MCB 5472

Perl: scalars, STDIN

Databanks, Blast homology

J. Peter Gogarten
Office: *BPB 404*
phone: *860 486-4061,*
Email: *gogarten@uconn.edu*

Assignment for Today

1) On the computer that you plan to use for your project set up a connection (or connections) to bbcxsrv1 and bbcsrv3 that allows you
(a) ssh to the server using a command line interface
(b) allows you to drop and drag files from your computer to the server.

2) check that your vi editor on bbcxsrv1 is set up to have context dependent coloring (do this, even if you don't plan to use vi on the server!).

3) if you do not want to use vi, install an editor on your computer that provides context dependent coloring.

[4) Read through U1-U26 of the Unix and Perl Primer for Biologists (available at the Korfflab at http://korflab.ucdavis.edu/Unix_and_Perl/]
Read through pages 53-61 of the Unix and Perl Primer for Biologists

5) Read U35 and http://kb.iu.edu/data/abdb.html on the chmod command in unix

6) Create first Perl Program- "Hello, world!" [make file executable using chmod]

```
#!/usr/bin/perl -w
print ("Hello, world! \n");
```

What happens if you leave out the new line character?
You can run the program by typing ./program_name.pl, if the file containing the program is made executable (using chmod u+x *.pl).

For next Monday:

1) What is the difference between a compiler and an interpreter?
2) When is it useful to make a script executable, when not?
3) What is the value of $i after each of the following operations?

```
$1=1;
$i++;
$i *= $i;
$i .= $i;
$i = $i/11;
$i = $i . “score and” . $i+3;
```

First make a guess, then test your prediction using a script.

4) If $a = 2 and $b=3, what is the type and values of the scalar stored in $c after each of the following statements:

```
$c = $a + $b;
$c = $a / $b;
$c = $a . $b;
$c = "$a + $b";
$c = '$a + $b’;
```

First make a guess, then test your prediction using a script.

5) Do “Hello world” example (class 1) using a variable!

6) Write a short Perl script that calculates the circumference of a circle given a radius provided by the user. (For inspiration see exercises 1-4 chapter 2 in Learning Perl). (One set of answers is given in Appendix A of the book)

send your homework to gogarten@uconn.edu.

vi

A vi tutorial is at http://www.eng.hawaii.edu/Tutor/vi.html -- however, if you run into problems google usually helps (e.g. google: vi replace unix gives you many pages of info on how to replace one string with another under vi)

`vi myprogram.pl` #starts the editor and loads the file myprogram.pl into the editor

The following should get you started:
The arrow keys move the cursor in the text
(if you have a really dumb terminal you can use the letter hjkl to move the cursor)

`x` deletes the character under the cursoresc (i.e. the escape key) leaves the edit modei enters the edit mode and inserts before the cursora enters the edit mode and appends
`esc :` opens a command line (here you can start searches, and replacements)
`:w` #saves the file
`:w` new_name _of_file #writes the file into a new file.
`:wq` #saves the file and exits vi
`:q!` #exits vi without saving

customizing vi

One of the beauties of vi is that usually it provides context dependent coloring.
You need to tell vi which terminal you use.
One way to do so is to add a file called .vimrc to your home directory.

The following works under both, MAS OSX and using ssh via the secure shell program under windows:
`vi .vimrc` #opens vi to edit .vimrc (Files that start with a dot are not listed if you list a directory. List with `ls -a`)
`set term=xterm-color` #tells the editor that you use a terminal that conforms to some standard
`syn on` # tells the editor program that you want to use syntax dependent coloring.
`esc:wq`

This might seem a little inconvenient, but it really comes in handy to trouble shoot the program in the same environment where you want to run it.
(comment on textwrangler alternative, ssh is included inside the program)

