Bioinformatics Facility of the Biotechnology

The Do and Don't of the Xserve Cluster

Pascal.Lapiere@uconn.edu

Head Node →

Xserve Cluster Physical Organization

Node001 To Node017

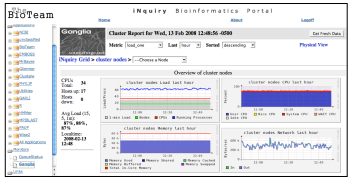
- 2 x 2.3GHz G5 processors
- 2 GB of memories per node (8 GB on node 17)
- 2.3 TeraBytes of Storage on the head node.
- 2 other mini cluster assigned for special projects

Basic Rules

- For research purpose only. Not a place to put your favorite MP3 or backup your HD.
- Do not overload the systems. It is ok to use ~6 nodes in period of low activities but when it gets busy, limit yourself to only 2-3 nodes if absolutely necessary.
- Always keep track of your jobs. Don't let analyses running unattended for months.
- Use the queue system whenever you can.
- Do not run jobs on the Head node.

Remote Access

- Via SSH or Web Interface



Useful Commands

(Help page available at : http://137.99.46.188/wiki/index.php/Main_Page)

-**qstat** : Shows the current status of the available Grid Engine queues and the jobs associated with the queues.

-**ls :** List directory contents

-**ps :** Display the process status. Allow to get process ID.
- -ps ux : Displays your process only
- -ps aux : displays all the process running on the node

-**du** : display disk usage statistics. Use du -h for a readable output

Useful Commands (cont)

-**mkdir** and **rmdir** : create and remove directories

-**cp** : copy files

-**mv** : moved files (can be used to rename files)

-**rm** : remove files. rm -r to remove files and sub-directories'

-**kill** : to kill a running process. Kill -9 'proc_id'

The queue system

"Managing Workload by Managing Resources and Policies"

-qstat : Display the queue status.

-qrsh : Queue remote shell. Automatically select an available node to log on.

-qsub : Queue submit. Automatically submit a job to an available node. Used in conjuncture with a shell script (see next slide).

-qdel : Delete a job running in the queue.

qdel - process_ID

How to submit a job using qsub?

A shell script is just a small text file pointing to what you want to run in the queue.

For example, if I want to submit a perl script (phyml.pl), I will create a text file name phyml.sh :

```
#!/bin/sh
cd /Users/nucleus/evolver
perl phyml_trees1.pl
#end of script
```

To submit the shell :
- qsub phyml.sh

Things to be cautious :

-While highly reliable, the cluster might sometimes run into problems and needed to be rebooted. This will cause to loose all the processes that were running at the time. Try to think of ways to break up or save at different stage of your analyses.

-The NFS (Network File System) has temporary amnesia when overwhelmed. The system will forget to write part of the output files. A workaround is to save to the scratch drive of the individual nodes (cd /scratch).

Tricks that I have learned

In Perl, Array of Arrays are useful for grid-like manipulations of data :

Seq.txt =

```
MRRAIATNQQ
MRLAIISRQD
MRRLSISRQQ
MRLAIIIRQQ
```

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

$infile = "seq.txt";
open (FILE,$infile);
While ($in = <FILE>) { #go infile line by line

chomp $in;
@data = split ('',$in); #split using ''
push @matrix,[@data]; # put the array @data
                        into @matrix
}
```

```
  0123456789
0 MRRAIATNQQ
1 MRLAIISRQD
2 MRRLSISRQQ
3 MRLAIIIRQQ
```

Print $matrix[2][4]; → S

Retrieving data directly from NCBI using E-tools

- fastacmd -s 49220 -d nr →

You can use E-tools to get the Genbank file for 49220 (http://www.ncbi.nlm.nih.gov/entrez/query/static/eutils_help.html)

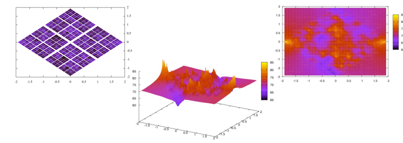
-read-tax.pl :

Database [Pubmed]: protein
Query [zanzibar]: 49220
Report [abstract]: genbank →

Gnuplot

(http://www.gnuplot.info/)

-Great at generating plots on the fly using Perl
-Can handle enormous datasets
-Easy to use, very powerful

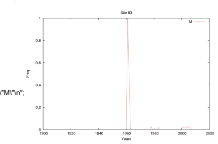


Gnuplot (cont)

-Only works if you are on the Head node!
-You can install and use it on you personal computer (PC/MAC)

