MCB 372

BLAST, unix, Perl
continued

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

Discussion of Old Assignment

1) On the computer that you plan to use for your project set up a connection (or connections) to bbcxsrv1 that allows you
(a) ssh to the server using a commandline 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) Create first Perl Program- "Hello, world!" [make file executable using chmod 755 *.pl]
#!/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 an executable.

5) Read chapter 1 of "Learning Perl". HERE (pay attention on pages 12-15)

Command Line

The favored operating system flavor in computational biology is UNIX/LINUX.
The command line is similar to DOS.
Some of the frequently used commands are here

```
pwd                    ps
ls                     ps aux
ls -l                  rm
chmod                  more
chmod a+x blastall.sh  cat
chmod 755 *.sh         vi (text editor)
cd                     ps
cd ..                  ps aux
cd $HOME               ssh
passwd                 sftp
```

For windows an "ok" ssh program is putty.
UConn also has a site license for the ssh program from ssh.com

UNIX

Basic UNIX commands
ls, cd, chmod, cp, rm, mkdir, more (or) less, vi, ps, kill −9, man
A brief listing is here

chmod is a particular pain in the ... .
Under unix every file has an owner and the owner, his group and everyone else have permissions to read, write and/or execute the file (or they don't). If you want to see which permissions are currently assigned to your files, type ls -l at the command prompt.
chmod a+x *.pl gives everyone execute permission for all files that end with .pl the * is a wildcard. (warning don't ever use rm in conjunction with *)
For more on chmod type "`man chmod`" or see here.
(In the OSX GUI you can control click at a file, and change permissions in the info box). Most ssh clients (FUGU and SSH) allow you to use a GUI to change file permissions (in FUGU ctrl click).

Unix - command line interface

If you tried to execute a command, and you made a mistake, for example, you mistyped a file name, you can recall the last command using the up arrow (down arrow for more recent).

If you are tired typing long filenames, you can use the tab key to complete the line, provided there is only one way to complete the line. E.g: cd /Desktop could be replaced by cd /D<tab>
If there are two or more choices you hear a boing, if you hit <tab> again, you get a list of choices.

writing Perl scripts

Use unix/ linux /OsX if possible (talk with Tim if you want to use windows).
A) open a terminal window ; type "which perl <return>"
B) SSH to a unix machine (cluster@bbcxrv1), log in, type "which perl <return>"
C) to check the version type perl -v <return>The response of the system should tell you, where Perl is installed on your machine (you need to know this for the first line of your perl program, which tells the operating system how to interpret what follows. On most installations this is #!/usr/bin/perl).
WINDOWS: If you use a windows machine, you can use an ssh program to connect to the biotech cluster. A good ssh client is available at ftp://ftp.ssh.com/pub/ssh/- highly recommended. A reasonable text editor is available at http://www.context.cx/
MAC OsX: If you use a Mac under OS X, and you do not want to (only) use the PERL locally, you want to install both jellyfish (ssh terminal) and fugu (a secure file transfer program). Both are available at ftp://ftp.uconn.edu/pub/packages/ssh/mac/ or through the people who wrote the software - GOOGLE) Also, the bbcxsrv1 is available as a server using ssh or apl. You can connect to to it from the finder menu (-> GO -> Connect to Server) pasting the following into the menu box afp://bbcxsrv1.biotech.uconn.edu (select your account).
LINUX: Most editors on linux systems recognize Perl programs and provide context dependent coloring. Ssh and Konquerer work well for file transfer.

characters at the end of lines

File tranfers from Windows to UNIX and return:
End of Line characters are a problem. Under Windows DO NOT use notepad, it does not understand UNIX newline symbols '\n'.
Best write your programs under UNIX using vi or vim (or any other editor you are comfortable with)
2nd best is to use a text editor like textwrangler (very nice and free program for UNIX). Like vi and vim it provides context dependent coloring.
3rd best is to remove end of line symbols in a UNIX editor or use sed (Stream EDitor) after you transferred the file:

```
sed s/.$// name_of_WINDOWS_infile > name_of_UNIX_outfile
```

(This replases the last non letter character before the eol ($) with nothing)

Some versions of office allow to change files as UNIX textfiles, but ...

