Cluster Computing With R

Stowers Institute for Medical Research
R/Bioconductor Discussion Group

Earl F. Glynn
Scientific Programmer
18 December 2007
Cluster Computing With R

• Accessing Linux Boxes from Windows
  • Linux Servers
  • Cluster Head Nodes
  • Linux Command Prompt
  • xterms
  • Linux Home Directories
  • Setup For Examples
• R Batch Jobs Using Linux
• Cluster Basics
• Sun Grid Engine
• R Batch Jobs on Cluster:
  Toy Examples and Time Perturbation Study
Accessing Linux Boxes from Windows

Wiki: http://research/PuTTYxming

PuTTY / Xming will be used to access Linux boxes from Windows
Accessing Linux Boxes from Windows

PuTTY / xming

Accessing Linux Hosts

"Right" Click Xming icon

Can access via PuTTY icon

genekc02 64-bit
genekc03 32-bit
Accessing Cluster Head Nodes

PuTTY / xming

Confusing Names

32-bit cluster nodes
cluster01 = Betelgeuse
cluster02 = Sirius

64-bit cluster nodes
cluster03 = Orion

"Right" Click Xming icon
Making Your Linux Bash Command Prompt Useful

http://www.expertsrt.com/tutorials/Matt/CmdPrompt.html
Accessing Linux Boxes from Windows

**Linux Command Prompt and xterms**

![Linux Command Prompt Example]

```plaintext
login as: efg
efg@genke03:~$ password:
Last login: Mon Dec 17 13:22:33 2007 from a00g7sws81.sto

[5 17Dec07 13:48:24 /home/efg]
```

```
$ xterm &
[1] 17809

[5 17Dec07 13:48:24 /home/efg]
```

```
$ xterm &
[2] 17815

[7 17Dec07 13:48:30 /home/efg]
```

```plaintext
[5 17Dec07 13:48:30 /home/efg]
```
LINUX Home Directories

• Files for cluster jobs normally should be in your home directory, e.g., /home/efg

• Suggest single directory per cluster job, e.g., /home/efg/cluster/R/simplegraph

• LINUX home directories can be accessed via Windows using an UNC name:
  \lnaskc01\unixhomes\efg\cluster\R\simplegraph

• Not all filesystems available on genekc02, ..., are available on clusters, e.g., $BLASTDB
Summary

- Use Linux boxes for R "batch" jobs (discussed next)
- Normally use genekc02 or genekc03 boxes
- Can monitor how busy a box is with "top" (cluster is possible alternative when busy)

```
top - 15:36:57 up 4 days, 4:08, 12 users, load average: 1.14, 0.70, 0.43
Tasks: 131 total, 2 running, 129 sleeping, 0 stopped, 0 zombie
Cpu(s): 29.3% us, 0.7% sy, 0.0% ni, 69.9% id, 0.0% wa, 0.0% hi, 0.0% si
Mem: 9095360k total, 5142824k used, 3952536k free, 150068k buffers
Swap:   0k total,   0k used,   0k free, 4151892k cached
```

<table>
<thead>
<tr>
<th>PID</th>
<th>USER</th>
<th>PR</th>
<th>NI</th>
<th>VIRT</th>
<th>RES</th>
<th>SHR</th>
<th>S</th>
<th>%CPU</th>
<th>%MEM</th>
<th>TIME+</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>22085</td>
<td>hull</td>
<td>23</td>
<td>0</td>
<td>51268</td>
<td>45m</td>
<td>2044</td>
<td>R</td>
<td>99</td>
<td>0.5</td>
<td>0:00.82</td>
<td>R</td>
</tr>
<tr>
<td>22502</td>
<td>hull</td>
<td>20</td>
<td>0</td>
<td>5766</td>
<td>4504</td>
<td>1340</td>
<td>S</td>
<td>6</td>
<td>0.0</td>
<td>0:00.17</td>
<td>perl</td>
</tr>
<tr>
<td>22504</td>
<td>hull</td>
<td>19</td>
<td>0</td>
<td>7712</td>
<td>4504</td>
<td>1340</td>
<td>S</td>
<td>6</td>
<td>0.0</td>
<td>0:00.17</td>
<td>perl</td>
</tr>
<tr>
<td>17622</td>
<td>hull</td>
<td>15</td>
<td>0</td>
<td>8396</td>
<td>1968</td>
<td>1320</td>
<td>S</td>
<td>0</td>
<td>0.0</td>
<td>0:01.33</td>
<td>ssdh</td>
</tr>
<tr>
<td>1 root</td>
<td>root</td>
<td>16</td>
<td>0</td>
<td>2444</td>
<td>552</td>
<td>472</td>
<td>S</td>
<td>0</td>
<td>0.0</td>
<td>6:40.06</td>
<td>init</td>
</tr>
<tr>
<td>2 root</td>
<td>root</td>
<td>16</td>
<td>0</td>
<td>2444</td>
<td>552</td>
<td>472</td>
<td>S</td>
<td>0</td>
<td>0.0</td>
<td>0:24.23</td>
<td>migration/0</td>
</tr>
<tr>
<td>3 root</td>
<td>root</td>
<td>34</td>
<td>19</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>S</td>
<td>0</td>
<td>0.0</td>
<td>0:03.58</td>
<td>ksoftirqd/0</td>
</tr>
</tbody>
</table>
Setup for Examples

