfasta or tfasta3, but not as good as Tfasty3.
`tfasty`3	Fasta3 package program in which a protein query sequence is compared to a nucleic acid database translated in all six frames **with frameshifts**, **within and between codons**. Recommended over Tfasta, Tfasta3, Tfastx and Tfastx3.
`fastx`	GCG package program in which a nucleic acid query sequence translated in all six frames **with frameshifts within codons** is compared to a protein database. Not as good as Fasty3.
`fastx`3	Fasta3 package program in which a nucleic acid query sequence translated in all six frames **with frameshifts within codons** is compared to a protein database. Not as good as Fasty3.

`fasty3`	Fasta3 package program in which a nucleic acid query sequence translated in all six frames **with frameshifts within and between codons** is compared to a protein database. Recommended over Fastx3.
`framesearch`	GCG program which compares a protein query sequence to a nucleic acid database translated in all six frames **with frameshifts** or a nucleic acid query sequence is compared to a protein database **with frameshifts using a full Smith-Waterman algorithm.** This is expensive. You might wish to use http://sgbcd.weizmann.ac.il/ or have me run it for you.

Criteria for statistical significance:

1. Search with filtered sequences. The NCBI Blast programs perform filtering automatically (BLASTP and TBLASTN filter with Seg, BLASTN filters query sequences with Dust, and BLASTX and TBLASTX filter translated sequences with Seg). Other search programs don't filter automatically. For all other programs, filter protein query sequences with Seg. Filter nucleic acid query sequences with Dust, when they are compared to nucleic acid databases.

2. For protein/protein, protein/nucleic acid, and nucleic acid protein searches sequences are homologous if $E \leq .001$ or .01. The cutoff for nucleic acid/nucleic acid searches is lower and has not as yet been established.

3. Align all putative homologous sequences with the full Smith-Waterman method. Filter the query and database sequence to see if their region of overlap is due to low complexity regions.

4. Test marginal hits further with PRSS3.

5. Statistics can be improved with an organism-specific BLAST search. If there are fewer than a 1000 sequences for an organism, see me. Likewise if you need to set up a database for an organism-specific search on cuccfa see me.

Suggested order of searches:

1. blastp protein/protein
2. fasta (or fasta3) protein/protein
3. ssearch3 protein/protein
4. blastn nucleic acid/nucleic acid
5. fasta (or fasta3) nucleic acid/nucleic acid
6. tblastn protein protein/nucleic acid
7. tfasty3 protein/nucleic acid
8. blastx nucleic acid/protein
9. fasty3 nucleic acid/protein

10. ssearch3 nucleic acid/nucleic acid
11. framesearch nucleic acid/protein
12. framesearch protein/nucleic acid

Lab:

I. The NCBI BLAST server:

(note you can do the following exercises using netblast, rather than the blast server, if you prefer).

O. Preparation:

Copy the protein file, human.pep, and the corresponding nucleic acid file (derived from an mRNA) human.n.seq from the directory /usr2/seq/seqclass/lab6 to a directory dedicated to lab 6. These sequences come to us courtesy of Professor Howard Lieberman of the Radiation Oncology Department. The following problems are geared to web-implementation of Blast,, but you can use the GCG program Netblast if you prefer.

Using Netscape Navigator, click on http://www.ncbi.nlm.nih.gov/BLAST/ .
Under BLAST 2.0 click on: Basic BLAST Search.

1. BLASTP search

A. Perform a BLASTP search on the SwissProt database using the human protein sequence as a query sequence (Hint: Convert your file to the proper format). Save the html file onto your diskette.

B. Examine the BLAST output:

1. Go back to Netscape.
2. How many hits are significant?
3. Click on the name of each of the best hits and save the resulting sequence file to your diskette.
4. What biological function does our unknown human protein perform?
5. What biochemical function does our unknown human protein have?

C. Perform a full Smith-Waterman alignment between the human sequence and the best database hit as follows:

1. Use netfetch to transfer the sequence to Cuccfa.
2. Use reformat to translate the sequence into GCG format.
3. Use Bestfit to align the sequences.

D. Compare the alignment given by Bestfit to the alignment given with the BLAST program.

1. If the two alignments are different, why?
2. What does this imply about the accuracy of BLAST2 alignments?
3. What should be done to improve upon BLAST2 alignments?
4. Further test the statistical significance of the match using PRSS. Is it a match?

