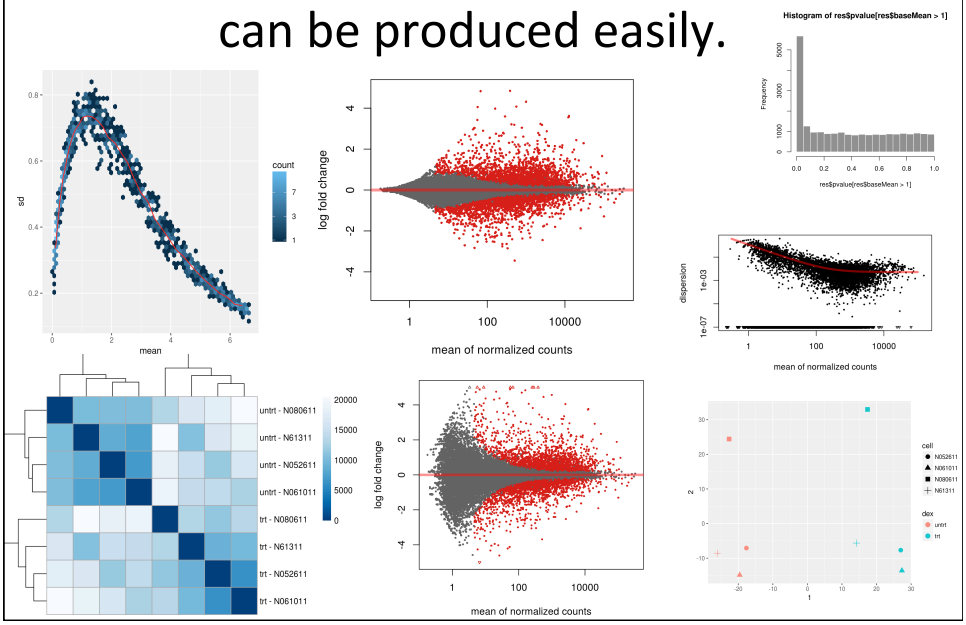
R and DEseq on Fiji

What is R?

- *R* is a language and environment for statistical computing and graphics.
- R can be extended (easily) via *packages*. Packages provide both new functionality (software) and access to public datasets.
- It's free, open source and complies on many platforms. (For example, I have copies on my laptop!)

Well-designed publication-quality plots can be produced easily.



On Fiji …

- Technically *THREE* versions of R:

1. https://fiji-viz.colorado.edu/rstudio/auth-sign-in
   R version 3.4.1

2. Module load R/3.3.0 (*legacy and deprecated*)

3. Base R (e.g. NO MODULE LOAD)
   R version 3.4.3

Interactive R on head node.

- PROs:
  - Simple and interactive
  - Most up to date version of R
  - Would be the same "environment" as BATCH mode (for package installs, commands, etc) so good for testing.
- CONs:
  - No windowing system (e.g. must capture graphs to files)
  - On head node, so want to be conscious of resources (compute) utilized.

Interactive Rstudio

- PROs:
  - Windowing system (ease of use)
  - Great for exploratory data analysis
- CONs:
  - Cannot use in BATCH mode (no parallelization)
  - Different version than Fiji (impacts packages)

Batch R via SLURM

- PRO
  - Permits large scale parallelization (many R instances running in parallel)
  - Plays friendly with the rest of folks on the cluster (i.e. not constantly on head node)
  - Scripts used become “self documenting”
- CON
  - A little clunky to write R scripts and Slurm wrappers
  - No windowing environment.

Library/Package installations are local to YOUR home directory

```
-bash-4.2$ R
```

… bunch of stuff removed here …

Type 'demo()' for some demos, 'help()' for on-line help, or 'help.start()' for an HTML browser interface to help.

Type 'q()' to quit R.

```
> source("https://bioconductor.org/biocLite.R")
```

Lets R access an online library of packages, in this case from Bioconductor.

```
> biocLite()
```

This installs the base package of Bioconductor and thereby enables the “automatic install” features of biocLite. This takes TIME.

```
> biocLite("DESeq")
```

This installs DESeq. This takes TIME. Also or alternatively can install DESeq2 which is the newest version but a different package.