- Login on genekc02 or genekc03
- In your home directory:

  mkdir simplegraph
cd simplegraph
cp /home/efg/cluster/R/simplegraph/* .
cd ..
  mkdir array
cd array
cp /home/efg/cluster/R/array/* .

Or, run: /home/efg/cluster/R/Rclass.bash
R Batch Jobs Using Linux

- Develop R script using Windows or Linux
- Develop parameterized R script [some limitations]

- R CMD BATCH

- Create Bash driver for R script
- Test Bash driver using "normal" Linux box
Simple R Script and Batch Driver

```
[51 17Dec07 16:33:32 /home/efg/cluster/R/simplegraph]
cat simplegraph0.R
# Base case before R script "parameterized" for command-line driver
# efg, 27 Sept 2007

# Default parameters (any or all to be command line parameters)
Filename <- "IdealGene"
Amplitude <- 1.0
Period <- 30.0  # minutes (default for zebrafish somitogenesis)
Color <- "black"
ChartTitle <- "Ideal Gene Expression"

cat(paste("Filename=" , Filename, "", Amplitude=" , Amplitude,
"", Period=" , Period, "", Color=" , Color, "", ChartTitle=" , ChartTitle, "\n", sep=" " ))

# Plot graphic from parameters passed to R
t <- 0:120  # minutes
pdf(file=paste(Filename," .pdf", sep=" " ), width=6.0, height=6.0)
plot(t, Amplitude*cos(2*pi*t/Period), type="o",
     xlab="Time [minutes]", ylab="Gene Expression",
     col=Color, main=ChartTitle)
dev.off()
```
R Batch Jobs Using Linux

**R CMD BATCH**

At Linux command prompt or in a script

R CMD BATCH --vanilla --slave scriptname.R output.txt

--vanilla

Combination of
--no-save: Don't save it
--no-restore: Don't restore anything
--no-site-file: Don't read the site-wide Rprofile
--no-init-file: Don't read the .Rprofile or ~/.Rprofile files
--no-environ: Don't read the site and user environment file

--slave: Make R run as quietly as possible

See man R on Linux
R Batch Jobs Using Linux

Simple R Script and Batch Driver

```r
[51 17Dec07 16:33:32 /home/efg/cluster/R/simplegraph]
cat simplegraph0.R
# Base case before R script "parameterized" for command-line driver
# efg, 27 Sept 2007

# Default parameters (any or all to be command line parameters)
Filename <- "IdealGene"
Amplitude <- 1.0
Period  <- 30.0 # minutes (default for zebrafish somitogenesis)
Color   <- "black"
ChartTitle <- "Ideal Gene Expression"

cat(paste("Filename",Filename,"",Amplitude,"",Amplitude,"",Period,"",Color,"",ChartTitle,"\n",sep=""))

# Plot graphic from parameters passed to R
t <- 0:120 # minutes
pdf(file=paste(Filename,".pdf", sep=""), width=6.0, height=6.0)
plot(t, Amplitude*cos(2*pi*t/Period), type="o",
    xlab="Time [minutes]", ylab="Gene Expression",
    col=Color, main=ChartTitle)
dev.off()
```

q() not needed

```bash
[52 17Dec07 16:33:45 /home/efg/cluster/R/simplegraph]
cat simplegraph0.bash
#!/bin/bash
# simplegraph0,R Driver. efg, 27 Sept 2007
R CMD BATCH --vanilla --slave simplegraph0.R simplegraph0.txt
```