PERL conventions and rules

Basic Perl Punctuation:
line ends with ";"
empty lines in program are ignored
comments start with #
first line points to path to interpreter:

```
#! /usr/bin/perl

# "#!" is known as "shebang";
```

keep one command per line for readability
For shell scripts the first line would refer to the type of shell to be used, e.g.:

```
#!/bin/bash
```

use indentation do show program blocks.
Variables start with $calars, @rrays, or %ashes
Scalars: floating point numbers, integers,
non decimal integers, strings

Scalar variable are placeholders that can be assigned a scalar value (either number or string).
Scalar variables begin with $

```
$n=3; #assigns the numerical value 3 to the variable $n.
#Variables are interpolated, for example if you print text

$b = 4 + ($a = 3); # assign 3 to $a, then add 4 to that
# resulting in $b getting 7
$d = ($c = 5); # copy 5 into $c, and then also into $d
$d = $c = 5; # the same thing without parentheses

$a = $a + 5; # without the binary assignment operator
$a += 5; # with the binary assignment operator

$str = $str . " "; # append a space to $str
$str .= " "; # same thing with assignment operator

"hello" . "world" # same as "helloworld"
'hello world' . "\n" # same as "hello world\n"
"fred" . " " . "barney" # same as "fred barney"
"fred" x 3 # is "fredfredfred"
"barney" x (4+1) # is "barney" x 5, or # "barneybarney……"
(3+2) x 4 # is 5 x 4, or really "5" x 4, which is ”5555”
```

Note: these are not mathematical equations but assignments!

Numbers can be manipulated using the typical symbols:

```
2 + 3 # 2 plus 3, or 5
5.1 - 2.4 # 5.1 minus 2.4, or approximately 2.7;
3 * 12 # 3 times 12 = 36;
2**3 # 2 taken to the third power = 2*2*2 = 8
14 / 2 # 14 divided by 2, or 7;
10.2 / 0.3 # 10.2 divided by 0.3, or approximately 34;
10 / 3 # always floating point divide, so approximately 3.3333333...
```

Special characters:

```
\n #newline
\t #tab
```

Double quoted strings are interpolated by the Perl interpreter:

```
"hello world\n" # hello world, and a newline
"coke\tsprite" # a coke, a tab, and a sprite
```

The backslash can precede many different characters to mean different things (typically called a backslash escape).

Variable interpolation - single quoted strings are not interpolated:

```
'hello' # five characters: h, e, l, l, o
'don\'t' # five characters: d, o, n, single-quote, t
'' # the null string (no characters)
'silly\\me' # silly, followed by backslash, followed by me
'hello\n' # hello followed by backslash followed by n
'hello
there' # hello, newline, there (11 characters total)
```

Example

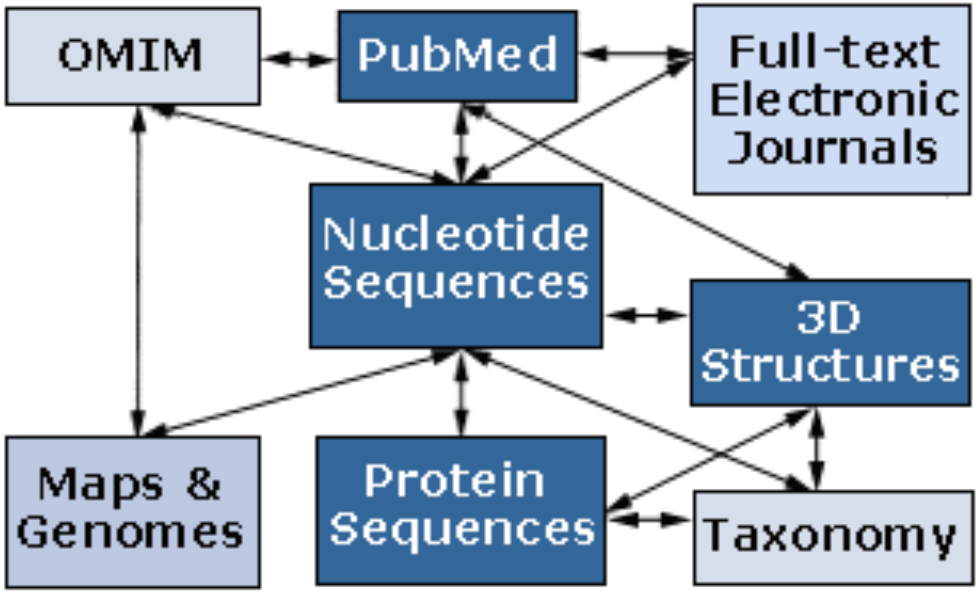
```
node011:~/perl2012/class02 jpgogarten$ vi demo.pl

#!/usr/bin/perl
use warnings;
use strict;
print "Enter a number:\n" ;
chomp(my $input = <STDIN>);
my $squared=$input**2;
print "the input squared is $squared\n";
```

