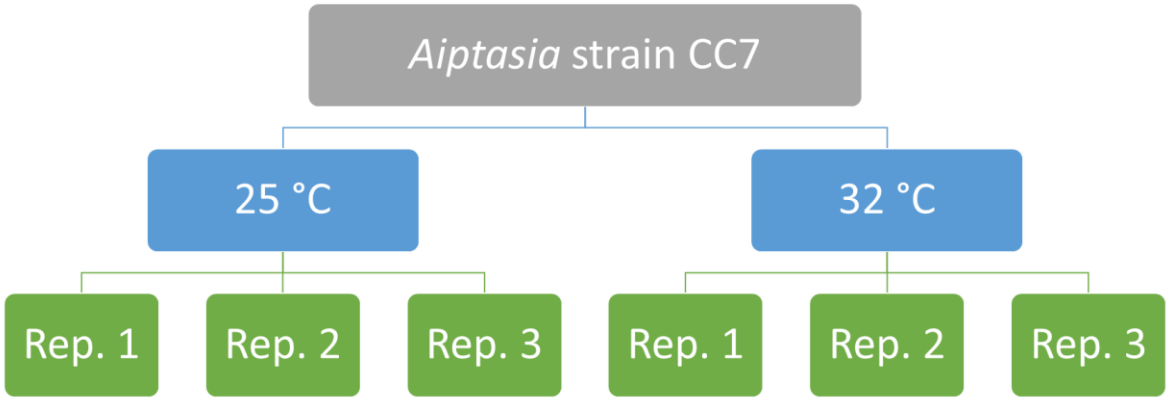
Stats in R: Day 4, hands-on session

Transcriptomics, from raw reads, to real results!

Note: all commands are case-sensitive. “ssh” will work; “Ssh” would not. If you have weird crashes, double check whether you have typed the commands exactly as shown.

0. Understanding the experimental setup, and what we’re doing

We’re dealing with a heat stress experiment performed on Aiptasia strain CC7. The normal temperature is 25 °C, the stress temperature is 32 °C. There are three replicates per temperature.



We’re interested in seeing which genes are differentially expressed under heat stress (using kallisto and sleuth).

Once we have a bunch of differentially expressed genes, we want to see what kind of biological function is associated with these genes, i.e. which genes are heat-stress-related genes? We use topGO for this.

kallisto is written in C and C++; sleuth and topGO are written in R (which helps me fulfil the R in “Stats in R”!)

1. Logging in to a server via the command line

On Mac OS X, it’s called “Terminal”.

Type

```
ssh stats@lithium.kaust.edu.sa
```

then press ENTER.

Type

```
yes
```

in the “Are you sure you want to continue connecting”, then press ENTER.

Type

stats

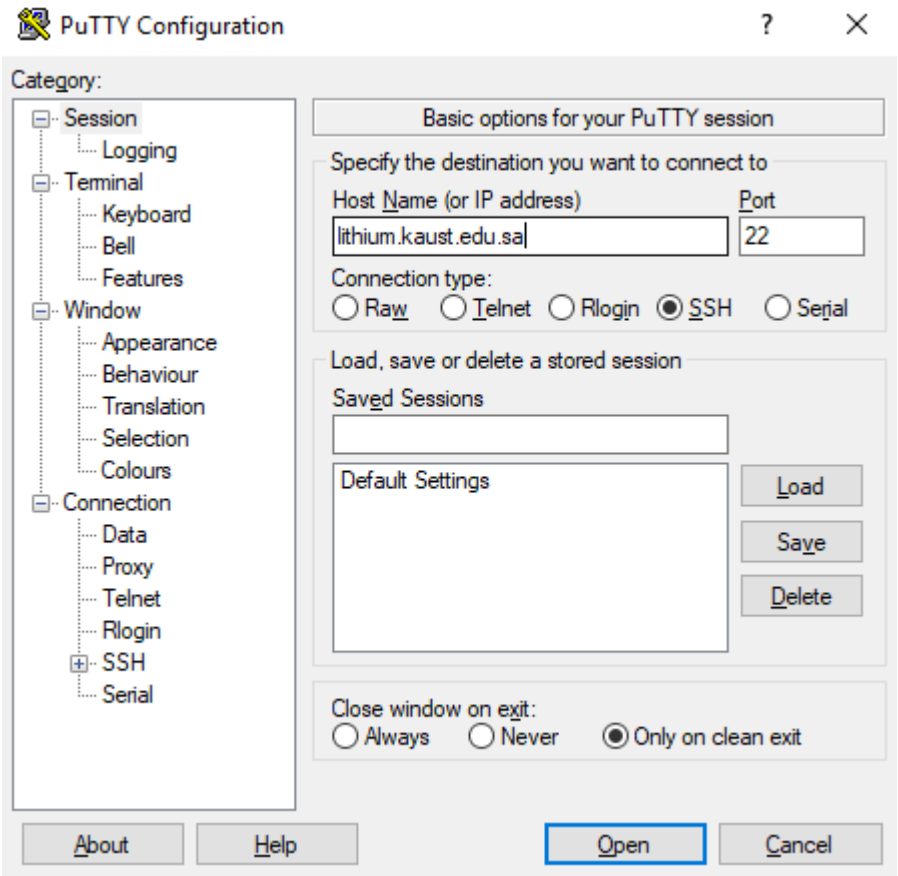
in the password column (without the quotes). You won't see the cursor moving, that's normal. Press ENTER.

```
liewy@Coral:~$ ssh stats@lithium.kaust.edu.sa
The authenticity of host 'lithium.kaust.edu.sa (10.74.186.113)' can't be established.
ECDSA key fingerprint is bf:ed:3e:1e:f4:9b:ae:fe:46:99:cc:d7:6e:fe:b4:bf.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'lithium.kaust.edu.sa,10.74.186.113' (ECDSA) to the list of known hosts.
stats@lithium.kaust.edu.sa's password:
Linux kw14764 4.14.0-1-amd64 #1 SMP Debian 4.14.2-1 (2017-11-30) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Jan 28 17:45:05 2018 from 10.74.186.114
stats@kw14764:~$
```

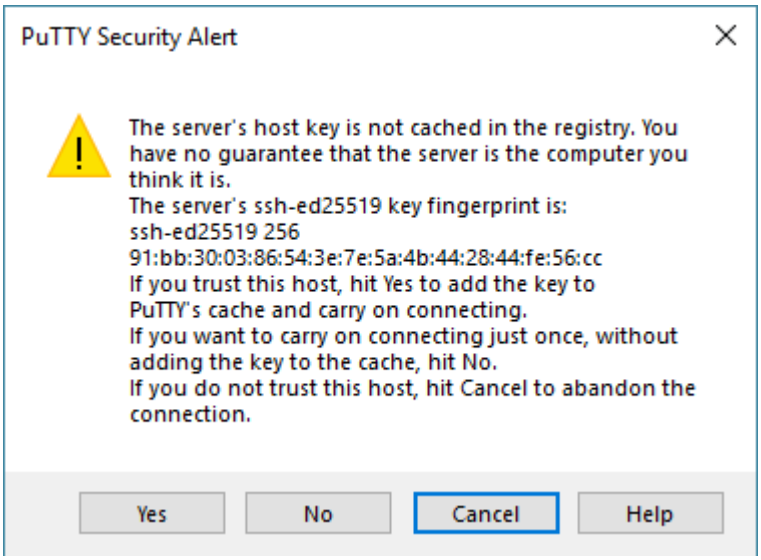
You're done here, skip the Windows-specific stuff below and go to (2).