Run the script

```bash
[53 17Dec07 16:33:51 /home/efg/cluster/R/simplegraph]
./simplegraph0.bash
```
Simple R Script and Batch Driver

R Batch Jobs Using Linux

Look at txt output file using Linux

Windows Explorer: Usually need to "refresh" to see files

Look at pdf output file using Windows

Use Windows file "extensions" so files can be processed in either Windows or Linux.
Including arguments in R CMD BATCH mode

When you have multiple computers or processors at your disposal and wish to run the same script with different arguments, use the following at the command line (here described for Linux; remove the linebreak, it is just there for display purposes):

```r
$ R CMD BATCH --no-save --no-restore '---args a=1
  b=c(2,5,6)'
test.R test.out
```

Where test.R is the R script file you wish to run and test.out is a text file to include the screen output of the R terminal. A key point here is that each argument must have no spaces because --args is space delimited.

To include the variables listed in --args, adapt the following code from test.R:

```r
### First read in the arguments listed at the command line
args<-(commandArgs(TRUE))
### args is now a list of character vectors
### First check to see if arguments are passed.
### Then cycle through each element of the list and evaluate the expressions.
if(length(args)==0){
  print("No arguments supplied.")
  ### supply default values
  a = 1
  b = c(1,1,1)
} else{
  for(i in 1:length(args)){
    eval(parse(text=args[[i]]))
  }
}
```
Parameterized R Script and Driver

Parameterized R Script and Driver

```
# Base case before R script "parameterized" for command-line driver
# efg, 27 Sept 2007

options(echo=TRUE)

# Default parameters (any or all to be command line parameters)
Filename <- "IdealGene"
Amplitude <- 1.0
Period <- 30.0 # minutes (default for zebrafish somitogenesis)
Color <- "black"
ChartTitle <- "Ideal Gene Expression"

# See Quantitative Ecology blog, Aug 16, 2007
# including-arguments-in-r-cmd-batch-mode.html
# "Including arguments in R CMD BATCH mode",
#
# commandArgs TRUE argument introduced in R 2.5.0
args <- commandArgs(TRUE)

if (length(args) > 0)
{
  for (i in 1:length(args))
  {
    eval( parse(text=args[[i]]) )
  }
}

# Plot graphic from parameters passed to R
t <- 0:120 # minutes
pdf(file=paste(Filename,".pdf", sep=""), width=6.0, height=6.0)
  plot(t, Amplitude*cos(2*pi*t/Period), type="o",
    xlab="Time [minutes]", ylab="Gene Expression",
    col=Color, main=ChartTitle)
dev.off()
```
Parameterized R Script and Driver

Mouse

Zebrafish

Should have overridden ChartTitle. Suggestions?
R Batch Jobs Using Linux

Parameterized R Script and Driver

```r
# Default parameters (any or all to be command line parameters)
Filename <- "IdealGene"
Amplitude <- 1.0
Period <- 30.0  # minutes (default for zebrafish mitogenesis)
Color <- "Black"
ChartTitle <- "Ideal Gene Expression"

# See Quantitative Ecology blog, Aug 16, 2007
# "Including-arguments-in-R-CMD-BATCH-mode.html"
# "Including arguments in R CMD BATCH mode".
# commandArgs(TRUE) argument introduced in R 2.5.0
args <- commandArgs(TRUE)

[i] "Filename=zebrafish" "Amplitude=0.5" "Period=30"

> length(args)
[i] 6

> # Use command line parameters as R variables -- no error checking
> if (length(args) > 0) {
+ for (i in 1:length(args)) {
+   eval(parse(text=args[i]))
+ }
+ }

> cat(paste("Filename=",Filename," Amplitude=",Amplitude,
+ " Period=",Period," Color=",Color," ChartTitle=",ChartTitle," \n"," sep=""))
Filenamexzebrafish, Amplitude=0.5, Period=30, Color=blue, ChartTitle=Ideal Gene Expression

> # Plot graphic from parameters passed to R
> t <- 0:120  # minutes
> pdf(file=paste(Filename,".pdf",sep=""), width=6.0, height=6.0)
> plot(t, Amplitude*cos(2*pi/Period), type="l",
+ xlab="Time [minutes]", ylab="Gene Expression",
+ col=Color, main=ChartTitle)
> dev.off()
null device
1

> proc.time()
user  system elapsed
1.000  0.100  1.104
```

