

# MCB 5472

Psi BLAST,  
Perl: Arrays, Loops

*J. Peter Gogarten*

Office: *BPB 404*

phone: *860 486-4061,*

Email: *[gogarten@uconn.edu](mailto:gogarten@uconn.edu)*

# 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

# Psi-Blast Model

Model of Psi-Blast:

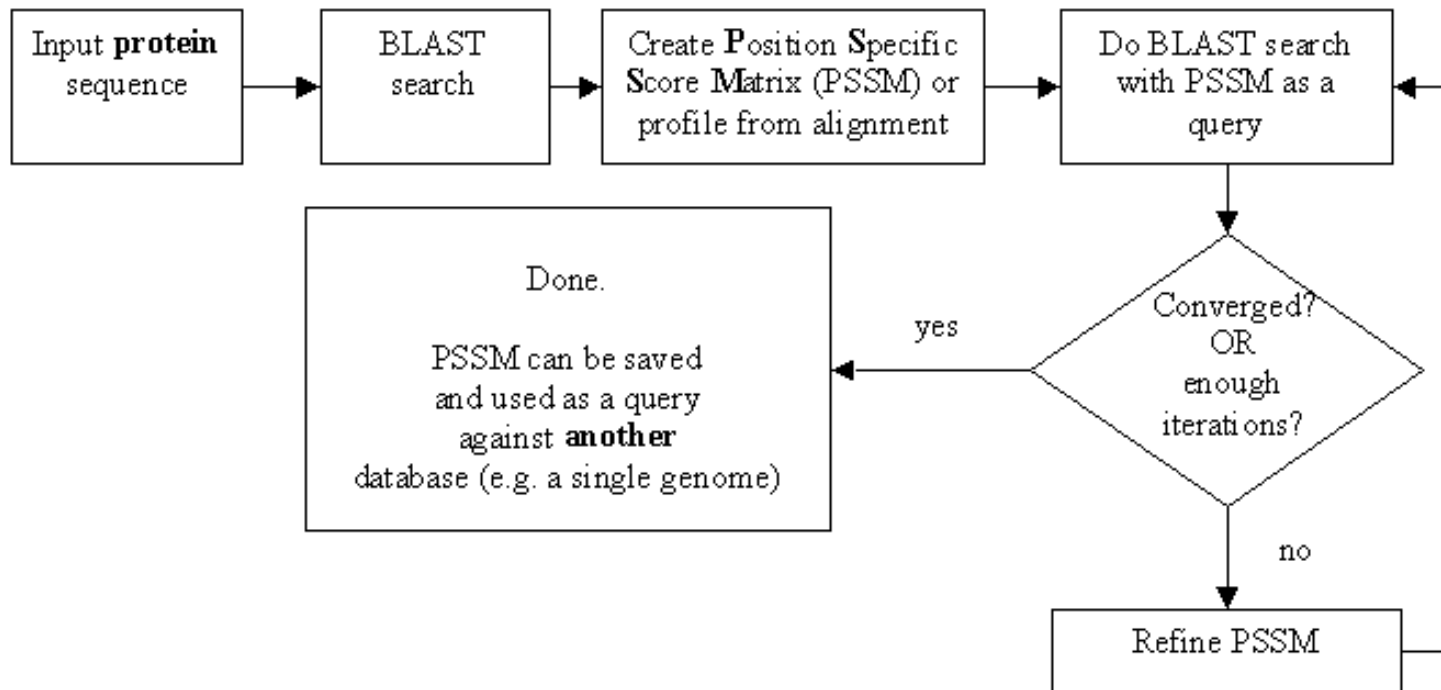
1. Use results of gapped BlastP query to construct a multiple sequence alignment
2. Construct a position-specific scoring matrix from the alignment
3. Search database with alignment instead of query sequence
4. Add matches to alignment and repeat

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

# PSI BLAST scheme



# Position-specific Matrix

POS	PROBE	CONSENSUS	PROFILE																				
			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	-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	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	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. 2b 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.