On Windows, it's called "PuTTY".

Fill in the hostname (lithium.kaust.edu.sa), then click Open.



Say Yes to this.



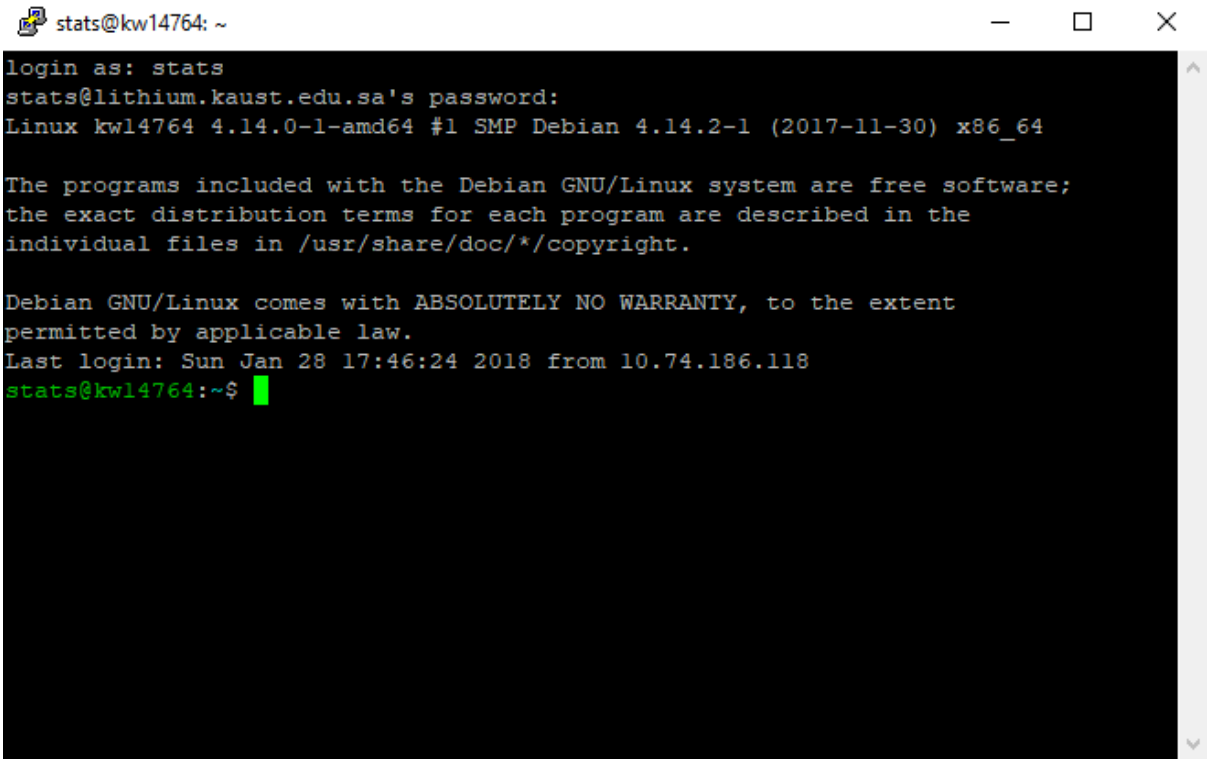
Type

stats

in “login as”, press ENTER; then type

stats

in the password field, and press ENTER.



2. Make a copy of the example files provided to your own personal folder

`cp -r example/ <your KAUST username>`

So, as my username is “liewy”, I’ll write

stats@kw14764: ~

```
stats@kw14764:~$ cp -r example/ liewy
stats@kw14764:~$
```

Then enter your own directory

`cd <your KAUST username>`

stats@kw14764: ~/liewy

```
stats@kw14764:~$ cp -r example/ liewy
stats@kw14764:~$ cd liewy/
stats@kw14764:~/liewy$
```

3. Exploring your data (briefly)

`ls` shows you the contents of your folder.

stats@kw14764: ~/liewy

```
stats@kw14764:~$ cp -r example/ liewy
stats@kw14764:~$ cd liewy
stats@kw14764:~/liewy$ ls
aiptasia_cds.fa  ngs_reads/  sleuth/  topgo/
stats@kw14764:~/liewy$
```

Hmm, let's look at the contents of aiptasia_cds.fa.

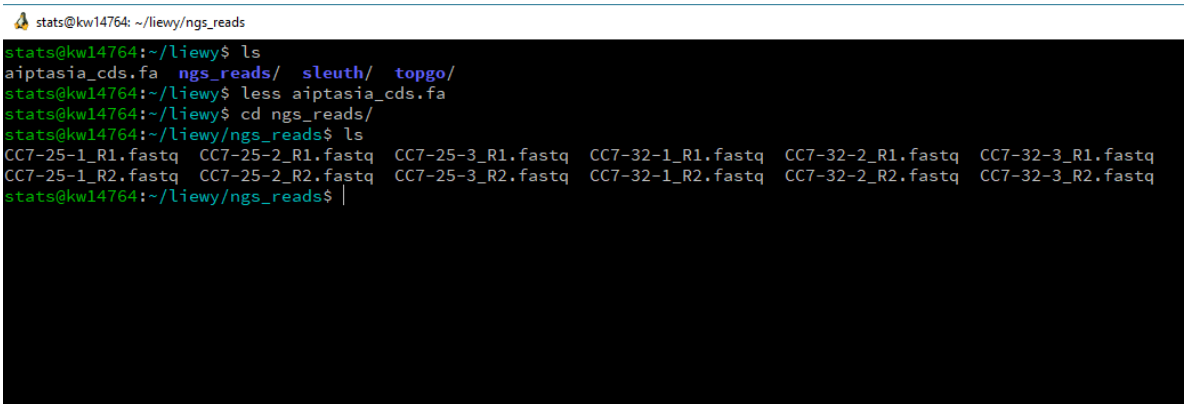
`less aiptasia_cds.fa`

This gives you a brief look at your data. Press **q** to get out ("quit").

Let's look at the folder "ngs_reads".