```
$output="gnu.out";
open(OUT, ">$output");
print OUT "set term postscript color\n";
print OUT "set output \"site_$x.ps\"\n";
print OUT "set title \"Site $x\"\n";
print OUT "set xlabel \"Years\"\n";
print OUT "set ylabel \"Freq\"\n";
print OUT "set yrange [0 to 1]\n";
print OUT "plot \"temp.out\" using 1:2 with line title \"M\"\n";

System ("gnuplot gnu.out");
…
```



Old Assignments

1) Review PSIblast Your questions?
2) Write a 3 sentence outline for your student project
   Send me an email on this!
3) Re-read chapter 2 p32 - p34 on control structures and page 142 -146 on for, foreach, and while loops

For next week:

Background: `@a=(0..50);`
#assigns numbers from 0 to 50 to an array, so that $a[0] =0; $a[1] =1; $a[50] =50

4) Write perlscripts that add all numbers from 1 to 50. Try to do this using at least to different control structures.

Control structures: Sum 1..50

while ( ) { }

```
#!/usr/bin/perl -w
$sum=0;
$count=0;

while ($count <50) {
    $count++; #this is tricky in the last loop $count is 49 and then increased to 50 and added
    $sum += $count;
    };
print "$sum\n"
```

for (, ,) { }

```
#!/usr/bin/perl/
$sum=0;
$count=0;

for ($count =0; $count < 51; $count++)  {
    #$sum=$sum+$count;
    $sum += $count
};
print "$sum\n"
```

Control structures: Sum 1..50

foreach ( ) { };

```
#!/usr/bin/perl/
$sum=0;
@array = (1..50);
foreach (@array)  {
    #$sum=$sum+$_;
    $sum += $_;
};
print "$sum\n"
```

Infinite loop with last:

while () {

if( ) {last};

};

```
#!/usr/bin/perl/
$sum=0;
$count=0;
while () {
    $sum += $count;
    $count+=1;
    if ($count >50) {last};
}
print "$sum\n"
```

Control structures: Sum 1..50

while (defined ( )) { };

```
#!/usr/bin/perl
$sum=0;
@array = (1..50);
$count=0;
while (defined($array[$count]))
{
    $sum += $array[$count];
    $count += 1;
    #print "$array[$count]\t $sum\n";
};
print "$sum\n"
```

for (, ,) { }

Counting elements of an array

Could have started at 0