Go through class2.pl
http://gogarten.uconn.edu/mcb5472_2012/class2.pl

Databanks (A)

Entrez
search and retrieval system

NCBI (National Center for Biotechnology Information) is a home for many public biological databases (see an older diagram below). All of the databases are interlinked, and they all have common search and retrieval system - **Entrez**.



A listing of databases in ENTRZ is here.

Entrez / Pubmed, continued

- An interactive Pubmed tutorial click here.
- An Entrez tutorial (non interactive) is here
- Use Boolean operators (**AND**, **OR**, **NOT**) to perform advanced searches. Here is an explanation of the Boolean operators from the Library of Congress Help Page.

- Explore features of Entrez interface: **Limits**, **Index**, **History**, and **Clipboard**.

Other Literature databanks and Services

While Pubmed is incorporating more and more non-medical literature, there might still be gaps in the coverage. Alternatives are local services offered at the UConn libraries. http://rdl.lib.uconn.edu/byTitle.php

The "Web of Science" database gives access to the Science Citation Index: a database that tracks cited references in journals. Note that these resources are restricted to UConn domain, so you either need to access it from a campus computer or through a proxy account.

However, the new Google Scholar system is about as complete as the fee for service sites (check here)

Search Robots

PubCrawler allows to run predefined literature searches. Results are written into a database and you are send an email, if there were new results. NCBI now offers a similar service (see My NCBI (Chubby), check the tutorial).

Swiss-Shop is offering the same service for proteins

Sequence and structure databanks

can be divided into many different categories.
One of the most important is

Supervised databanks with gatekeeper. Examples:

Swissprot
Refseq (at NCBI)

Entries are checked for accuracy.
+ more reliable annotations
-- frequently out of date

Repositories without gatekeeper. Examples:

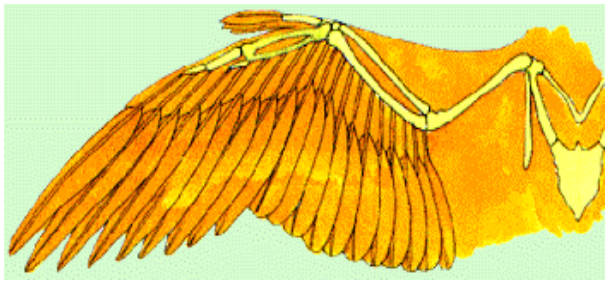
GenBank
EMBL
TrEMBL

Everything is accepted
+ everything is availabel
-- many duplicates
-- poor reliability of annotations

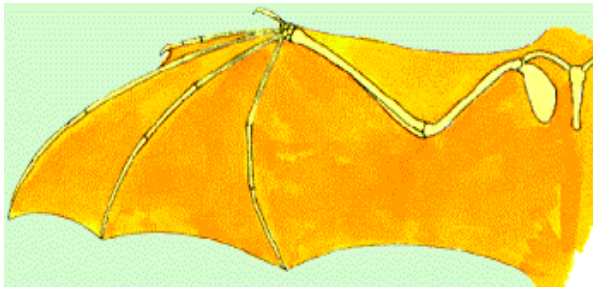
Theodosius Dobzhansky:

"Nothing in biology makes sense except in the light of evolution"

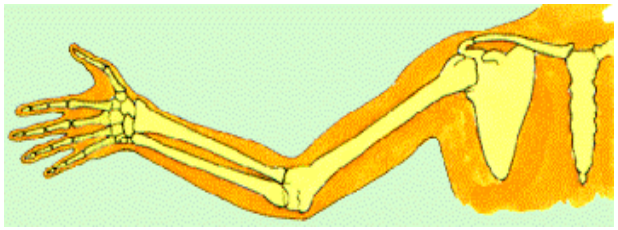
Homology



bird wing



bat wing



human arm

by Bob Friedman

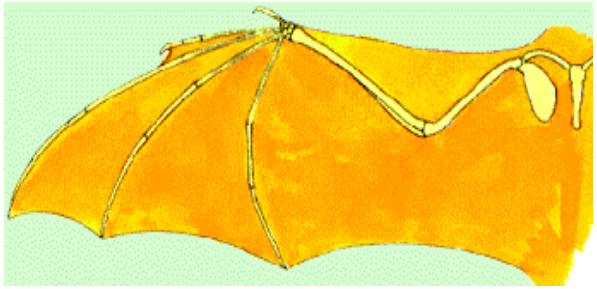
homology *vs* analogy

A priori sequences could be similar due to convergent evolution

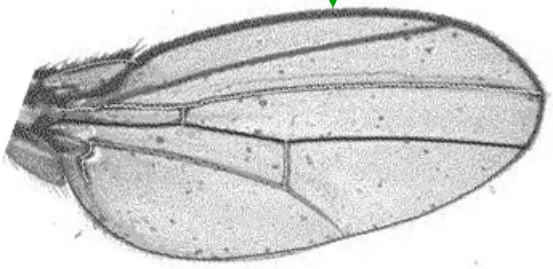
Homology (shared ancestry) *versus* Analogy (convergent evolution)

bird wing

butterfly wing



bat wing



fly wing

Related proteins

Present day proteins evolved through substitution and selection from ancestral proteins.

Related proteins have similar sequence AND similar structure AND similar function.

In the above mantra "similar function" can refer to:

- identical function,
- similar function, e.g.:
  - identical reactions catalyzed in different organisms; or
  - same catalytic mechanism but different substrate (malic and lactic acid dehydrogenases);
  - similar subunits and domains that are brought together through a (hypothetical) process called domain shuffling, e.g. nucleotide binding domains in hexokinse, myosin, HSP70, and ATPsynthases.

homology

Two sequences are homologous, if there existed an ancestral molecule in the past that is ancestral to both of the sequences

Homology is a "yes" or "no" character (don't know is also possible). Either sequences (or characters share ancestry or they don't (like pregnancy). Molecular biologist often use homology as synonymous with similarity of percent identity. One often reads: sequence A and B are 70% homologous. To an evolutionary biologist this sounds as wrong as 70% pregnant.

Types of Homology

Orthology: bifurcation in molecular tree reflects speciation

Paralogy: bifurcation in molecular tree reflects gene duplication

no similarity *vs* no homology

If two (complex) sequences show significant similarity in their primary sequence, they have shared ancestry, and probably similar function.

THE REVERSE IS NOT TRUE:

PROTEINS WITH THE **SAME OR SIMILAR FUNCTION DO NOT ALWAYS SHOW SIGNIFICANT SEQUENCE SIMILARITY**
for one of two reasons:

a) they evolved independently
(e.g. different types of nucleotide binding sites);
or
b) they underwent so many substitution events that there is no readily detectable similarity remaining.