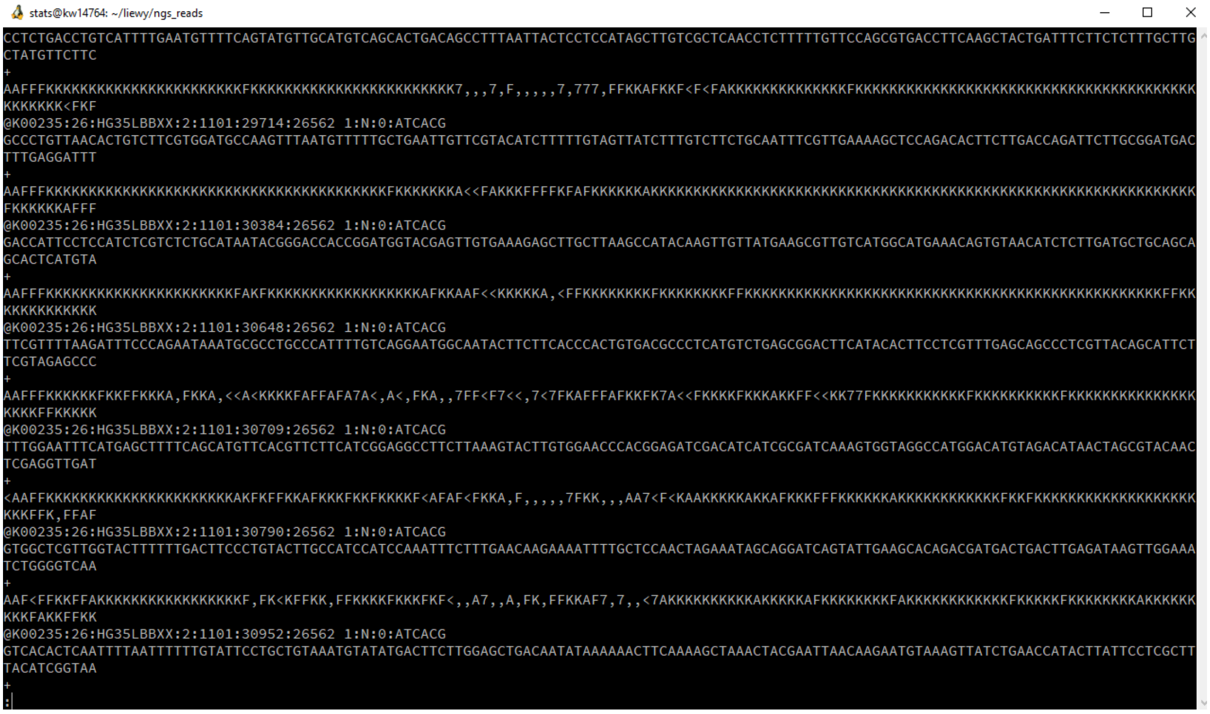
`cd ngs_reads`

`ls`



Ooh, FASTQs. How does a FASTQ file look like?

`less CC7-25-1_R1.fastq`



```
CCTCTGACCTGTCATTTTGAATGTTTTCAGTATGTTGCATGTCAGCACTGACAGCCTTTAATTACTCCTCCATAGCTTGTCGCTCAACCTCTTTTTGTTCCAGCGTGACCTTCAAGCTACTGATTTCTTCTCTTTGCTTG
CTATGTTCTTC
+
AAFFFKKKKKKKKKKKKKKKKKKKKKKKFKKKKKKKKKKKKKKKKKKKKKKKK7,,,7,F,,,,,7,777,FFKKAFKKF<F<FAKKKKKKKKKKKKKKFKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKK
KKKKKKK<FKF
@K00235:26:HG35LBBXX:2:1101:29714:26562 1:N:0:ATCACG
GCCCTGTTAACACTGTCTTCGTGGATGCCAAGTTTAATGTTTTTGCTGAATTGTTCGTACATCTTTTTGTAGTTATCTTTGTCTTCTGCAATTTCGTTGAAAAGCTCCAGACACTTCTTGACCAGATTCTTGCGGATGAC
TTTGAGGATTT
+
AAFFFKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKFKKKKKKKA<<FAKKKFFFFKFAFKKKKKAKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKK
FKKKKKKAFFF
@K00235:26:HG35LBBXX:2:1101:30384:26562 1:N:0:ATCACG
GACCATTCCTCCATCTCGTCTCTGCATAATACGGGACCACCGGATGGTACGAGTTGTGAAAGAGCTTGCTTAAGCCATACAAGTTGTTATGAAGCGTTGTCATGGCATGAAACAGTGTAACATCTCTTGATGCTGCAGCA
GCACTCATGTA
+
AAFFFKKKKKKKKKKKKKKKKKKKKKKFAKFKKKKKKKKKKKKKKKKKKAFKKAAF<<KKKKKA,<FFKKKKKKKKFKKKKKKKKFFKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKFFKK
KKKKKKKKKKK
@K00235:26:HG35LBBXX:2:1101:30648:26562 1:N:0:ATCACG
TTCGTTTTAAGATTTCCCAGAATAAATGCGCCTGCCCATTTTGTCAGGAATGGCAATACTTCTTCACCCACTGTGACGCCCTCATGTCTGAGCGGACTTCATACACTTCCTCGTTTGAGCAGCCCTCGTTACAGCATTCT
TCGTAGAGCCC
+
AAFFFKKKKKKFKKFFKKKA,FKKA,<<A<KKKKFAFFAFA7A<,A<,FKA,,7FF<F7<<,7<7FKAFFFAFKKFK7A<<FKKKKFKKKAKKFF<<KK77FKKKKKKKKKKKFKKKKKKKKKKKFKKKKKKKKKKKKKK
KKKKFFKKKKK
@K00235:26:HG35LBBXX:2:1101:30709:26562 1:N:0:ATCACG
TTTGGAATTTCATGAGCTTTTCAGCATGTTCACGTTCTTCATCGGAGGCCTTCTTAAAGTACTTGTGGAACCCACGGAGATCGACATCATCGCGATCAAAGTGGTAGGCCATGGACATGTAGACATAACTAGCGTACAAC
TCGAGGTTGAT
+
<AAFFKKKKKKKKKKKKKKKKKKKKAKFKFFKKAFKKKFKKFKKKKF<AFAF<FKKA,F,,,,,7FKK,,,AA7<F<KAAKKKKKAKKAFKKKFFFKKKKKKAKKKKKKKKKKKKKKFKKFKKKKKKKKKKKKKKKKKKK
KKKFFK,FFAF
@K00235:26:HG35LBBXX:2:1101:30790:26562 1:N:0:ATCACG
GTGGCTCGTTGGTACTTTTTTGACTTCCCTGTACTTGCCATCCATCCAAATTTCTTTGAACAAGAAAATTTTGCTCCAACTAGAAATAGCAGGATCAGTATTGAAGCACAGACGATGACTGACTTGAGATAAGTTGGAAA
TCTGGGGTCAA
+
AAF<FFKKFFAKKKKKKKKKKKKKKKKKF,FK<KFFKK,FFKKKKFKKKFKF<,,A7,,A,FK,FFKKAF7,7,,<7AKKKKKKKKKKAKKKKKAFKKKKKKKKFAKKKKKKKKKKKFKKKKKFKKKKKKKKKAKKKKKK
KKKFAKKFFKK
@K00235:26:HG35LBBXX:2:1101:30952:26562 1:N:0:ATCACG
GTCACACTCAATTTTAATTTTTTGTATTCCTGCTGTAAATGTATATGACTTCTTGGAGCTGACAATATAAAAAACTTCAAAAGCTAAACTACGAATTAACAAGAATGTAAAGTTATCTGAACCATACTTATTCCTCGCTT
TACATCGGTAA
+
:
```

Yep, this is what bioinformaticists mean when they “deal with NGS data”. This data was generated from a next-generation sequencer (Illumina 2000 for this case). Whee.