E. Repeat the search against the nonredundant protein database and save the output as above.

1. Does the nonredundant database help you identify the biochemical function of the protein?
2. Do the same proteins have the same E() values? If not explain why?

2. BLASTN search

A. Compare the nucleic acid sequence to the nonredundant nucleic acid database using the BLASTN program.

1. What categories do the statistically significant hits fall into?
2. Note the reading sense of the phosphatase genes. Since the query nucleic acid sequence is based on mRNA, it represents the correct reading sense.
3. Is our protein a phosphatase?

3. TBLASTN search

Compare the protein sequence to the nonredundant nucleic acid database with TBLASTN. Does this search tell you anything new?

4. BLASTX search

Compare the nucleic acid sequence to the nonredundant protein database. Does this search tell you anything new?

5. TBLASTX search

Let's skip TBLASTX. This program compares a nucleic acids query sequence, translated in all six reading frames, to a nucleic acid database, translated in all six reading frames. This program accesses the EST and STS databases, but not databases of identified nucleic acids sequences. Hence it is not of interest to us here.

6. Single organism BLASTP search

Click on Advanced BLAST. Perform a BLASTP search against the C. Elegans database.

II. The GCG FastA program and Programs in the Fasta3 package.

1. Use of Seg and dust to mask low complexity regions:

Since the Fasta program in GCG and the programs in the Fasta3 packages do not mask low complexity regions you must use the program Seg to filter out low complexity regions in human.pep.

A. Filter the human protein sequence with the GCG Seg program by typing:

seg

I suggest that you designate the output file human.fd.pep (where "fd" stands for "filter default").

B. Filter the nucleic acid sequence by typing

```
tofasta human.n.seq –D

dust human.n.tfa > human.fn.tfa
```

Where *n* stands for "nucleic acid" and *fn* stands for "filter nucleic". Change the sequence name on the first line of the file to *human.fn .*

Then convert back to GCG format by typing:

```
fromfasta human.fn.tfa -D
```

2. Introduction to the Fasta Programs in GCG and the Fasta3 package.

There are two implementations of the Fasta series of programs on cuccfa: The programs in the GCG package, based on Fasta2, and the programs in William Pearson's Fasta3 package. Each implementation has its advantages and disadvantages, but a good rule-of-thumb is that the GCG Version is easier-to-use and more clearly documented, whereas the Fasta3 package is more powerful. We will use both packages, so that we have the best tools at our disposal.

3. Use of the Fasta3 to search the nonredundant protein database.

I update the NCBI nonredundant protein database on cuccfa weekly in fasta format, so that it is available for use with search methods that are more sensitive than BLAST2. This database is only searchable by the programs in the Fasta3 package, not by GCG. To use Fasta3 to search the nonredundant protein database type

```
fasta3
```

and answer the questions the program asks you.

Note: The exercises after this point are optional. The following exercises place such strains on cuccfa that if all the students in the lab were to run them at once they would prevent e-mail from getting through. I therefore request that you run them on your own, after lab, before class next week. Don't run more than one of these at once!

4. Use of the GCG implementation of Fasta to compare a nucleic acid query sequence to a nucleic acid database.

The GCG implementation of Fasta is an improvement over the Pearson version, in that it searches both nucleic acid strands of the database sequences at once. I therefore recommend it for nucleic acid-nucleic acid comparisons. Apply it with the following command line

```
fasta -nomon -in1=human.fn.seq -in2=ge:* -D &
```

Discuss any putative hits.

You can log off in the middle of a run and have the run complete if you put the above comand (without the "&" in an executable file and execute it with nohup. For example if you place the command

```
fasta -nomon -in1=human.fn.seq -in2=ge:* -D
```

in a file called runfasta, make runfasta executable with the command

```
chmod +x runfasta
```

and execute the command

```
nohup runfasta &
```

you can log off while the program is running.