```
#!/usr/bin/perl -w
$sum=0;
@array = (0..50);
$count=0;
for ($count=1; ($count<51); $count++){
    $sum += $array[$count];
    #$temp=$array[$count];
    #print "\$count=$count sum is $temp\t $
    }
print "$sum\n";
```

For Wednesday

- Email your 3 sentence project outline
- Read NCBI info on geneplot (here)
- Try geneplot comparing your favorite genomes (e.g. here)
- What might be a problem using geneplot?

For Monday

- Read chapter 3 in Learning Perl

Write a script that reads in a sequence and prints out the reverse complement.

- Modify your script to that it can handle a sequence that goes over several lines?

Background: `$comp =~ tr/ATGC/TACG;`
#translates every A in $comp into a T; every T into an A; every G into a C and every C into a G

Geneplot

In a perfect world you do not want to plot gi numbers but positions in a genome. The script addnumnuc.pl adds the nucleotide position of the ORF (the central one) to the beginning of the annotation line.

.ptt files

Available on the ftp server at NCBI or each chromosome. E.g.

Fervidobacterium nodosum Rt17-B1, complete genome - 1..1948941
1750 proteins

Location	Strand	Length	PID	Gene	Synonym	Code	COG	Product
43..1377	+	444	154248706	-	Fnod_0001	-	-	chromosomal replicati
1453..1635	+	60	154248707	-	Fnod_0002	-	-	4Fe-4S ferredoxin iror
1976..3829	+	617	154248708	-	Fnod_0003	-	-	hypothetical protein
3826..4926	+	366	154248709	-	Fnod_0004	-	-	basic membrane lipop
5136..6701	+	521	154248710	-	Fnod_0005	-	-	ABC transporter relat
6698..7732	+	344	154248711	-	Fnod_0006	-	-	inner-membrane trans
7729..8688	+	319	154248712	-	Fnod_0007	-	-	inner-membrane trans
8734..9132	+	132	154248713	-	Fnod_0008	-	-	protein of unknown fu
9261..9617	+	118	154248714	-	Fnod_0009	-	-	hypothetical protein
9745..10020	+	91	154248715	-	Fnod_0010	-	-	histone fa
10098..11342	-	414	154248716	-	Fnod_0011	-	-	metal dep
11361..13514	-	717	154248717	-	Fnod_0012	-	-	hypothetic
13511..14161	-	216	154248718	-	Fnod_0013	-	-	hypothetic
14158..15102	-	314	154248719	-	Fnod_0014	-	-	putative m
15115..15969	-	284	154248720	-	Fnod_0015	-	-	putative a
16022..17008	-	328	154248721	-	Fnod_0016	-	-	putative C
17156..17566	+	136	154248722	-	Fnod_0017	-	-	protein of
17594..19282	+	562	154248723	-	Fnod_0018	-	-	sigma54 s
19623..19859	+	78	154248724	-	Fnod_0019	-	-	hypothetic
19856..20074	+	72	154248725	-	Fnod_0020	-	-	hypothetic
20095..20289	+	64	154248726	-	Fnod_0021	-	-	hypothetic

....

Addnumnuc.pl part 1

```perl
#!/usr/bin/perl -w
#decided to have input file entered in command line
#call program followed by genome name.
#the program assumes that a file with the extensions ptt and faa exist in the same dirctory.
#####INPUT Name of multiple seq file containing ORF of genome, open file and assign IN filehandle #
unless(@ARGV==1) {die "please provide genome name in command line \n
file should contain multiple sequences in fasta format \n
a file with the ptt table should be in the same directory\n\n";}
#$num=0;
$filename=$ARGV[0];
@nameparts=split(/\./, $filename);
#print $parts[0];
$orfs="$nameparts[0]"."\.faa";
$ptt="$nameparts[0]"."\.ptt";

open(IN, "< $ptt") or die "cannot open $ptt:$!";
$line=<IN>; # read t1st line
    if ($line=~/complete genome/) {  #look forheader
        print "$line\n";};
$line=<IN>; # read 2nd line
    print "$line\n";
$line=<IN>; # read 3rd line
if  ($line=~/Location   Strand/) {  #look for beginning of table

    while (defined ($line=<IN>)){ # read through rest of table line by line

        @parts=split/\t/,$line;
        @fromto=split/\.\./,$parts[0];
        $middle = (($fromto[1]+$fromto[0])/2);
        print "$fromto[1]\t$fromto[0]\t$middle\t$parts[3]\t";
 $gi_hash{$parts[3]}=$middle;
        print "\n";
    }
}
@gi_names = sort(keys(%gi_hash));
$total=scalar(@gi_names);

print "total number of GIs= $total\n";
foreach (@gi_names) {
    print "gi number $_ is located at $gi_hash{$_}\n";
    }

close(IN);
```

Addnumnuc.pl part 2

```perl
# read in and process faa file

open(IN, "< $orfs") or die "cannot open $filename:$!";
$outfilename = "$nameparts[0]"."\.num\.faa";
open(OUT, "> $outfilename")||die "cannot open: $!";

#################################################


while (defined ($line=<IN>)){ # read through file line by line

    if ($line=~/^>/) {  #look for beginning of line starting with > (^ is an anchor for the beginning of the line)
        $line =~ m/gi\|(\d+)\|/; #match gi|number capture number in $1
        $num=$gi_hash{$1};
        $line =~ s/^>//;
        # print "$1 $num \n";
        $line= ">"."$num\t"." $line";
        };
print "$line"; #print to screen
print OUT "$line"; #print to OUT
    }

close(IN);
close(OUT);
```

Results in a multiple fasta file where each annotation line starts with the nucleotide position in the chromosome:

Tmar.num.faa:

```
>385.5     gi|15642776|ref|NP_227817.1| hypothetical protein TM0001 [Thermotoga maritima MSB8]
MVYGKEGYGRSKNILLSECVCGIISLELNGFQYFLRGMETL
>545.5     gi|15642777|ref|NP_227818.1| hypothetical protein TM0002 [Thermotoga maritima MSB8]
MSPEDWKRLICFHTSKEVLKQTLDDAQQNISDSVSIPLRKY
>1828      gi|15642778|ref|NP_227819.1| hypothetical protein TM0003 [Thermotoga maritima MSB8]
METVKAYEVEDIPAIGFNNSLEVWKLFPASSSRSTSSSFQ
>1974.5    gi|15642779|ref|NP_227820.1| hypothetical protein TM0004 [Thermotoga maritima MSB8]
MKDLYERFNNSLEVWKLVELFGTSIRIHLFQ
>4131      gi|15642780|ref|NP_227821.1| DNA helicase, putative [Thermotoga maritima MSB8]
MTVQQFIKKLVRLVELERNAEINAMLDEMKRLSGEEREKKGRAVLGLTGKFIGEELGYFLVRFGRRKKID
TEIGVGDLVLISKGNPLKSDYTGTVVEKGERFITVAVDRLPSWKLKNVRIDLFASDITFRRQIENLMTLS
SEGKKALEFLLGKRKPEESFEEEFTPFDEGLNESQREAVSLALGSSDFFLIHGPFGTGKTRTLVEYIRQE
VARGKKILVTAESNLAVDNLVERLWGKVSLVRIGHPSRVSSHLKESTLAHQIETSSEYEKVKKMKEELAK
LIKKRDSFTKPSPQWRRGLSDKKILEYAEKNWSARGVSKEKIKEMAEWIKLNSQIQDIRDLIERKEEIIA
SRIVREAQVVLSTNSSAALEILSGIVFDVVVVDEASQATIPSILIPISKGKKFVLAGDHKQLPPTILSED
AKDLSRTLFEELITRYPEKSSLLDTQYRMNELLMEFPSEEFYDGKLKAAEKVRNITLFDLGVEIPNFGKF
WDVVLSPKNVLVFIDTKNRSDRFERQRKDSPSRENPLEAQIVKEVVEKLLSMGVKEDWIGIITPYDDQVN
LIRELIEAKVEVHSVDGFQGREKEVIIISFVRSNKNGEIGFLEDLRRLNVSLTRAKRKLIATGDSSTLSV
HPTYRRFVEFVKKKGTYVIF
```

....

Geneplot using EXCEL part 1

Format databank using Tpet.num.faa

```
>formatdb -i Tpet.num.faa -p T -o T
```

Search databank using Tmar.num.faa using blastall with -m8