Again, **q** to quit.

Let’s do a quick count of the number of lines of your files.

```
wc -l *
```

wc = “word count”

-l = “number of lines”. This is what we call a “flag”. Flags are optional parameters, and flags are specific to your program. -l might mean something else in another program.

* = “all files”

```
stats@kw14764: ~/liewy/ngs_reads
stats@kw14764:~/liewy$ ls
aiptasia_cds.fa  ngs_reads/  sleuth/  topgo/
stats@kw14764:~/liewy$ less aiptasia_cds.fa
stats@kw14764:~/liewy$ cd ngs_reads/
stats@kw14764:~/liewy/ngs_reads$ ls
CC7-25-1_R1.fastq  CC7-25-2_R1.fastq  CC7-25-3_R1.fastq  CC7-32-1_R1.fastq  CC7-32-2_R1.fastq  CC7-32-3_R1.fastq
CC7-25-1_R2.fastq  CC7-25-2_R2.fastq  CC7-25-3_R2.fastq  CC7-32-1_R2.fastq  CC7-32-2_R2.fastq  CC7-32-3_R2.fastq
stats@kw14764:~/liewy/ngs_reads$ wc -l *
   40000 CC7-25-1_R1.fastq
   40000 CC7-25-1_R2.fastq
   40000 CC7-25-2_R1.fastq
   40000 CC7-25-2_R2.fastq
   40000 CC7-25-3_R1.fastq
   40000 CC7-25-3_R2.fastq
   40000 CC7-32-1_R1.fastq
   40000 CC7-32-1_R2.fastq
   40000 CC7-32-2_R1.fastq
   40000 CC7-32-2_R2.fastq
   40000 CC7-32-3_R1.fastq
   40000 CC7-32-3_R2.fastq
  480000 total
stats@kw14764:~/liewy/ngs_reads$
```

Each file has 40,000 lines, i.e. 10,000 reads (as each NGS read occupies 4 lines in the file).

(The real files have ~10,000,000 reads, but it's huge, so I cut out a small portion of the real thing.)

OK, that's all, let's go back to the previous folder.

cd ..

```
stats@kw14764: ~/liewy
stats@kw14764:~/liewy$ ls
aiptasia_cds.fa  ngs_reads/  sleuth/  topgo/
stats@kw14764:~/liewy$ less aiptasia_cds.fa
stats@kw14764:~/liewy$ cd ngs_reads/
stats@kw14764:~/liewy/ngs_reads$ ls
CC7-25-1_R1.fastq  CC7-25-2_R1.fastq  CC7-25-3_R1.fastq  CC7-32-1_R1.fastq  CC7-32-2_R1.fastq  CC7-32-3_R1.fastq
CC7-25-1_R2.fastq  CC7-25-2_R2.fastq  CC7-25-3_R2.fastq  CC7-32-1_R2.fastq  CC7-32-2_R2.fastq  CC7-32-3_R2.fastq
stats@kw14764:~/liewy/ngs_reads$ wc -l *
   40000 CC7-25-1_R1.fastq
   40000 CC7-25-1_R2.fastq
   40000 CC7-25-2_R1.fastq
   40000 CC7-25-2_R2.fastq
   40000 CC7-25-3_R1.fastq
   40000 CC7-25-3_R2.fastq
   40000 CC7-32-1_R1.fastq
   40000 CC7-32-1_R2.fastq
   40000 CC7-32-2_R1.fastq
   40000 CC7-32-2_R2.fastq
   40000 CC7-32-3_R1.fastq
   40000 CC7-32-3_R2.fastq
  480000 total
stats@kw14764:~/liewy/ngs_reads$ cd ..
stats@kw14764:~/liewy$
```

4. Trimming reads

Most older differential expression packages (DESeq2, edgeR, baySeq, ...) requires you to trim adapter sequences from reads.

Nowadays, new packages allow you skip this step. Fortunately, we're using one (kallisto) that allows us to skip this, so skip this we will!

I left this section in just in case your colleagues/bioinformatician/PI prefer the older packages. If you need to trim adapters, look into TrimGalore or trimmomatic.

5. Running kallisto

"kallisto" is the package that calculates relative frequencies of the transcripts. The unit of measurement is "tpm": **t**ranscripts **p**er **m**illion sequenced transcripts.

(Just a quick illustration: if 20 sequenced reads out of 2 million reads maps to Gene X, the gene has a tpm value of 10, i.e. 10 reads per 1 million reads.)

5a: how do I see what kallisto can do?

There's an online manual (https://pachterlab.github.io/kallisto/manual), or... type kallisto with no arguments. Well-written programs usually give you hints on how to run it.

```
kallisto
```

stats@kw14764: ~/liewy

```
stats@kw14764:~/liewy$ kallisto
kallisto 0.43.1

Usage: kallisto <CMD> [arguments] ..

Where <CMD> can be one of:

    index         Builds a kallisto index
    quant         Runs the quantification algorithm
    pseudo        Runs the pseudoalignment step
    h5dump        Converts HDF5-formatted results to plaintext
    version       Prints version information
    cite          Prints citation information

Running kallisto <CMD> without arguments prints usage information for <CMD>

stats@kw14764:~/liewy$
```

Note the line "Running kallisto <CMD> without arguments prints usage information for <CMD>". This means that help's always at hand!

5b: run kallisto index

To get more info on how to run kallisto index, run it with no arguments.

```
kallisto index
```

stats@kw14764: ~/liewy

```
stats@kw14764:~/liewy$ kallisto index

kallisto 0.43.1
Builds a kallisto index

Usage: kallisto index [arguments] FASTA-files

Required argument:
-i, --index=STRING          Filename for the kallisto index to be constructed

Optional argument:
-k, --kmer-size=INT         k-mer (odd) length (default: 31, max value: 31)
    --make-unique           Replace repeated target names with unique names

stats@kw14764:~/liewy$
```

A-ha. Run

`kallisto index -i aip_cds aiptasia_cds.fa`

stats@kw14764: ~/liewy

```
stats@kw14764:~/liewy$ kallisto index

kallisto 0.43.1
Builds a kallisto index

Usage: kallisto index [arguments] FASTA-files

Required argument:
-i, --index=STRING          Filename for the kallisto index to be constructed

Optional argument:
-k, --kmer-size=INT         k-mer (odd) length (default: 31, max value: 31)
    --make-unique           Replace repeated target names with unique names

stats@kw14764:~/liewy$ kallisto index -i aip_cds aiptasia_cds.fa

[build] loading fasta file aiptasia_cds.fa
[build] k-mer length: 31
[build] warning: clipped off poly-A tail (longer than 10)
        from 13 target sequences
[build] warning: replaced 1271540 non-ACGUT characters in the input sequence
        with pseudorandom nucleotides
[build] counting k-mers ...
```

Let this run for a bit, about 5 mins or so. Toilet break!

When it finishes, you should see a new file called “aip_cds”.