A related problem is encountered by Mac users. Most text editors will use MAC carriage returns at the end of the line. Most unix programs will not be able to handle these. In a terminal window you could use the following command to convert your file:
tr '\r' '\n' < name_of_the_Mac_file > name_of_the_unix_file
If you are working in a GUI environment, you also could use the convertNewLines.app program (install it in your application folder, drag the file you want to convert into the icon). The program is available here. This is very inconvenient, but there really is no easy solution, tough luck; and you better know about this incase something goes wrong.

vi

A short introduction to vi is at http://goforit.unk.edu/unix/unix11.htm -- 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 grogram)

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
use indentation do show program blocks.
Variables start with $calars, @rrays, or %ashes
Scalars: foating 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: '=' Does not denote a 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
"new \177" # new, space, and the delete character (octal 177)
"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)
```

Do "Hello world" example with variable (class 1)!

Example demo SSH to bbcxsrv.biotech.uconn.edu

A) Move files to bbcxsrv p_abyssi.faa and t_maritima.faa (using ssh or enter afp://bbcxsrv.biotech.uconn.edu in finder -> Go -> connect to server)

(check options for blastall and formatdb)

```
formatdb -i p_abyssi.faa -o T -p T
blastall -i t_maritima.faa -d p_abyssi.faa -o
  blast.out -p blastp -e 10 -m 8 -a2