```
> blastall -p blastp -d Tlet.num.faa -i
Tmar.num.faa -o Tlet_Tmar.tab -F F -m 8 -W 2 -a
2 -e 0.001
```

You could use different E values

Load output (in this case Tlet_Tmar.tab) into Excel;
(note the script addnimnuc added an extra tab - tell the import to ignore consecutive \t s)

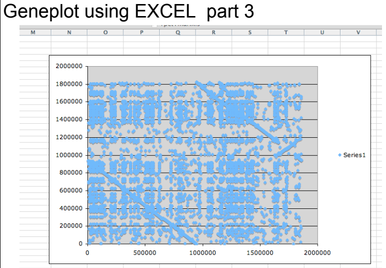
Geneplot using EXCEL part 2

TpetTmar.xls

	A	B	C	D	E	F	G	H	I	J	K	L
1	4131	939008	99.38	650	4	0	1	650	1	650	0	1182
2	5622.5	937516.5	99.71	345	1	0	1	345	1	345	0	638
3	6803	936336	99.32	438	3	0	1	438	1	438	0	833
4	7832	935307	99.55	224	1	0	1	224	1	224	######	435
5	8419	934720	99.3	142	1	0	1	142	1	142	1.00E-60	224
6	8419	1278886.5	37.32	142	78	3	6	141	5	141	1.00E-13	67.8
7	8419	843801	32.89	152	91	4	1	141	1	152	2.00E-11	60.5
8	9541.5	933597.5	100	607	0	0	1	607	1	607	0	1206
9	9541.5	1360664	52.49	602	272	5	7	596	27	626	0	626
10	9541.5	700745.5	52.64	530	247	1	7	536	5	530	######	561
11	9541.5	932477	35.79	95	57	2	8	99	33	126	2.00E-10	59.7
12	9541.5	908506.5	46.3	54	26	2	541	594	189	239	1.00E-08	54.3
13	9541.5	1361839.5	36.78	87	43	3	2	81	69	150	3.00E-06	46.2
14	9541.5	932040	30.34	89	57	2	2	86	75	162	4.00E-06	45.8
15	9541.5	684306.5	23.15	324	200	12	89	412	66	340	4.00E-05	42.4
16	9541.5	896459.5	40.74	54	29	1	543	596	207	257	5.00E-05	42
17	9541.5	1364003.5	41.46	41	23	1	548	588	12	51	4.00E-04	39.3
18	10665	932477	100	126	0	0	3	128	1	126	3.00E-56	209
19	10665	933597.5	35.79	95	57	2	35	128	8	99	3.00E-11	59.7
20	10665	700745.5	29.79	94	62	2	35	127	6	96	5.00E-04	35.8
21	10665	1360664	26.5	117	67	3	16	121	14	122	5.00E-04	35.4
22	11120	932040	99.38	162	1	0	15	176	1	162	1.00E-90	324
23	11120	1361839.5	43.59	156	85	2	20	175	2	154	1.00E-32	131
24	11120	701797.5	41.73	139	79	2	38	175	18	155	4.00E-29	119
25	11120	1358741.5	38.1	84	46	3	89	171	565	643	4.00E-08	50.1
26	11120	1360664	33.71	89	52	2	95	176	28	116	9.00E-08	48.9
27	11120	933597.5	30.34	89	57	2	89	176	2	86	8.00E-07	45.8
28	11120	1364883.5	29.11	79	45	3	99	176	1	69	4.00E-05	40

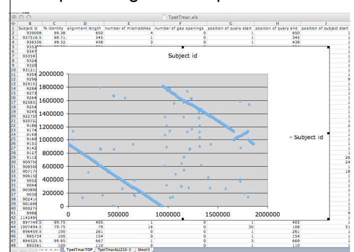
Plotting column B against A ->

Geneplot using EXCEL part 3



To only plot the top scoring hits use extract_lines.pl -->

Geneplot using EXCEL part 4



Plotting Tpet_Tmar.tab.top

PSIBlast to find transposase homologs

- Download transposase sequence transposase.fa
- Download genome as nucleotide sequence
- Format genome

- `formatdb -i Tpet.fna -p F -o T`
- `blastpgp -i transposase.fa -d nr -I T -h 0.00001 -j 6 -C transposase.chk -a2`
- `blastall -i transposase.fa -d Tpet.fna -p psitblastn -R transposase.chk -o transposase_Tpet.tab -a2 -m8 -F F`

transposase_Tpet.tab:

```
ate5 jpgoparten$ more transposase_Tpet.tab
1       gi|157262870|ref|NC_009028.1|   18.54   426     334     9       11      423     463614  462364  3e-101  361
1       gi|157262870|ref|NC_009028.1|   15.46   196     158     5       10      197     462999  462448  4e-14   72.7
1       gi|157262870|ref|NC_009028.1|   12.21   434     335     17      5       392     1945857 1947041 2e-08   53.4
1       gi|157262870|ref|NC_009028.1|   19.20   125     92      5       249     364     1079762 1080133 6e-08   51.9
1       gi|157262870|ref|NC_009028.1|   12.29   293     247     12      13      295     669830  669084  2e-07   50.0
1       gi|157262870|ref|NC_009028.1|   14.61   178     132     8       160     317     1375657 1375151 1e-05   44.6
1       gi|157262870|ref|NC_009028.1|   10.29   175     145     6       144     306     334563  337062  5e-05   42.3
1       gi|157262870|ref|NC_009028.1|   16.12   273     199     14      149     391     1314911 1315603 7e-05   41.9
1       gi|157262870|ref|NC_009028.1|   12.93   348     291     8       8       343     2023445 2022588 0.001   38.0
1       gi|157262870|ref|NC_009028.1|   11.25   160     125     7       257     399     1942255 1942806 0.001   38.0
```