`ls`

```
stats@kw14764: ~/liewy
stats@kw14764:~/liewy$ kallisto index

kallisto 0.43.1
Builds a kallisto index

Usage: kallisto index [arguments] FASTA-files

Required argument:
-i, --index=STRING          Filename for the kallisto index to be constructed

Optional argument:
-k, --kmer-size=INT         k-mer (odd) length (default: 31, max value: 31)
    --make-unique           Replace repeated target names with unique names

stats@kw14764:~/liewy$ kallisto index -i aip_cds aiptasia_cds.fa

[build] loading fasta file aiptasia_cds.fa
[build] k-mer length: 31
[build] warning: clipped off poly-A tail (longer than 10)
        from 13 target sequences
[build] warning: replaced 1271540 non-ACGUT characters in the input sequence
        with pseudorandom nucleotides
[build] counting k-mers ... done.
[build] building target de Bruijn graph ...  done
[build] creating equivalence classes ...  done
[build] target de Bruijn graph has 454656 contigs and contains 62398097 k-mers

stats@kw14764:~/liewy$ ls
aip_cds  aiptasia_cds.fa  ngs_reads/
stats@kw14764:~/liewy$
```

Done!

5c: Create empty folders to contain results

To create these folders, there's a hardworking (but easier to understand) way and a lazy (but harder to understand) way. Both ways lead to Rome.

EITHER

```
mkdir results results/CC7-25-1 results/CC7-25-2 results/CC7-25-3
results/CC7-32-1 results/CC7-32-2 results/CC7-32-3
```

```
stats@kw14764: ~/liewy
stats@kw14764:~/liewy$ ls
aip_cds  aiptasia_cds.fa  ngs_reads/
stats@kw14764:~/liewy$ mkdir results results/CC7-25-1 results/CC7-25-2 results/CC7-25-3 results/CC7-32-1 results/CC7-32-2 results/CC7-32-3
stats@kw14764:~/liewy$ ls
aip_cds  aiptasia_cds.fa  ngs_reads/  results/
stats@kw14764:~/liewy$ ls results/
CC7-25-1/  CC7-25-2/  CC7-25-3/  CC7-32-1/  CC7-32-2/  CC7-32-3/
stats@kw14764:~/liewy$
```

OR

```
mkdir results && for a in 25 32; do for b in 1 2 3; do mkdir results/CC7-
${a}-${b}; done; done
```

stats@kw14764: ~/liewy

```
stats@kw14764:~/liewy$ ls
aip_cds  aiptasia_cds.fa  ngs_reads/
stats@kw14764:~/liewy$ mkdir results && for a in 25 32; do for b in 1 2 3; do mkdir results/CC7-${a}-${b}; done; done
stats@kw14764:~/liewy$ ls
aip_cds  aiptasia_cds.fa  ngs_reads/  results/
stats@kw14764:~/liewy$ ls results/
CC7-25-1/  CC7-25-2/  CC7-25-3/  CC7-32-1/  CC7-32-2/  CC7-32-3/
stats@kw14764:~/liewy$
```

5d: calculate TPMs via kallisto quant

Same as previous, hardworking vs. lazy.

EITHER

Run six commands, one after another

```
kallisto quant -i aip_cds -o results/CC7-25-1 --bias --rf-stranded -b 100 ngs_reads/CC7-25-1_R1.fastq ngs_reads/CC7-25-1_R2.fastq
```

```
kallisto quant -i aip_cds -o results/CC7-25-2 --bias --rf-stranded -b 100 ngs_reads/CC7-25-2_R1.fastq ngs_reads/CC7-25-2_R2.fastq
```

```
kallisto quant -i aip_cds -o results/CC7-25-3 --bias --rf-stranded -b 100 ngs_reads/CC7-25-3_R1.fastq ngs_reads/CC7-25-3_R2.fastq
```

```
kallisto quant -i aip_cds -o results/CC7-32-1 --bias --rf-stranded -b 100 ngs_reads/CC7-32-1_R1.fastq ngs_reads/CC7-32-1_R2.fastq
```

```
kallisto quant -i aip_cds -o results/CC7-32-2 --bias --rf-stranded -b 100 ngs_reads/CC7-32-2_R1.fastq ngs_reads/CC7-32-2_R2.fastq
```

```
kallisto quant -i aip_cds -o results/CC7-32-3 --bias --rf-stranded -b 100 ngs_reads/CC7-32-3_R1.fastq ngs_reads/CC7-32-3_R2.fastq
```

```
stats@kw14764: ~/liewy
stats@kw14764:~/liewy$ kallisto quant -i aip_cds -o results/CC7-25-1 --bias --rf-stranded -b 100
ngs_reads/CC7-25-1_R1.fastq ngs_reads/CC7-25-1_R2.fastq

[quant] fragment length distribution will be estimated from the data
[index] k-mer length: 31
[index] number of targets: 27,553
[index] number of k-mers: 62,398,097
[index] number of equivalence classes: 165,696
[quant] running in paired-end mode
[quant] will process pair 1: ngs_reads/CC7-25-1_R1.fastq
                             ngs_reads/CC7-25-1_R2.fastq
[quant] finding pseudoalignments for the reads ... done
[quant] learning parameters for sequence specific bias
[quant] processed 10,000 reads, 7,201 reads pseudoaligned
[quant] estimated average fragment length: 189.067
[   em] quantifying the abundances ... done
[   em] the Expectation-Maximization algorithm ran for 521 rounds
[bstrp] running EM for the bootstrap: 100

stats@kw14764:~/liewy$
```

OR

Run one very complex command (it's a loop that runs 6 commands in succession)

`for a in 25 32; do for b in 1 2 3; do kallisto quant -i aip_cds -o results/CC7-${a}-${b} --bias --rf-stranded -b 100 ngs_reads/CC7-${a}-${b}_R1.fastq ngs_reads/CC7-${a}-${b}_R2.fastq; done; done`

```
stats@kw14764: ~/liewy
stats@kw14764:~/liewy$ for a in 25 32; do for b in 1 2 3; do kallisto quant -i aip_cds -o results
/CC7-${a}-${b} --bias --rf-stranded -b 100 ngs_reads/CC7-${a}-${b}_R1.fastq ngs_reads/CC7-${a}-${
b}_R2.fastq; done; done

[quant] fragment length distribution will be estimated from the data
[index] k-mer length: 31
[index] number of targets: 27,553
[index] number of k-mers: 62,398,097
[index] number of equivalence classes: 165,696
[quant] running in paired-end mode
[quant] will process pair 1: ngs_reads/CC7-25-1_R1.fastq
                             ngs_reads/CC7-25-1_R2.fastq
[quant] finding pseudoalignments for the reads ... done
[quant] learning parameters for sequence specific bias
[quant] processed 10,000 reads, 7,201 reads pseudoaligned
[quant] estimated average fragment length: 189.067
[   em] quantifying the abundances ... done
[   em] the Expectation-Maximization algorithm ran for 521 rounds
[bstrp] running EM for the bootstrap: 100


[quant] fragment length distribution will be estimated from the data
[index] k-mer length: 31
[index] number of targets: 27,553
[index] number of k-mers: 62,398,097
```

Done!