Windows:
Use WordPad (but NOT NotePad) to view .txt file
Cluster Basics

• R is configured nearly identically on 32-bit and 64-bit platforms.
• Limitations: Can be subtle differences in environmental variables, executable paths, between head node and cluster nodes, or head node and other Linux boxes.
• A "good" job for cluster is mostly CPU intensive - many computations.
• Short jobs, or jobs with a lot of I/O, may not be good cluster jobs.
• Must search standard out/error files for errors
Many changes are coming over next few months. Wiki pages are a bit dated right now.
Cluster Basics

Betelgeuse / cluster01


Confusing Names

32-bit cluster nodes
cluster01 = Betelgeuse
cluster02 = Sirius

64-bit cluster nodes
cluster03 = Orion
Cluster Basics

Betelgeuse / cluster01

Load Monitor
Cluster Basics

Betelgeuse / cluster01
Cluster Head Node

• Directly login to cluster head node using Xming (right click, SIMR cluster, <cluster name>)

• Don't do any unnecessary work on the head node.

• Use head node **ONLY** to submit jobs via the Sun Grid Engine (discussed next)

• If you forget, kill any jobs accidentally run on cluster head node as soon as possible.
Sun Grid Engine

- qrsh: Interactively run job when genekc02, genekc03, ..., are too busy

- qsub: Submit cluster jobs

- qstat: Status of cluster jobs submitted

- qdel: Delete cluster job

- qmon: GUI Job Monitor (submit, manage, monitor)
qrsh

- What if genekc02, genekc03, ... are fully loaded and you want to use another CPU?
- What if you have a very long-running job and would like to not "hog" the main Linux boxes?
- Some people have no choice where to run jobs due to license limitations. When licensed CPUs are in use, they have no options. Be kind to your neighbor by using cluster node when possible.

- If cluster nodes are available, qrsh can be used as an alternative.
1. Check that cluster nodes are available.

2. Login to cluster head node:
   cluster01 (Betelgeuse) for now
   [right click Xming | SIMR Cluster | cluster01]
3. Open xterm if desired.

4. Issue qrsh command:
   `qrsh -q all.q`

   In this case we were assigned node0050

5. Change to work directory and process work
5. Process any work on cluster node

```
[6 17Dec07 17:38:33 /home/efg]
qrsh -q all,q

[6 17Dec07 17:40:37 /home/efg]
cd cluster/R/simplegraph

[7 17Dec07 17:42:15 /home/efg/cluster/R/simplegraph]
./batch.bash
  + simplegraph.bash mouse 2.0 120 red
  + R CMD BATCH --vanilla --slave '--args Filename=""mouse"" Amplitude=2.0 Period=120 Color=""red"" Flag=FALSE Extra=""No.Blanks""' simplegraph.R mouse.txt
  + simplegraph.bash zebrafish 0.5 30 blue
  + R CMD BATCH --vanilla --slave '--args Filename=""zebrafish"" Amplitude=0.5 Period=30 Color=""blue"" Flag=FALSE Extra=""No.Blanks""' simplegraph.R zebrafish.txt
```
qrsh

6. Enter "exit" to close qrsh session on node. Enter "exit" to close cluster01 session.
**qsub**

Submit job to cluster

- Write "wrapper" script
- Process a "chunk" of work, perhaps a file, when script is invoked
- Script can read/write any number of files
- Avoid unnecessary network I/O
- Submit script to cluster for execution on node selected by Sun Grid Engine
- Consider submission script for documentation

```
qsub -cwd -o output.txt -j yes -N "name" wrapper.bash [parms]
```

Standard Out and Standard Error will be written to output.txt.
Use `-e error.txt` (with default `-j no`) to capture Standard Error.
Sun Grid Engine

qsub

Submit array job to cluster

- Wrapper script to perform task
- Sun Grid Engine sets up environment variable $SGE_TASK_ID with numeric value
- Script decides what work to do based on $SGE_TASK_ID environment variable

qsub -cwd -t 1-4 -N "ArrayJob" arrayjob.bash

- t start-stop:increment, start > 0
** CAUTION **

- Can submit job to various queues, but not all queues are equal right now.