Corollary: PROTEINS WITH SHARED ANCESTRY DO NOT ALWAYS SHOW SIGNIFICANT SIMILARITY.

homology

Two sequences are homologous, if there existed an ancestral molecule in the past that is ancestral to both of the sequences

Types of Homology

Orthologs: "deepest" bifurcation in molecular tree reflects speciation.
These are the molecules people interested in the taxonomic classification of organisms want to study.

Paralogs: "deepest" bifurcation in molecular tree reflects gene duplication. The study of paralogs and their distribution in genomes provides clues on the way genomes evolved. Gen and genome duplication have emerged as the most important pathway to molecular innovation, including the evolution of developmental pathways.

Xenologs: gene was obtained by organism through horizontal transfer. The classic example for Xenologs are antibiotic resistance genes, but the history of many other molecules also fits into this category: inteins, selfsplicing introns, transposable elements, ion pumps, other transporters,

Synologs: genes ended up in one organism through fusion of lineages. The paradigm are genes that were transferred into the eukaryotic cell together with the endosymbionts that evolved into mitochondria and plastids
(the -logs are often spelled with "ue" like in orthologues)

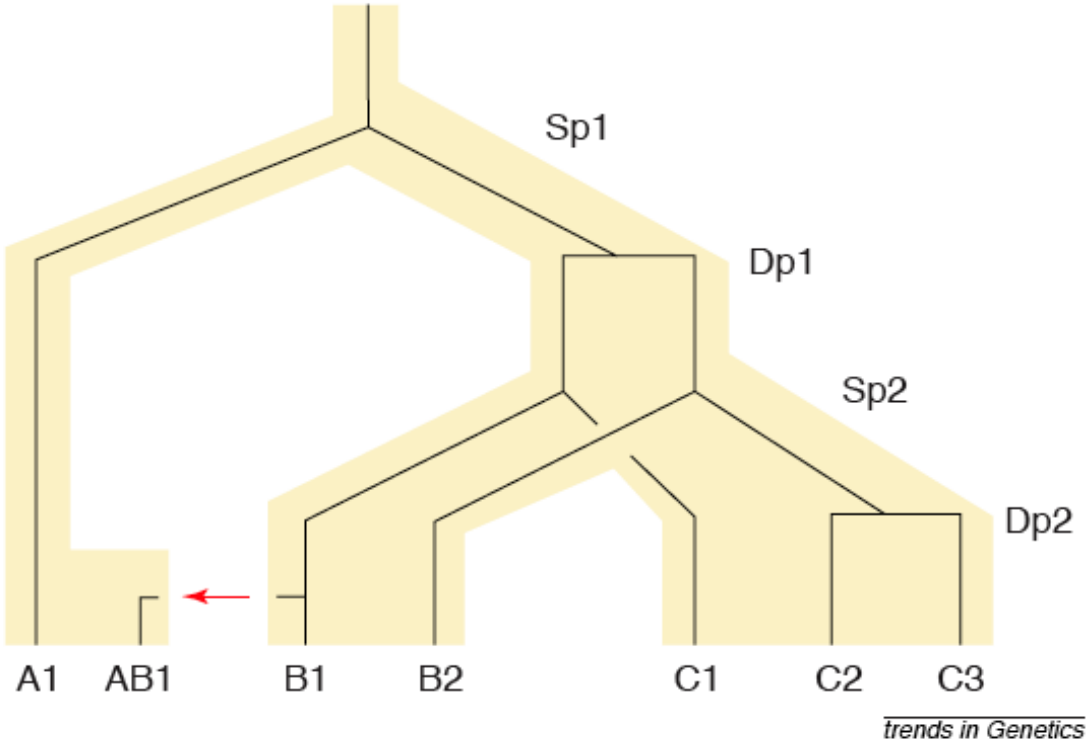
see Fitch's article in TIG 2000 for more discussion.