6. <u>Running sleuth</u>

Go into the sleuth folder, and look around.

`cd sleuth`

`ls`

stats@kw14764: ~/liewy/sleuth

```
stats@kw14764:~/liewy$ cd sleuth/
stats@kw14764:~/liewy/sleuth$ ls
expt_setup.tsv  sleuth_analysis.R
stats@kw14764:~/liewy/sleuth$
```

I have provided you the code to run the sleuth analysis—if you're curious, feel free to read it (**`less sleuth_analysis.R`**)—but let's just go ahead and run the R script with...

`Rscript sleuth_analysis.R`

stats@kw14764: ~/liewy/sleuth

```
stats@kw14764:~/liewy$ cd sleuth/
stats@kw14764:~/liewy/sleuth$ ls
expt_setup.tsv  sleuth_analysis.R
stats@kw14764:~/liewy/sleuth$ Rscript sleuth_analysis.R
Loading required package: methods
Loading required package: ggplot2
Loading required package: dplyr

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union

  sample condition               path
1   25-1       25C ../results/CC7-25-1
2   25-2       25C ../results/CC7-25-2
3   25-3       25C ../results/CC7-25-3
4   32-1       32C ../results/CC7-32-1
5   32-2       32C ../results/CC7-32-2
6   32-3       32C ../results/CC7-32-3
reading in kallisto results
dropping unused factor levels
......
normalizing est_counts
241 targets passed the filter
normalizing tpm
merging in metadata
summarizing bootstraps
......
fitting measurement error models
shrinkage estimation
computing variance of betas
fitting measurement error models
shrinkage estimation
computing variance of betas
stats@kw14764:~/liewy/sleuth$
```

The R script produces two files. We're interested in one of them.

`ls`

```
stats@kw14764: ~/liewy/sleuth
stats@kw14764:~/liewy/sleuth$ ls
expt_setup.tsv  normalised_abundances.tsv  sleuth_analysis.R  sleuth_results.tsv
stats@kw14764:~/liewy/sleuth$
```

Let's have a look at the file

`less sleuth_results.tsv`

```
stats@kw14764: ~/liewy/sleuth
target_id       mean_obs        b       se_b    test_stat       pval    qval    rss     sigma_sq        tech_var        var_obs sigma_sq_pmax   smooth_sigma_
AIPGENE10025    2.32931826874632        0.583636685366174       0.460771322057971       1.68469098888958        0.194302294762102       0.529915349351186
AIPGENE1009     2.29304565231056        -0.36249566443291       0.510302256793015       0.373972382653607       0.540846901487156       0.822413828935485
AIPGENE10186    1.62388101302263        0.148245125773393       0.417066674528103       0.126111964204669       0.722497848448123       0.898442920349998
AIPGENE10262    1.47795847927713        -0.199501194816 46      0.430779666128441       0.223104364696445       0.636684660617644       0.855590701801791
AIPGENE10274    2.22357530474316        -0.532005603440028      0.29181376910042        3.42922852644808        0.0640520154697088      0.307449674254602
AIPGENE10422    1.85234512024069        0.368020858222942       0.375242532180726       0.962671060943876       0.326514822178114       0.705833870704535
AIPGENE10493    3.09086325415455        0.014010649942281 7     0.201462057600904       0.297344617109515       0.585551916027302       0.833803530178546
AIPGENE10661    2.33653618192266        0.269405626013105       0.277999860782274       1.59811687992131        0.20617028056728        0.549787414846079
AIPGENE10690    2.53459932953649        -0.893859946865622      0.299379202972507       6.69346448693138        0.00967669751612123     0.145150462741818
AIPGENE1080     1.59271292433429        -0.20140297516559       0.452340570848115       0.198477141720597       0.655952871381373       0.855590701801791
AIPGENE10853    1.85181188146959        -0.269665294326628      0.39693610737836        0.461953464617546       0.496712431303393       0.784905187020377
AIPGENE1102     1.67150826495602        0.187422530770552       0.430532410628267       0.189293674393245       0.663505137401589       0.860763421493953
AIPGENE11113    1.10540794904127        -0.531216513586524      0.802220509291818       1.48893072284337        0.222382438359751       0.555956095899377
AIPGENE11173    1.46331758297285        -2.51816787553406       0.523975042822565       11.1398232778769        0.000844939407516352    0.133480149358161
AIPGENE11225    2.00212751761477        0.584853642758063       0.335205130178285       3.05310172052393        0.0805831088811954      0.340975158868779
AIPGENE11332    2.08141904293756        0.743845988037689       0.465834344999904       2.61996306608789        0.10552689699943        0.395316072011225
AIPGENE11334    2.07918464463003        -0.628547850198279      0.334176131020811       3.56991808178221        0.0588352136115894      0.294176068057947
AIPGENE11436    1.6320995988862 -0.24702488605977       0.416182896158001       0.351979342698728       0.552994805149371       0.829492207724057       0.097
AIPGENE11491    1.5111642022273 -0.777544761619821      0.423467814783015       3.44877010376841        0.0632988548674101      0.307449674254602       1.736
AIPGENE11684    2.18361078387933        0.990304706433666       0.638899563801684       2.4835509700523 0.115042257661046       0.412091669233596       3.920
AIPGENE11806    1.26303087558137        -2.12023294321846       0.545398413607741       9.04546183473395        0.00263347695710463     0.133480149358161
AIPGENE11829    1.72124256658173        -1.37592857561266       0.424933709754384       7.92959011590093        0.00486328125622953     0.133480149358161
AIPGENE11903    1.69957960457581        -0.114266845369561      0.379526475458451       0.090405762356093       0.763661945247372       0.935096259486578
AIPGENE12090    2.0099443295294 0.186110633631661       0.345514675653032       0.326430656384298       0.567768444430204       0.830145897339334       0.214
AIPGENE12161    2.28880900569868        0.186624781420407       0.261393741108567       0.824002481404381       0.364012527291955       0.728025054583909
AIPGENE12299    1.61617245893993        0.794781416104117       0.401482892634558       3.9188306464698 0.0477480815336748      0.291538056645428       1.117
AIPGENE12315    1.68122848894373        -1.04965285364819       0.520363752350885       3.8714775515489 0.0491132665266984      0.291538056645428       3.277
AIPGENE12475    2.46017769219232        0.219403996754023       0.252503411738678       1.33033845629213        0.248745014991831       0.591077263346926
```

"qval" (i.e. post-Benjamini-Hochberg multiple testing corrected "pval") is the column that tells me whether a gene is significantly expressed or not.

If qval < 0.05, the gene is significantly differentially expressed at 32 C.

If qval > 0.05, the gene is not differentially expressed.

Press **q** to get out of less.

How do we which genes are differentially expressed, from the command line?

`awk '{if ($7 < 0.05) print}' sleuth_results.tsv`

stats@kw14764: ~/liewy/sleuth

```
stats@kw14764:~/liewy/sleuth$ less sleuth_results.tsv
stats@kw14764:~/liewy/sleuth$ awk '{if ($7 < 0.05) print}' sleuth_results.tsv
stats@kw14764:~/liewy/sleuth$
```

... uh, no results? What.