# Psi-Blast Results      Query: 55670331 (intein)

NEW	<input checked="" type="checkbox"/>	<a href="#">gi 6706000 dbj BAAU6142.2 </a>	DNA-dependent DNA polymerase [Pyrococ...	<a href="#">48</a>	7e-04	
NEW	<input checked="" type="checkbox"/>	<a href="#">gi 2708498 gb AAB92484.1 </a>	ribonucleotide reductase homolog [Baci...	<a href="#">48</a>	7e-04	
NEW	<input checked="" type="checkbox"/>	<a href="#">gi 50812254 ref NP_389888.2 </a>	hypothetical protein BSU20060 [Baci...	<a href="#">48</a>	8e-04	<b>G</b>
NEW	<input checked="" type="checkbox"/>	<a href="#">gi 7475800 pir  A69927</a>	ribonucleoside-diphosphate reductase (alp...	<a href="#">48</a>	8e-04	
NEW	<input checked="" type="checkbox"/>	<a href="#">gi 15211863 emb CAC51100</a>	bun...	<a href="#">46</a>	0.002	
NEW	<input checked="" type="checkbox"/>	<a href="#">gi 57867420 ref YP_18907</a>	hat...	<a href="#">46</a>	0.003	<b>G</b>
NEW	<input checked="" type="checkbox"/>	<a href="#">gi 14590941 ref NP_143015.1 </a>	ATP-dependent helicase LHR [Pyrococ...	<a href="#">46</a>	0.003	<b>G</b>

link to sequence [here](#),  
check BLink 😊

Run PSI-Blast iteration 3

## Sequences with E-value WORSE than threshold

<input type="checkbox"/>	<a href="#">gi 14590539 ref NP_142607.1 </a>	secretory protein kinase [Pyrococcu...	<a href="#">44</a>	0.006	<b>G</b>
<input type="checkbox"/>	<a href="#">gi 45513096 ref ZP_00164662.1 </a>	COG1372: Intein/homing endonuclea...	<a href="#">44</a>	0.009	
<input type="checkbox"/>	<a href="#">gi 156250075 ref YP_18907</a>		<a href="#">44</a>	0.009	<b>G</b>

# 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

## PSI Blast and finding gene families within genomes

use PSSM to search genome:

A) Use protein sequences encoded in genome as target:

```
blastpgp -d target_genome.faa -i query.name -a 2 -R query.ckp -o  
query.out3 -F f
```

B) Use nucleotide sequence and tblastn. This is an advantage if you are also interested in pseudogenes, and/or if you don't trust the genome annotation:

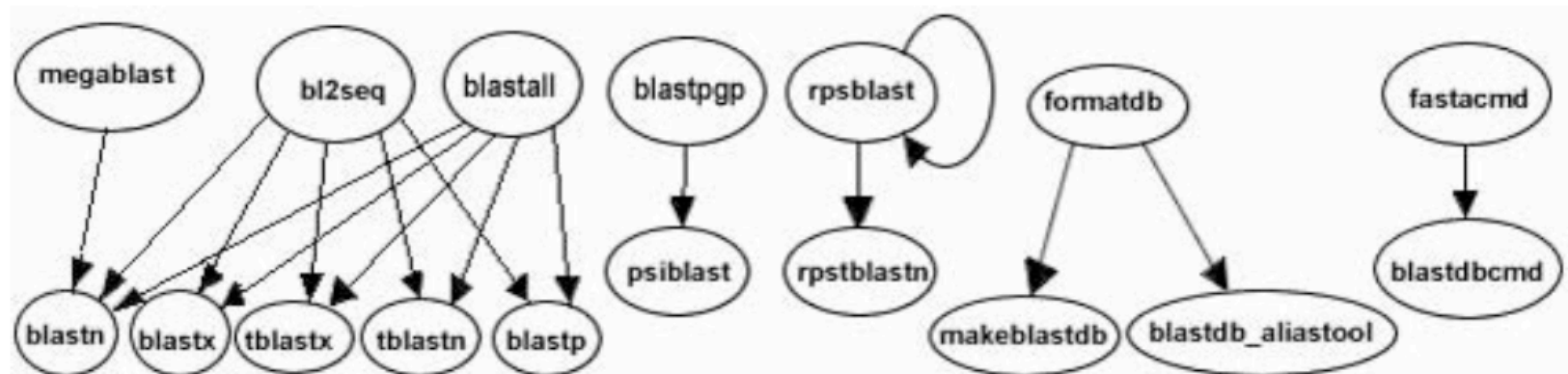
```
blastall -i query.name -d target_genome_nucl.ffn -p psitblastn -R  
query.ckp
```