Unfortunately …

- R version on the viz interface (3.4.1) is not the same as the R interpreter installed on the head nodes (3.4.3). So beware, you can get all kinds of warnings if you build a library on one and try to use it on the other.

There are technical reasons this is not easy to fix, that said … IT is working on it.

The R environment

- The simplest way to use R is via the R environment prompt (“>”) on the head node.
- By default it knows “where” (in directory space) you started R and saves/looks for files there.
- Similar to man pages, R has a built in help:
  > help(solve)
- You can quit the environment with q() or quit().

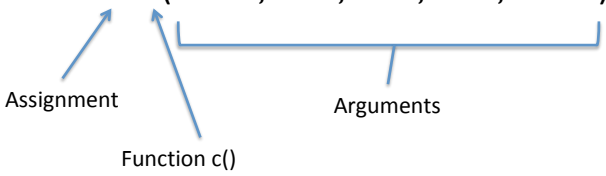
Basic R interactions.

- Technically R is an expression language with a very simple syntax. It is *case sensitive*.
- Elementary commands consist of either expressions or assignments.
- Commands are separated either by a semi-colon (';'), or by a newline.
- Comments can be put anywhere (begin "#").
- Commands can continue to a second line by ending the first line with a "+".

R operates on named data structures.

- Vectors

```
X <- c(10.4, 5.6, 3.1, 6.4, 21.7)
```



- Other types of Objects: Matricies, Factors, Lists, Data Frames, Functions.

There are many good, free resources on R online.

- https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf
- https://www.datacamp.com/courses/free-introduction-to-r
- http://www.r-tutor.com/r-introduction

DESeq vs DESeq2

- The basic tutorial on DESeq:

https://bioconductor.org/packages/release/bioc/vignettes/DESeq/inst/doc/DESeq.pdf

- A good tutorial on DESeq2:

https://bioconductor.org/packages/3.7/bioc/vignettes/DESeq2/inst/doc/DESeq2.html

DESeq2 is the most up to date version, but DESeq is still preferred in some scenarios.

DEseq/DEseq2 requires two kinds of data:

1. A 2D matrix of count/summary data across all samples.
   - The value in the *i*-th row and the *j*-th column of the matrix tells how many reads were assigned to gene *i* in sample *j*
   - Typically un-normalized (DEseq handles normalization)
2. The best data are useless without metadata, therefore you also need an associated *design formula*. This effectively describes the structure of the experiment.

An aside: Counting reads per gene

- Bedtools
  - PRO: it's simple to run and works on any coordinate list
  - CON: it's "stupid" with respect to what those coordinates might mean
- HTseq
  - PRO: It recognizes .gtf and .gff3 annotation files and can properly count (exons only)
  - CON: It's a little more involved to run (not much)

Running R in batch mode

(aka R scripts)

- File.Rscript
  - One command per line (similar to SLURM) of what you would like R to execute
  - Capture output to files from within R!!

```
sink('thisjob.Routput')
png('file.png')
plotDE( res )
dev.off()
write.table(tableVar, filename, append, separator)
sink()
```

Must then submit the R job to SLURM within a wrapper script.

Wrapper script example.slurm

```
#!/bin/bash
#SBATCH --job-name=Rtest # Job name
#SBATCH --mail-type=NONE# Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=username@colorado.edu # Where to send mail
#SBATCH --nodes=1 # Run on a single node
#SBATCH --ntasks=1     # Number of CPU (processer cores i.e. tasks) In this example I use 1. I only need one, since none of the commands I run are parallelized.
#SBATCH --mem=1gb # Memory limit
#SBATCH --time=01:00:00 # Time limit hrs:min:sec
#SBATCH --output=/scratch/Users/dowellde/eofiles/R.%j.out # Standard output
#SBATCH --error=/scratch/Users/dowellde/eofiles/R.%j.err # Standard error log


R CMD BATCH --no-save --no-restore exampleRscript.R
```