Unfortunately, because we only used a tiny tiny fraction of the NGS data produced by the experiment, the pipeline failed to find any genes that were differentially expressed. The underlying reason is because very few genes had had reads mapping to it. Differential expression works best when you have low (but detectable) expression in the three replicates of one condition and high expression in the other three replicates.

Doesn't matter for now—we'll proceed by defining genes that have p < 0.5 (ha) as significantly differentially expressed.

`awk '{if ($7 < 0.5) print}' sleuth_results.tsv`

stats@kw14764: ~/liewy/sleuth

```
32739125027184  0.0442851389443665      0.10226268606107        0.146547825005436       0.0442851389443665      0.0245143730658433      0.
AIPGENE456      2.41224075438874        0.421575765187678       0.255086060181527       3.21113516839378        0.0731387243292444      0.
02726968688046  -0.0154595911161752     0.0960053448537844      0.0805453937376094      0       0.0244666053243663      0.0244666053243663
AIPGENE4584     2.50259955013825        0.517725934146626       0.388140911997186       1.86894806663765 0.171595607403434      0.49973392
72804   0.157762115661303       0.103433968284305       0.261196083945608       0.157762115661303       0.0213890147047545      0.15776211
AIPGENE5447     2.1501690791238 -0.601718152672718      0.305206722997879       3.97495102253396        0.0461817963569874      0.29153805
036497  0.0208484590200698      0.13953991618723        0.160388375207299       0.0208484590200698      0.0089069569613764      0.02084845
AIPGENE5564     2.33169407511492        0.690099092964016       0.322958095667327       4.23013843810488        0.0397122013483894      0.
4016672650703   0.14397880058709        0.124054544714317       0.268033345301407       0.14397880058709        0.0227628754297242      0.
AIPGENE5608     2.53849782793707        0.837833117170856       0.306086668902497       5.99353558901423        0.0143583953370624      0.
1508079162131   0.221941375534063       0.101074782790199       0.323016158324263       0.221941375534063       0.0194821490737221      0.
AIPGENE5781     2.32794979655436        0.320917239499378       0.267364467327 14       2.1567049831772 0.141948797250552       0.46772655
383015  -0.0575670618804988      0.106071400057102       0.048504338176603       0       0.0225688261351705      0.0225688261351705      1
AIPGENE5814     2.09094478680262        -0.749387484602621      0.420534732973368       3.16738893247403        0.075122455599481       0.
0346917294046   0.189082945972906       0.191610888615185       0.380693834588091       0.189082945972906       0.00397406872799283     0.
AIPGENE6256     2.18099773384915        -0.652244252094166      0.316238042554643       4.3443732115391 0.0371314323889722      0.27139995
622753  0.0365063092915021      0.149728909633049       0.186235218924551       0.0365063092915021      0.011973480594759       0.03650630
AIPGENE6863     2.73710753195917        -0.563177963133736      0.367100523580758       2.43744388810298        0.118469132300394       0.
843308937188    0.167571052583074       0.0892951261606853      0.256866178743759       0.167571052583074       0.0116718686430607      0.
AIPGENE6868     3.60406130561031        -0.250641304396172      0.187889294654435       1.86928926356072        0.171556503230262       0.
06045917478569  0.0291955563687984      0.0320136271269155      0.0612091834957138      0.0291955563687984      0.0182259762772455      0.
AIPGENE6957     2.38503600208099        0.529679317864872       0.317351197173179       2.83234195096487        0.092383548934368       0.
2511096374419   0.0820916835253727      0.122930509223466       0.205022192748838       0.0820916835253727      0.0244739902564024      0.
AIPGENE7996     2.50782049407007        -0.584832987336476      0.24770738952823        5.65211186223119        0.0174343634393028      0.
93625627518044  0.0287478208225261      0.0899773046810827      0.118725125503608       0.0287478208225261      0.0211225316221963      0.
AIPGENE8197     1.87552568858925        0.80936943027865        0.376972764539814       4.46922359340406        0.0345106700032776      0.
7055243731145   0.120950821281762       0.213159666180528       0.33411048746229        0.120950821281762       6.44755925545454e-05    0.
AIPGENE8325     2.45267628092863        -0.541670147512279      0.280456091000035       4.002597363512  0.0454302036922759      0.29153805
613849  -0.00357093370341625    0.116186413626186       0.11261547992277        0       0.0235493931542679      0.0235493931542679      1
AIPGENE865      4.16948114049548        -0.195849953007383      0.149474866527594       1.85826958791887        0.172824647530097       0.
915922204801 44 0.0217471765577779      0.0165712675382509      0.0383184440960288      0.0217471765577779      0.023657698730703       0.
AIPGENE8668     2.19259819484259        1.31019098796464        0.365121633737695       8.29886574292747        0.00396698469091009     0.
7478348195534   0.476339580946175       0.198617115444893       0.674956696391069       0.476339580946175       0.0129207798635923      0.
AIPGENE8796     1.55683969986854        -2.18419491510916       0.722718753388894       6.79443614703502        0.00914423993044749     0.
2899955197831   1.88285187419725        0.175147229759364       2.05799910395661        1.88285187419725        0.00081731502423708 1   1.
AIPGENE9147     3.25812428888543        0.544303547049346       0.183881583597743       7.35867179282024        0.00667400746790806     0.
72641505526631  0.0733657559852889      0.0411625451200373      0.114528301105326       0.0733657559852889      0.0149173764838321      0.
AIPGENE9256     2.08368870012543        0.98840548399731        0.333781342283795       7.122581466492  0.00761190260382122     0.14052743
01333   0.227241488957096       0.167034899645571       0.394276388602667       0.227241488957096       0.00358227569170963     0.22724148
stats@kw14764:~/liewy/sleuth$
```

Hallelujah, there IS something produced.

To find out how many genes that are "differentially expressed",

`awk '{if ($7 < 0.5) print}' sleuth_results.tsv | wc -l`

```
stats@kw14764: ~/liewy/sleuth
stats@kw14764:~/liewy/sleuth$ awk '{if ($7 < 0.5) print}' sleuth_results.tsv | wc -l
83
stats@kw14764:~/liewy/sleuth$
```

(I hope you remember what "|" was—I covered it in my morning session. Recall also the "%<%" thingy that Nate presented in day 1. This is piping—you produce some text output in the first command, which is then piped into a line-counter, "word count dash line".)

We have 83 genes differentially expressed! Yay!