The NCBI has released a new version of blast. The command line version is blast+ . The new version is faster and allows for more flexibility, but at present we still have problems with running it on the cluster.

The new commands are equivalent to the blastall commmands:

### Functionality offered by BLAST+ applications

The functionality offered by the BLAST+ applications has been organized by program type, as to more closely resemble Web BLAST. The following graph depicts a correspondence between the NCBI C Toolkit BLAST command line applications and the BLAST+ applications:



The `legacy_blast.pl` script that is part of `blast+` translates `blastall` commands into the `blast+` syntax. E.g.:

```
$ ./legacy_blast.pl megablast -i query.fsa -d nt -o mb.out --print_only
/opt/ncbi/blast/bin/blastn -query query.fsa -db "nt" -out mb.out
$
```

From the `blast+` manual:

The easiest way to get started using these command line applications is by means of the `legacy_blast.pl` PERL script which is bundled along with the BLAST+ applications. To utilize this script, simply prefix it to the invocation of the C toolkit BLAST command line application and append the `--path` option pointing to the installation directory of the BLAST+ applications. For example, instead of using

```
blastall -i query -d nr -o blast.out
```

use

```
legacy_blast.pl blastall -i query -d nr -o blast.out
--path /opt/blast/bin
```

## 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>

Example (~perl/class2/demo.pl)

```
#!/usr/bin/perl -w -s  
  
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_2010/class2.pl) [http://gogarten.uconn.edu/mcb5472\\_2010/class2.pl](http://gogarten.uconn.edu/mcb5472_2010/class2.pl)

## Old assignments:

### 1) What is the difference between a compiler and an interpreter?

A compiler takes program and translates it in low level executable language/code.

An interpreter goes through a program line by line and executes commands. The traditional distinction between compiled and interpreted languages is being blurred.

### 2) When is it useful to make a script executable, when not?

You save a little bit of typing when you make it executable, but else it is pretty equivalent. (If you start the program with `$ perl script_name.pl`, you don't **need** the shebang line. But the `-w` flag to use warnings is recognized.

Comment on `use strict;` and `use warnings, .`

# Old assignments:

3) What is the value of \$i after each of the following operations?

```
$i=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.



```
#!/usr/bin/perl #-w
```

```
my $i="";
```

```
print "\$i= $i\n";
```

```
$i = 1;
```

```
print "\$i= $i\n";
```

```
$i++;
```

```
print "\$i= $i\n";
```

```
$i *= $i;
```

```
print "\$i= $i\n";
```

```
$i .= $i;
```

```
print "\$i= $i\n";
```

```
$i = $i/11;
```

```
print "\$i= $i\n";
```

```
$i = $i . "score and" . $i+3 ;
```

```
print "\$i= $i\n";
```

```
$i = $i+3 . "score and" . $i;
```

```
print "\$i= $i\n";
```

```
$i=
```

```
$i= 1
```

```
$i= 2
```

```
$i= 4
```

```
$i= 44
```

```
$i= 4
```

```
$i= 7
```

```
$i= 10score and7
```

discuss and run test.pl with and without -w flag

Discuss and run the hello\_world script with variable and input

hello\_world\_variable.pl

```
#!/usr/bin/perl -w
# This is a Hello World program in Perl using a variable
my $who;                # Declare variable.
# You only need to use the declaration if you use strict
$who = "world";        # Assign variable.
print "Hello, $who!\n"; # Print result.
```