Homologs, orthologs, and paralogs

- Homologous structures or characters evolved from the same ancestral structure or character that *existed in some organism in the past*.
- Orthologous characters present in two organism (A and B) are homologs that are derived from a structure *that existed in the most recent common ancestor* (MRCA) of A and B (orthologs often have the same function, but this is NOT part of the definition; e.g. human arms, wings or birds and bats).
- Paralogous characters in the same or in two different organisms are homologs that are not derived from the same character in the MRCA, rather they are *related* (at their deepest node) *by a gene duplication event*.

Examples

FIGURE 1. Orthology, paralogy and xenology



B1 is an ortholog to C1 and to A1
C2 is a paralog to C3 and to B1;
BUT
A1 is an ortholog to both B1, B2,and to C1, C2, and C3

From: Walter Fitch (2000): *Homology: a personal view on some of the problems*, TIG 16 (5) 227-231

Uses of Blast in bioinformatics

The Blast web tool at NCBI is limited:

- custom and multiple databases are not available
- tBlastN (gene prediction) not available
- “time-out” before long searches are completed

What if researcher wants to use tBlastN to find all olfactory receptors in the mosquito? Or, if you want to check the presence of a (pseudo)gene in a preliminary genome assembly?
Answer: Use Blast from command-line

Also: The command-line allows the user to run commands repeatedly

Types of Blast searching

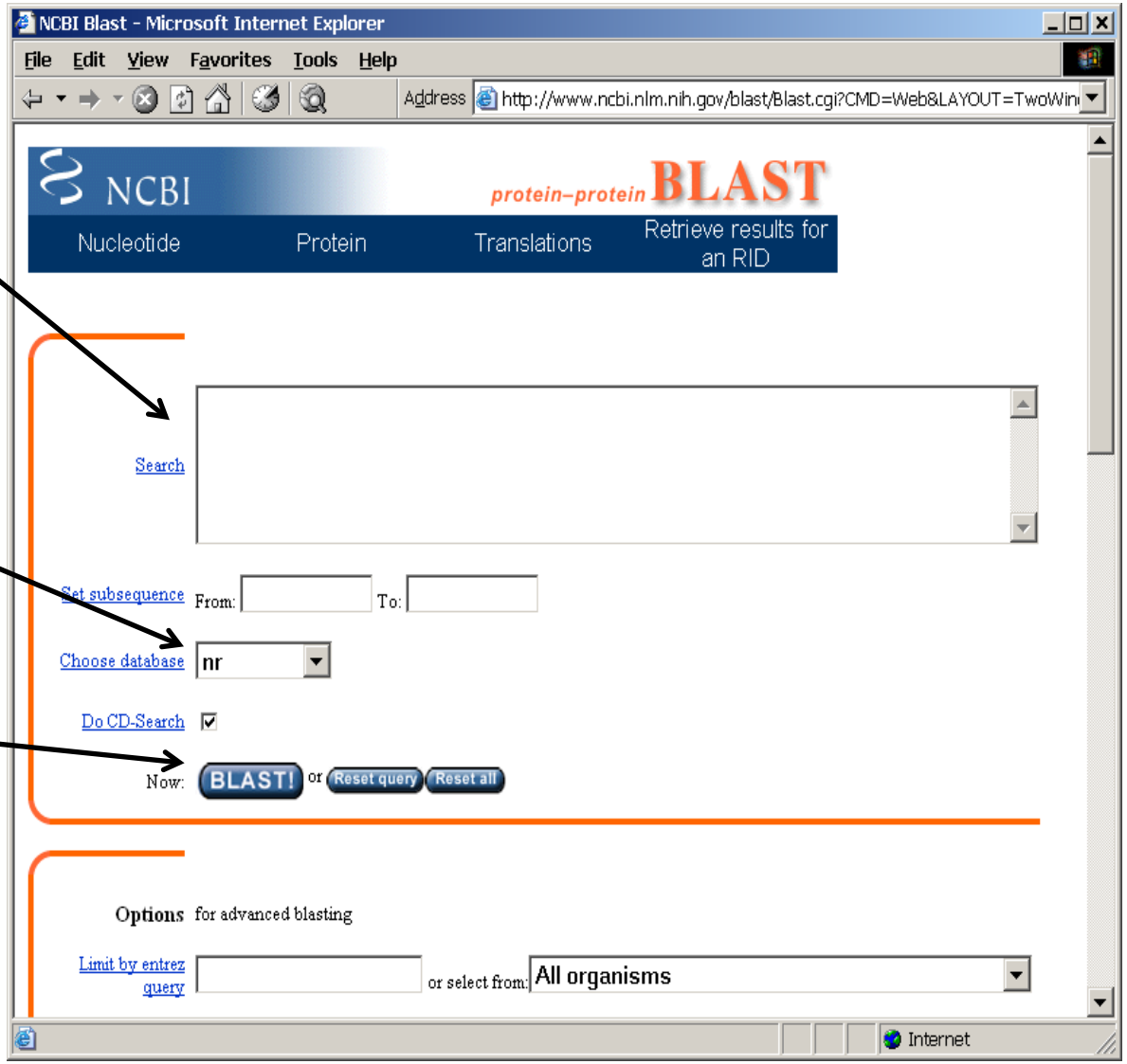
- blastp compares an amino acid query sequence against a protein sequence database
- blastn compares a nucleotide query sequence against a nucleotide sequence database
- blastx compares the six-frame conceptual protein translation products of a nucleotide query sequence against a protein sequence database
- tblastn compares a protein query sequence against a nucleotide sequence database translated in six reading frames
- tblastx compares the six-frame translations of a nucleotide query sequence against the six-frame translations of a nucleotide sequence database.