To see which genes they are, we change the command after the pipe.

```
awk '{if ($7 < 0.5) print}' sleuth_results.tsv | cut -f 1
```

```
stats@kw14764: ~/liewy/sleuth
AIPGENE24701
AIPGENE25102
AIPGENE25162
AIPGENE25192
AIPGENE25893
AIPGENE26044
AIPGENE27150
AIPGENE27660
AIPGENE28021
AIPGENE28295
AIPGENE28423
AIPGENE28530
AIPGENE28637
AIPGENE2901
AIPGENE3056
AIPGENE3352
AIPGENE3439
AIPGENE345
AIPGENE3988
AIPGENE4474
AIPGENE456
AIPGENE4584
AIPGENE5447
AIPGENE5564
AIPGENE5608
AIPGENE5781
AIPGENE5814
AIPGENE6256
AIPGENE6863
AIPGENE6868
AIPGENE6957
AIPGENE7996
AIPGENE8197
AIPGENE8325
AIPGENE865
AIPGENE8668
AIPGENE8796
AIPGENE9147
AIPGENE9256
stats@kw14764:~/liewy/sleuth$
```

The cut command "cuts" out the first (-f 1) column of the table. Let's save these bunch of genes into a file!

```
awk '{if ($7 < 0.5) print}' sleuth_results.tsv | cut -f 1 >
diff_expr_genes.txt
```

And let's verify that the file exists.

```
ls
```

```
stats@kw14764:~/liewy/sleuth$ awk '{if ($7 < 0.5) print}' sleuth_results.tsv | cut -f 1 > diff_expr_genes.txt
stats@kw14764:~/liewy/sleuth$ ls
diff_expr_genes.txt  expt_setup.tsv  normalised_abundances.tsv  sleuth_analysis.R  sleuth_results.tsv
stats@kw14764:~/liewy/sleuth$
```

Let's get back to the previous folder

cd ..

Done!

7. Finale: Running a GO term analysis to find enriched GO terms

Enter the topgo folder, and look around.

cd topgo

ls

```
stats@kw14764: ~/liewy/topgo
stats@kw14764:~/liewy$ cd topgo/
stats@kw14764:~/liewy/topgo$ ls
aip_go_annots.all.tsv  aip_topgo_usage.R  topGO_output/
stats@kw14764:~/liewy/topgo$
```

Again, I have made life easier for you. I have modified the R script needed to do this section as "aip_topgo_usage.R".

So... run it :)

Rscript aip_topgo_usage.R

This process takes a while, so toilet break #2!

```
        Level 14:      1 nodes to be scored    (0 eliminated genes)
        Level 13:      3 nodes to be scored    (3 eliminated genes)
        Level 12:      2 nodes to be scored    (3 eliminated genes)
        Level 11:      4 nodes to be scored    (36 eliminated genes)
        Level 10:      7 nodes to be scored    (45 eliminated genes)
        Level 9:       16 nodes to be scored   (158 eliminated genes)
        Level 8:       21 nodes to be scored   (574 eliminated genes)
        Level 7:       28 nodes to be scored   (3226 eliminated genes)
        Level 6:       38 nodes to be scored   (3620 eliminated genes)
        Level 5:       43 nodes to be scored   (6382 eliminated genes)
        Level 4:       44 nodes to be scored   (8970 eliminated genes)
        Level 3:       28 nodes to be scored   (14269 eliminated genes)
        Level 2:       8 nodes to be scored    (16162 eliminated genes)
        Level 1:       1 nodes to be scored    (18670 eliminated genes)
stats@kw14764:~/liewy/topgo$
```

You should see this as the script ends. The script produces a few files in the folder topGO_output.

cd topGO_output

ls

```
stats@kw14764: ~/liewy/topgo
stats@kw14764:~/liewy/topgo$ cd topGO_output/
stats@kw14764:~/liewy/topgo/topGO_output$ ls
bp_diff_expr_genes.txt  cc_diff_expr_genes.txt  mf_diff_expr_genes.txt  summarise_topGO_output.sh*
stats@kw14764:~/liewy/topgo/topGO_output$
```

The circled files were produced from the R script. Feel free to look at them using **less**. If you do, remember to press **q** to quit.

To process these files, run the shell script in the same folder.

EITHER

```
source summarise_topGO_output.sh
```

OR (the lazier way)

```
./summarise_topGO_output.sh
```

The script produces an additional file.

```
ls
```

```
stats@kw14764: ~/liewy/topgo
stats@kw14764:~/liewy/topgo$ cd topGO_output/
stats@kw14764:~/liewy/topgo/topGO_output$ ls
bp_diff_expr_genes.txt  cc_diff_expr_genes.txt  mf_diff_expr_genes.txt  summarise_topGO_output.sh*
stats@kw14764:~/liewy/topgo/topGO_output$ ./summarise_topGO_output.sh
stats@kw14764:~/liewy/topgo/topGO_output$ ls
bp_diff_expr_genes.txt  cc_diff_expr_genes.txt  mf_diff_expr_genes.txt  summarise_topGO_output.sh*  summary_diff_expr_genes.txt
stats@kw14764:~/liewy/topgo/topGO_output$
```

Let's check out the summary file.

```
less summary_diff_expr_genes.txt
```

```
stats@kw14764: ~/liewy/topgo
-- bp_diff_expr_genes.txt --
        GO.ID   Term    Annotated       Significant     Expected        P_value
1       GO:0006412      translation     519     20      1.51    2.3e-18
2       GO:0006414      translational elongation        49      5       0.14    1.6e-06
4       GO:0071353      cellular response to interleukin-4      9       2       0.03    0.00030
5       GO:0000022      mitotic spindle elongation      15      2       0.04    0.00086
6       GO:0035914      skeletal muscle cell differentiation    19      2       0.06    0.00139
7       GO:0006614      SRP-dependent cotranslational protein targeting to membrane     21      2       0.06    0.00170
8       GO:0006987      activation of signaling protein activity involved in unfolded protein response  24      2       0.07
17      GO:0001892      embryonic placenta development  39      2       0.11    0.00579
26      GO:0006457      protein folding 140     3       0.41    0.00796
27      GO:0007476      imaginal disc-derived wing morphogenesis        48      2       0.14    0.00866
31      GO:0021837      motogenic signaling involved in postnatal olfactory bulb interneuron migration  5       1       0.01
32      GO:0072177      mesonephric duct development    5       1       0.01    0.01451
33      GO:0006869      lipid transport 185     3       0.54    0.01684
34      GO:0030835      negative regulation of actin filament depolymerization  46      2       0.13    0.01712
35      GO:0007182      common-partner SMAD protein phosphorylation     6       1       0.02    0.01738
```

Thus, we can conclude that heat-stressed genes tend to be translation-related / translation-elongation-related. It could perhaps be that under heat stress, there is an increased expression of chaperone genes, which in turn, aid in the correct expression and folding of proteins.

8. Conclusion

This is basically how I'd carry out a transcriptomics analysis.

The analysis results in a list of GO terms that describe what sorts of genes tend to be differentially expressed. Use this list to guide you in designing future experiments. Some people use this list and basically write it up as a paper, a practice which I generally dislike, because of the absence of experimental proof.

Please DO NOT trust the results of today's practical—remember, we used $p < 0.5$ to decide whether a gene was differentially expressed, just to squeeze out something for the later steps. If you publish results with $p < 0.5$, you deserve all the scorn you get from your reviewers! :p

I hope you enjoyed the ride! If you want more info about the GO term analysis, check out

https://github.com/lyijin/topGO_pipeline

I have not written up the kallisto/sleuth bits, but if I ever do, it'll be on my github.