Discuss and run the hello\_world script with variable and input

hello\_world\_variable\_input.pl

```
#!/usr/bin/perl -w
use strict;
# This is a Hello World program in Perl using a variable
# and input
my $who;                # Declare variable.
$who = "world";        # Assign variable.
print "please enter your name: ";
chomp ($who = <>);
print "\nHello, $who!\n"; # Print result.
```

The best way to find which module to use is google. You can search core modules at <http://perldoc.perl.org/search.html?>

# Old assignments:

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.

```

$a=1;
$b=2;
$c = $a +$b;
print "\$c= $c\n";
$c = $a / $b;
print "\$c= $c\n";
$c = "$a + $b";
print "\$c= $c\n";
$c = '$a + $b';
print "\$c= $c\n";
$c = $a + $b++; # better use parenthesis $b is 3 at the end of this line
print "\$c= $c\n";
$c += $a ; #add the value of $a to $c and stores the result in $c
print "\$c= $c\n";
□

```

```

$c= 3
$c= 0.5
$c= 1 + 2
$c= $a + $b
$c= 3
$c= 4

```

Run and discuss test2.pl

- 2) Why does the first of these get along without `chomp ($line)`;  
(`chomp` is a built-in command in Perl to remove a trailing newline, if any, from a string).
- 3) Write a short Perl script that calculates the circumference of a circle given a radius provided by the user.

```
#!/usr/bin/perl -w
use strict;
print "This program finds the circumference of a circle.\n";
print "What is your radius?\n";
chomp (my $radius = <STDIN>);
print "The circumference of a circle with radius of $radius is\n";
print 2*3.141592654*$radius."\n"; #Equation for circle circumference
```

```
#!/usr/bin/perl -w
#As usual there are 1000 ways to do this.
#one is to define $pi or the constant PI, eg. as follows
#use constant PI => 4*atan2(1,1);
#or use a module
use Math::Trig; #allows to use the Math::Trig module that is part of perl
$circumference=0; #reset variables
print "\nEnter radius:";
chomp (my $radius=<>);
$circumference= $radius*pi*2;
print "\nwith radius=$radius ,\nthe circumference is $circumference\n\n";
```

From Wednesday:

For the following array declaration `@myArray = ('A', 'B', 'C', 'D', 'E');` what is the value of the following expressions:

`$#myArray`

`length (@myArray)`

`$myArray[1]`

`$n=@myArray`

`reverse (@myArray)`

```
#!/usr/bin/perl -w
print "\n\n";
@myArray = ('A', 'B', 'C', 'D', 'E');
print $#myArray; # returns highest number of field in array
print "\n";
print length($myArray[0]); # returns length of scalar - no idea what it does with an array
print "\n";
print $myArray[1]; #returns value in slot 1 (the 2nd entry - perl starts a 0)
print "\n";
print $n=@myArray; #one way to get the number of elements in an array
print "\n";
print reverse (@myArray); #comes in handy for DNA sequences.
print "\n";
```

4  
1  
B  
5  
EDCBA

Run and discuss myArray.pl

# Assignment for Monday (class 4)

- 1) Write a 2 sentence outline for your student project
- 2) Read chapter P5 and P12 conditional statements and on “for, foreach, and while” loops.

[http://korflab.ucdavis.edu/Unix\\_and\\_Perl/unix\\_and\\_perl\\_v2.3.3.pdf](http://korflab.ucdavis.edu/Unix_and_Perl/unix_and_perl_v2.3.3.pdf)

Background:

```
@a=(0..50);
```

```
# This assigns numbers from 0 to 50 to an array,
```

```
# so that $a[0]=0; $a[1]=1; $a[50]=50
```

- 3) Write perl scripts that add all numbers from 1 to 50. Try to do this using at least two different control structures.
- 4) Create a program that reads in a sequence stored in a file handed to the program on the command line and determines GC content of a sequence. Use class3.pl as a starting point.



Go through [class3.pl](#) script.

If time go through Olga's search for distant homologs  
webpage at

<http://www.mta.ca/~ozhaxybayeva/bioinf2010/class10.html>