Routine BlastP search

FASTA formatted text
or Genbank ID#

Protein database

Run



BlastP parameters

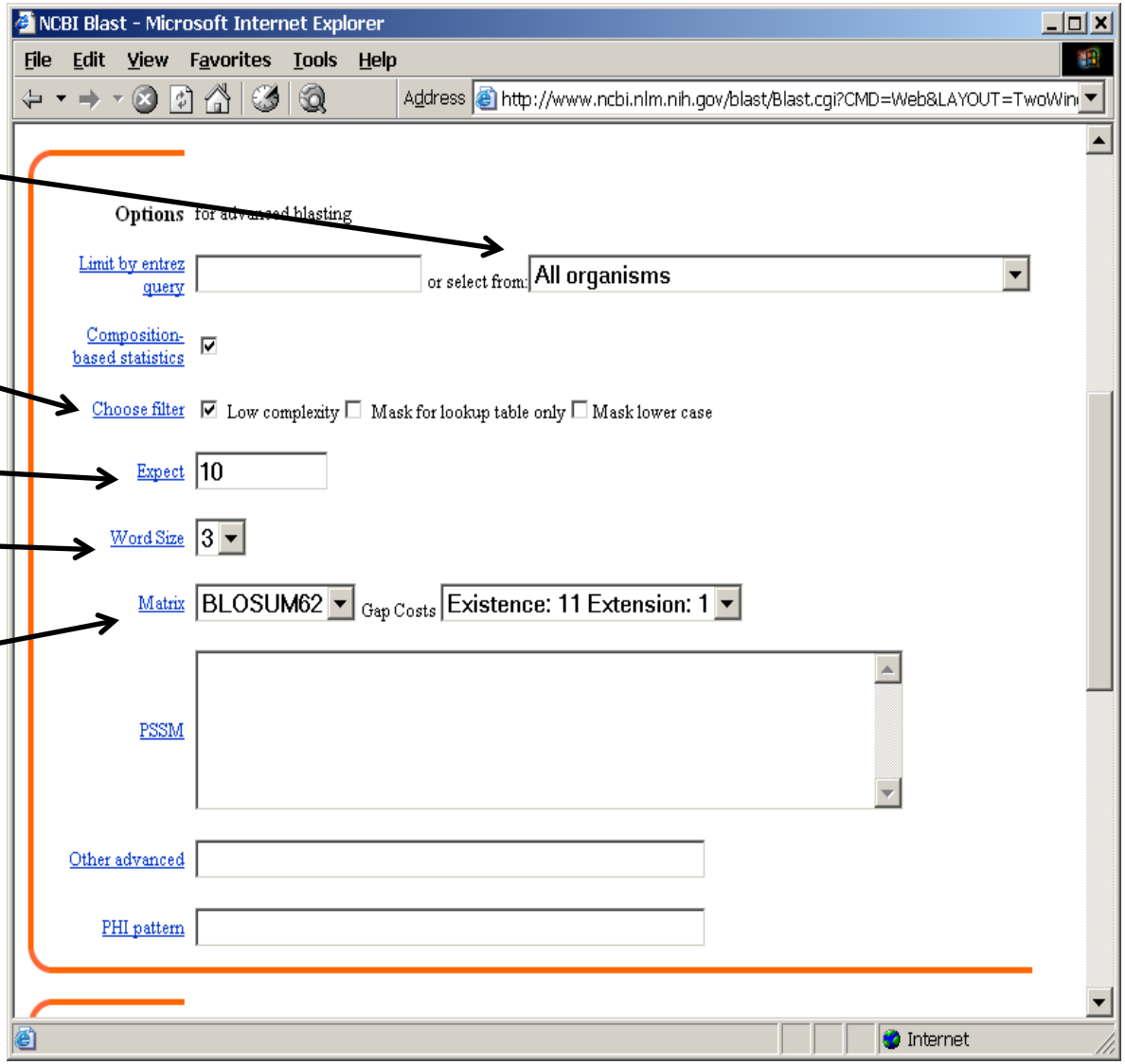
Restrict by taxonomic group

Filter repetitive regions

Statistical cut-off

Size of words in look-up table

Similarity matrix (cost of gaps)



by Bob Friedman

Establishing a significant “hit”

Blast’s E-value indicates statistical significance of a sequence match

Karlin S, Altschul SF (1990) Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. PNAS 87:2264-8

E-value is the Expected number of sequence (HSPs) matches in database of *n* number of sequences

- database size is arbitrary
- multiple testing problem
- E-value calculated from many assumptions
- so, E-value is not easily compared between searches of different databases

Examples:

E-value = 1 = expect the match to occur in the database by chance 1x

E-value = .05 = expect 5% chance of match occurring

E-value = $1x10^{-20}$ = strict match between protein domains

When are two sequences significantly similar? PRSS

One way to quantify the similarity between two sequences is to

1. compare the actual sequences and calculate an alignment score
2. randomize (scramble) one (or both) of the sequences and calculate the alignment score for the randomized sequences.
3. repeat step 2 at least 100 times
4. describe distribution of randomized alignment scores
5. do a statistical test to determine if the score obtained for the real sequences is significantly better than the score for the randomized sequences

z-values give the distance between the actual alignment score and the mean of the scores for the randomized sequences expressed as multiples of the standard deviation calculated for the randomized scores.

For example: a z-value of 3 means that the actual alignment score is 3 standard deviations better than the average for the randomized sequences. z-values > 3 are usually considered as suggestive of homology, z-values > 5 are considered as sufficient demonstration.

E-values and significance

Usually E values larger than 0.0001 are not considered as demonstration of homology.

For small values the E value gives the probability to find a match of this quality in a search of a databank of the same size by chance alone.

E-values give the expected number of matches with an alignment score this good or better,
P-values give the probability of to find a match of this quality or better.
P values are [0,1], E-values are [0,infinity).
For small values E=P

Problem: If you do 1000 blast searches, you expect one match due to chance with a P-value of 0.0001

"One should" use a correction for multiple tests, like the Bonferroni correction.