./extract_lines.pl blast.out
```

Perl script that only retains the first hit and gets rid of comment lines
sftp results
    load into spreadsheet
    sort data, do histogram …
the extract_lines.pl script is here (you can sftp it into your account, you'll need to chmod 755 extr*.pl afterwards)

vi blast.out, extract_lines.pl, check spreadsheet
do histogram example, discuss % id, #id residues, alignment lengths.

Assignment for Wednesday

1) Read through the Perl scripts extract_lines.pl and extract_lines_mod.pl
2) Why does the first of these get along without `chomp ($line);`
3) Write a short Perl script that calculates the circumference of a circle given a radius provided by the user (see exercises 1-4 chapter 2 in Learning Perl). (One set of answers is given in Appendix A of the book)

Psi-Blast: Detecting structural homologs

Psi-Blast was designed to detect homology for highly divergent amino acid sequences

Psi = position-specific iterated

Psi-Blast is a good technique to find "potential candidate" genes

Example: Search for Olfactory Receptor genes in Mosquito genome
Hill CA, Fox AN, Pitts RJ, Kent LB, Tan PL, Chrystal MA, Cravchik A, Collins FH, Robertson HM, Zwiebel LJ (2002) G protein-coupled receptors in Anopheles gambiae. Science 298:176-8

by Bob Friedman

Psi-Blast Model

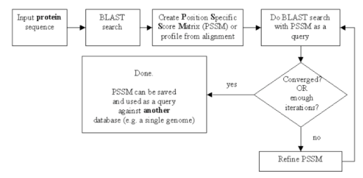
Model of Psi-Blast:
1. Use results of gapped BlastP query to construct a multiple sequence alignment
2. Construct a position-specific scoring matrix from the alignment
3. Search database with alignment instead of query sequence
4. Add matches to alignment and repeat

Similar to Blast, the E-value in Psi-Blast is important in establishing matches
E-value defaults to 0.001 & Blosom62

Psi-Blast can use existing multiple alignment - particularly powerful when the gene functions are known (prior knowledge) or use RPS-Blast database

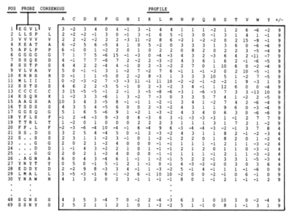
by Bob Friedman

PSI BLAST scheme


Input protein sequence
BLAST search
Create Position Specific Score Matrix (PSSM) or profile from alignment
Do BLAST search with PSSM as a query
Converged? OR enough iterations?
yes
no
Done.
PSSM can be saved and used as a query against another database (e.g. a single genome)
Refine PSSM

© Olga Zharybayeva

Position-specific Matrix



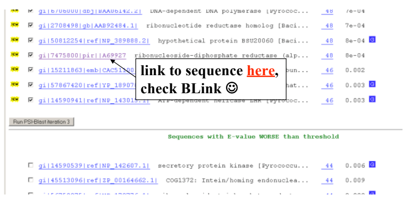
POS	PROBE	CONSENSUS	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y	+/-
1	E G V L	V	3	-2	3	4	0	4	-1	3	-1	4	4	1	1	1	-2	1	2	6	-6	-2	9
2	L L S P	L	2	-2	-2	-1	3	0	-1	3	-1	6	5	-1	3	0	-1	3	1	4	1	-1	9
3	V V V V	V	2	2	-2	-2	2	2	-3	11	-2	8	6	-2	1	-2	-2	0	2	15	-9	-1	9
4	K E A T	A	6	-2	5	6	-5	4	1	0	5	-2	0	3	3	3	1	3	6	0	-6	-4	9
5	A P L P	P	6	-1	0	1	-2	2	0	1	0	2	2	0	8	2	0	2	2	3	-5	-4	9
6	G G G G	G	7	1	7	5	-6	15	-1	-3	0	-4	-3	4	3	2	-3	6	4	2	-11	-7	9
7	S S Q E	D	4	-1	7	7	-6	7	2	-2	2	-3	-2	4	3	6	1	6	2	-1	-6	-5	9
8	S S T P	S	4	4	2	2	-4	4	-1	0	2	-3	-2	2	7	0	1	10	6	0	-2	-4	9
9	V L V A	V	5	0	-1	-1	3	1	-2	7	-2	7	6	-1	1	-1	-3	0	2	10	-5	-1	9
10	K R R S	R	0	-1	1	1	-5	0	2	-2	8	-3	1	3	3	3	10	5	1	-2	7	-5	9
11	M L I I	I	0	-2	-3	-2	7	-3	-3	11	-1	11	10	-2	-2	-1	-2	-2	1	9	-3	1	9
12	S S T S	S	4	6	2	2	-3	5	-1	0	2	-3	-2	3	4	-1	1	12	6	0	0	-4	9
13	C C C C	C	3	15	-5	-5	-1	2	-1	3	-5	-8	-6	-3	1	-6	-3	7	3	3	-13	10	9
14	K S Q R	K	1	-2	3	3	-6	1	3	-2	7	-3	0	3	3	5	7	4	1	-2	2	-5	9
15	A A G S	A	10	3	4	3	-5	8	-1	-1	1	-2	-1	3	4	1	-2	7	4	2	-6	-4	9
16	T S D S	S	4	3	5	4	-5	6	0	0	2	-3	-2	4	3	1	1	9	6	0	-3	-4	9
17	G G S Q	G	5	1	6	5	-6	9	1	-2	1	-3	-2	4	3	4	0	6	3	0	-6	-6	9
18	Y F L S	F	-1	2	-4	-3	9	-3	0	4	-3	6	3	-1	-3	-3	-3	1	-1	2	7	7	9
19	T T R L	T	1	-2	0	1	0	0	0	2	2	2	3	1	1	1	3	1	7	2	1	-2	9
20	F F . L	F	-2	-3	-6	-4	10	-4	-1	6	-4	9	6	-3	-4	-4	-3	-2	-1	3	7	8	4
21	S S . D	S	3	2	5	4	-4	5	0	-1	2	-3	-2	4	3	1	1	8	2	-1	-2	-3	4
22	S . . S	S	2	3	1	1	-2	3	-1	0	1	-2	-1	2	2	0	1	8	2	0	1	-2	4
23	. . . G	G	2	0	2	1	-2	4	0	0	0	-1	-1	1	1	1	-1	2	1	1	-3	-2	4
24	. . . D	D	1	-1	4	3	-2	2	1	0	1	-1	-1	2	1	2	0	1	1	0	-3	-1	4
25	. . . G	G	2	0	2	1	-2	4	0	0	0	-1	-1	1	1	1	-1	2	1	1	-3	-2	4
26	. A G N	A	6	0	4	3	-4	6	1	-1	1	-2	-1	5	2	2	-1	3	3	1	-5	-3	4
27	Y N Y T	Y	0	5	0	-1	5	-1	2	1	-1	0	-1	4	-3	-2	-2	0	3	0	3	6	4
28	E D D Y	D	2	-2	9	8	-3	3	4	-1	1	-3	-2	5	-1	4	-1	1	1	-1	-6	0	9
29	L M A L	L	3	-5	-3	-1	6	-1	-2	6	-1	10	10	-2	0	0	-2	-1	0	6	-1	0	9
30	Y N A W	N	4	1	3	2	0	2	3	-1	1	-1	-1	8	0	1	-1	2	1	-1	-1	2	9
.	.												.										
.	.												.										
.	.												.										
48	S G N S	S	4	3	5	3	-4	7	0	-2	2	-4	-3	6	3	1	0	10	3	0	-2	-4	9
49	S S N Y	S	2	5	2	1	1	2	1	0	1	-2	-2	5	1	-1	0	8	1	-1	3	1	9

FIG. 1. The concept of a profile. (*a*) A flow diagram of profile analysis. (*b*) A 49-residue sample profile for the immunoglobulin variable-region domain, generated from the four-probe sequences shown at the left (see Fig. 2*b* for details). The profile is shown in the box. The rightmost column of the profile gives the penalty for insertion/deletion (+/−). Positions 31–47 of the profile are omitted from the figure for clarity. Notice that where gaps appear in some of the probe sequences, the insertion/deletion penalty is lower than elsewhere.

M Gribskov, A D McLachlan, and D Eisenberg (1987) Profile analysis: detection of distantly related proteins. PNAS 84:4355-8.

by Bob Friedman

Psi-Blast Results **Query: 55670331 (intein)**