- Unfortunately, "q -all.q" is not consistently shown in examples here.

- For now, always use -q all.q with qsub. 
  qsub -q all.q -cwd -t 1-4 -N "ArrayJob" arrayjob.bash
Sun Grid Engine

qstat and qdel

- List your jobs in SGE queue (login ID xxx)
  \texttt{qstat -u xxx}

- Delete all your jobs in SGE queue
  \texttt{qdel -u xxx}

\texttt{qmon} is an alternative
R Batch Jobs on Cluster

"Toy" Examples to Show Concepts
  • SimpleGraph Example
  • ArrayJob Example

Overview of Research Project
Time Perturbation Somitogenesis Studies
R Batch Job on Cluster

SimpleGraph Example

Modify batch script: batch.bash

```bash
[56 17Dec07 16:42:35 /home/efg/cluster/R/simplegraph]
cat batch.bash
#!/bin/bash
# efg, 27 Sept 2007
set -x

simplegraph.bash mouse 2.0 120 red
simplegraph.bash zebrafish 0.5 30 blue
```

Submission Script: submit.bash

```bash
[30 18Dec07 10:51:54 /home/efg/cluster/R/simplegraph]
cat submit.bash
#!/bin/bash
# efg, 27 Sept 2007

qsub -cwd simplegraph1.bash mouse 2.0 120 red
qsub -cwd simplegraph1.bash zebrafish 0.5 30 blue

"Mouse,Expression"
"Zebrafish,Expression"
```

Fix Editing in Windows

```bash
[31 18Dec07 10:57:35 /home/efg/cluster/R/simplegraph]
dos2unix submit.bash
dos2unix: converting file submit.bash to UNIX format ...
```
SimpleGraph Example

Job Script: simplegraph1.bash

```
[34 18Dec07 11:04:05 /home/efg/cluster/R/simplegraph]
cat simplegraph1.bash
#!/bin/bash
set -x
# efg, 27 Sept 2007
#
# Assume script is called with these parameters:
# simplegraph,bash RunName Amplitude Period Color ChartTitle
# $1 $2 $3 $4 $5
#
# where RunName and Color are assumed to be strings, and Amplitude and Period are numeric.

R CMD BATCH --vanilla --slave
  "--args Filename='$1' Amplitude=$2 Period=$3 Color='$4' ChartTitle='$5'"
  simplegraph1.R $1.txt
```
R Batch Job on Cluster

SimpleGraph Example

R Script: simplegraph1.R

```
> ChartTitle <- "Ideal Gene Expression"
---
> ChartTitle <- "Ideal.Gene.Expression"
> # Fixup for "." separators in title
> ChartTitle <- gsub("\\."," ", ChartTitle)
>```

[37 18Dec07 11:08:12 /home/efg/cluster/R/simplegraph]
diff simplegraph.R simplegraph1.R
11c11
< ChartTitle <- "Ideal Gene Expression"
---
> ChartTitle <- "Ideal.Gene.Expression"
> # Fixup for "." separators in title
> ChartTitle <- gsub("\\."," ", ChartTitle)
>```
Run submit.bash to submit jobs to cluster:
1. Login to Cluster Head Node
2. Change to working directory
3. Execute the submit.bash script

```bash
[23 18Dec07 11:21:35 /home/efg]
  cd cluster/R/simplegraph

[29 18Dec07 11:23:14 /home/efg/cluster/R/simplegraph]
  ./submit.bash
  Your job 373170 ("simplegraph1.bash mouse 2.0 120 red Mouse.Expression") has been submitted
  Your job 373171 ("simplegraph1.bash zebrafish 0.5 30 blue Zebrafish.Expression") has been submitted

[30 18Dec07 11:23:24 /home/efg/cluster/R/simplegraph]
  qstat -u efg
  job-ID  prior  name       user  state submit/start at queue slots     ja-task-ID
  ------------------ -------------------------- ----------------- --------
  373170  0.00000 simplegrap efg  qw  12/18/2007 11:23:24
  373171  0.00000 simplegrap efg  qw  12/18/2007 11:23:24

Delete all jobs: qdel -u efg
R Batch Job on Cluster

**SimpleGraph Example**

**Output files:**

Standard Out and Standard Error Files have no useful information and can be deleted.
R Batch Job on Cluster