5. Use of the Fasty3 program to compare a nucleic acid query sequence to a protein database.

Fasty3 compares a nucleic acid query sequence translated in all six frames to a protein database. Use Fasty3 to compare our sequence to Swissprot. Discuss the results.

```
fasty3 -q human.fn.tfa /usr2/seq/blast2/data/nr >
human.fn.fasty3  &
```

(this must be typed w/o carriage returns even if the text overflows a line)

6. Use of Tfasty3 to compare a protein query sequence to a nucleic acid database.

Tfasty3 compares a protein query sequence to a nucleic acid database translated in all six reading frames and taking frameshifts into account. Tfasty3 is an improvement over the Tfasta program discussed in my GCG tutorial. Use Tfasty3 to compare our protein sequence to the Genbank + EMBL w/o ESTs and STSs databases, by typing the following command:

```
tfasty3 -q human.fd.pep
@/usr2/seq/fasta/fastadata/ge.nam > human.fd.tfasty3 &
```

(this must be typed w/o typing a carriage return even if the text overflows a line)

Discuss the results.

7. Use of Ssearch3 to compare a protein sequence to a protein database:

Compare the protein sequence to the NCBI nonredundant protein database using Ssearch3, which gives full Smith-Waterman searches. The command you type is:

```
ssearch3 -q human.fd.pep /usr2/seq/blast2/data/nr >
human_fd. nr.ssearch3 &
```

(this must be typed w/o a carriage return even if the text overflows a line)

Web sites:

Links to the following sites::
http://www.ccc.columbia.edu/~friedman/lesson_6.html

On-line GCG Manual
http://www.ccc.columbia.edu/genhelp/

NCBI BLAST Server
http://www.ncbi.nlm.nih.gov/BLAST/

FASTA Programs at the U. of Virginia:
http://alpha10.bioch.virginia.edu/fasta/

Convert Sequence Formats Using ReadSeq
http://dot.imgen.bcm.tmc.edu:9331/seq-util/readseq.html

Baylor Protein Database Search Launcher
http://dot.imgen.bcm.tmc.edu:9331/seq-search/protein-search.html

Baylor Species-Specific Protein Database Search Launcher
http://dot.imgen.bcm.tmc.edu:9331/seq-search/protein-search.html

Baylor Nucleic Acid Database Search Launcher
http://dot.imgen.bcm.tmc.edu:9331/seq-search/nucleic_acid-search.html

Baylor Sequence Utilities Launcher
http://dot.imgen.bcm.tmc.edu:9331/seq-util/seq-util.html

Six Frame Translation
http://dot.imgen.bcm.tmc.edu:9331/seq-util/Options/sixframe.html

Reverse Complement
http://dot.imgen.bcm.tmc.edu:9331/seq-util/Options/revcomp.html

Weizmann institute server
http://sgbcd.weizmann.ac.il/

Bibliography

1. http://www.ncbi.nlm.nih.gov/BLAST/newblast.html
2. GCG program manual entries on Blast, Netblast, Fasta, Tfasta, Seg, and Fromgenbank.
3. /usr2/seq/doc/fasta3intro .
4. Pearson, W.R, and Lipman, PNAS, 85, 2444-2448, (1998)
5. Altschul, et al., J. Mol. Biol., 215, 403-410 (1990)
6. Altschul et al.., Nuc. Acids. Res., 25, 3389- 3402.

Written problems (optional)

1. A Blast2 search is performed with a protein query sequence 100 residues long against the Swissprot database (26 million residues) and the nonredundant database (133 million residues). A database sequence is hit with a score of 103. How many sequences do you expect to have the same or higher score in each database? For the matrix used, K=.082 and λ=.219 .

2. Compute the entropy of the word "isaac" using a 26 letter alphabet and a 5 letter window.

Key to homework from lesson 5.

Needleman-Wunsch-Gotoh global alignment (final paths indicated in bold):

		T	C	C	G	G	T
	0	-2	-4	-6	-8	-10	-12
T	-2	2	0	-2	-4	-6	-8
C	-4	0	4	2	0	-2	-4
G	-6	-2	2	3	4	2	0
G	-8	-4	0	1	5	6	4
C	-10	-6	-2	-1	3	4	5
		T T	C _	C C	G G	G G	T C
		T T	C C	C _	G G	G G	T C

Smith-Waterman local alignment (final path indicated in bold):

		T	C	C	G	G	T
	0	0	0	0	0	0	
T	0	2	0	0	0	0	
C	0	0	4	2	0	0	0
G	0	0	2	3	4	2	0
G	0	0	0	1	5	6	4
C	0	0	0	0	3	4	5
		T T	C C	C _	G G	G G	