```
gi|6706000|dbj|BAA06142.2|  DNA-dependent DNA polymerase [Pyrococ...   48   7e-04
gi|2708498|gb|AAB92484.1|  ribonucleotide reductase homolog [Baci...   48   7e-04
gi|50812254|ref|NP_389888.2|  hypothetical protein BSU20060 [Baci...   48   8e-04
gi|7475800|pir||A69927  ribonucleoside-diphosphate reductase (alp...   48   8e-04
gi|15211863|emb|CAC51...                                        ...bun...   46   0.002
gi|57867420|ref|YP_18907...                                     ...hat...   46   0.003
gi|14590941|ref|NP_143015.1|  ATP-dependent helicase LHR [Pyrococ...   46   0.003
```

Run PSI-Blast iteration 3

```
                    Sequences with E-value WORSE than threshold

gi|14590539|ref|NP_142607.1|  secretory protein kinase [Pyrococcu...   44   0.006
gi|45513096|ref|ZP_00164662.1|  COG1372: Intein/homing endonuclea...   44   0.009
```

link to sequence here, check BLink ☺

PSI BLAST and E-values!

Psi-Blast is for finding matches among divergent sequences (position-specific information)
WARNING: For the nth iteration of a PSI BLAST search, the E-value gives the number of matches to the profile NOT to the initial query sequence! The danger is that the profile was corrupted in an earlier iteration.

PSI Blast from the command line

Often you want to run a PSIBLAST search with two different databanks - one to create the PSSM, the other to get sequences:
To create the PSSM:

blastpgp -d nr -i subI -j 5 -C subI.ckp -a 2 -o subI.out -h 0.00001 -F f

blastpgp -d swissprot -i gamma -j 5 -C gamma.ckp -a 2 -o gamma.out -h 0.00001 -F f

Runs a 4 iterations of a PSIblast
the -h option tells the program to use matches with E <10^-5 for the next iteration,
(the default is 10^{-3})
-C creates a checkpoint (called subI.ckp),
-o writes the output to subI.out,
-i option specifies input as using subI as input (a fasta formatted aa sequence).
The nr databank used is stored in `/common/data/`
-a 2 use two processors

(It might help to use the node with more memory (017)
(command is `ssh node017`)

To use the PSSM:

```
blastpgp -d /Users/jpgogarten/genomes/msb8.faa -i subI -a 2 -R
subI.ckp -o subI.out3 -F f

blastpgp -d /Users/jpgogarten/genomes/msb8.faa -i gamma -a 2 -R
gamma.ckp -o gamma.out3 -F f
```

Runs another iteration of the same blast search, but uses the databank `/Users/jpgogarten/genomes/msb8.faa`

-R tells the program where to resume
-d specifies a different databank
-i input file - same sequence as before
-o output_filename
-a 2 use two processors

More on blastall:

BLAST
by Joseph Bedell; Ian Korf; Mark Yandell
Publisher: **O'Reilly**
Pub Date: **July 2003**
ISBN: **0-596-00299-8**
Pages: **360**
Slots: **1.0**

available at safari books online
http://proquestcombo.safaribooksonline.com/

Installation instructions and info on parameters at the NCBI:
http://www.ncbi.nlm.nih.gov/staff/tao/URLAPI/blastall/
ftp://ftp.ncbi.nlm.nih.gov/blast/documents/formatdb.html
ftp://ftp.ncbi.nlm.nih.gov/blast/documents/blast.html
ftp://ftp.ncbi.nlm.nih.gov/blast/documents/blastpgp.html
ftp://ftp.ncbi.nlm.nih.gov/blast/documents/fastacmd.html
ftp://ftp.ncbi.nlm.nih.gov/blast/documents/

http://www.bioinformatics.ubc.ca/resources/tools/blastall

http://en.wikipedia.org/wiki/BLAST