R Array Job Example

Submission Script: submit.bash

```
[47 18Dec07 12:19:57 /home/efg/cluster/R/array]
cat submit.bash
#!/bin/bash
# efg, 27 Sept 2007
qsub -t 1-4 -cwd -N ArrayJob arrayjob.bash
```

Job Script: arrayjob.bash

```
[48 18Dec07 12:20:00 /home/efg/cluster/R/array]
cat arrayjob.bash
#!/bin/bash
# efg, 27 Sept 2007
R CMD BATCH --vanilla --slave arrayjob.R Rout-\$JOB_ID-\$SGE_TASK_ID.txt
```

R Script: arrayjob.R

```
[49 18Dec07 12:20:25 /home/efg/cluster/R/array]
cat arrayjob.R
options(echo=TRUE)
taskid <- Sys.getenv("SGE_TASK_ID")
cat("Task ID = ", taskid, 
```
R Batch Job on Cluster

Submit Array Job

```
[50] 18Dec07 12:20:30 /home/efg/cluster/R/array
submit_bash
Your job-array 373173.1-4:1 ("ArrayJob") has been submitted
```

```
qstat -u efg
```

```
job-ID  prior name user      state      submit/start at       queue slots  ja-task-ID
373173  0.00000 ArrayJob  efg  qu 12/18/2007 12:20:43
```

Output files

```
Address: \\hask\0\unix\homes\efg\cluster\R\array
```

```
Name                     Size     Type                      Date Modified
----------------------------------------------------------------------------------
 Rout-373173-4.txt       1 KB     Text Document              12/18/2007 12:21 PM
 Rout-373173-3.txt       1 KB     Text Document              12/18/2007 12:21 PM
 Rout-373173-2.txt       1 KB     Text Document              12/18/2007 12:21 PM
 Rout-373173-1.txt       1 KB     Text Document              12/18/2007 12:21 PM
 ArrayJob.o373173.4      0 KB     File                      12/18/2007 12:21 PM
 ArrayJob.e373173.4      0 KB     File                      12/18/2007 12:21 PM
 ArrayJob.o373173.3      0 KB     File                      12/18/2007 12:21 PM
 ArrayJob.o373173.2      0 KB     File                      12/18/2007 12:21 PM
 ArrayJob.o373173.1      0 KB     File                      12/18/2007 12:21 PM
 ArrayJob.e373173.3      0 KB     File                      12/18/2007 12:21 PM
 ArrayJob.e373173.2      0 KB     File                      12/18/2007 12:21 PM
 ArrayJob.e373173.1      0 KB     File                      12/18/2007 12:21 PM
```

Standard Out and Standard Error Files are zero-length and can be deleted.
Output files

Cannot create JPEGs or PNGs on cluster nodes for now, but PDFs will work for graphical output.
Time Perturbation Somitogenesis Studies

R Batch Job on Cluster

Time Perturbation Study: Controlled Experiments, Gaussian Experiment, Chick, Mouse, Zebrafish

Dataflow Diagram

- **LogExpression.csv**
- **Perturbations.txt**
- **RandomLogExpression.csv**

**Microarray Experiment or**
- [0] CreateControlledExperiment.bash

**CreateTimePerturbations.R**

**RandomizeLogExpression.R**
- [0] RandomizeLogExpression.bash

**LombScargleTimeVariation**

**LogExpression.csv**

**UpdateGeneList.R**
- [0] Manually, R in Windows

**RankProduct.R**
- [2] RankProduct.bash

**RankOrder-perturb.csv**

**LombScargleTimeVariation**

**/random/00001.csv ... 10000.csv**

**perturb-SpreadHistograms.pdf**
- perturb-Analysis-period.csv
- perturb-Analysis-p.csv

**PerturbationStats.R**
- [4] PerturbationStats.bash

**perturb-stats.csv**

**<species>-compareN.pdf**

**CompareN.R**

**PerturbAnalysis.R**
- [3] PerturbAnalysis.bash

**random-SpreadHistograms.pdf**
- random-Analysis-period.csv
- random-Analysis-p.csv

**RandomLogExpression.csv**

For comparison purposes.
R Batch Job on Cluster

Time Perturbation Somitogenesis Studies

submit.bash

submit-batch.bash & submit-batch.bash &

Compare250.bash

See files in /home/efg/cluster/TimePerturb/mouse17
Questions?