RNA seq:
differential expression analysis

For INF-BIO 4121/9121
Fall semester 2015

Monica Hongrø Solbakken
m.h.solbakken@ibv.uio.no

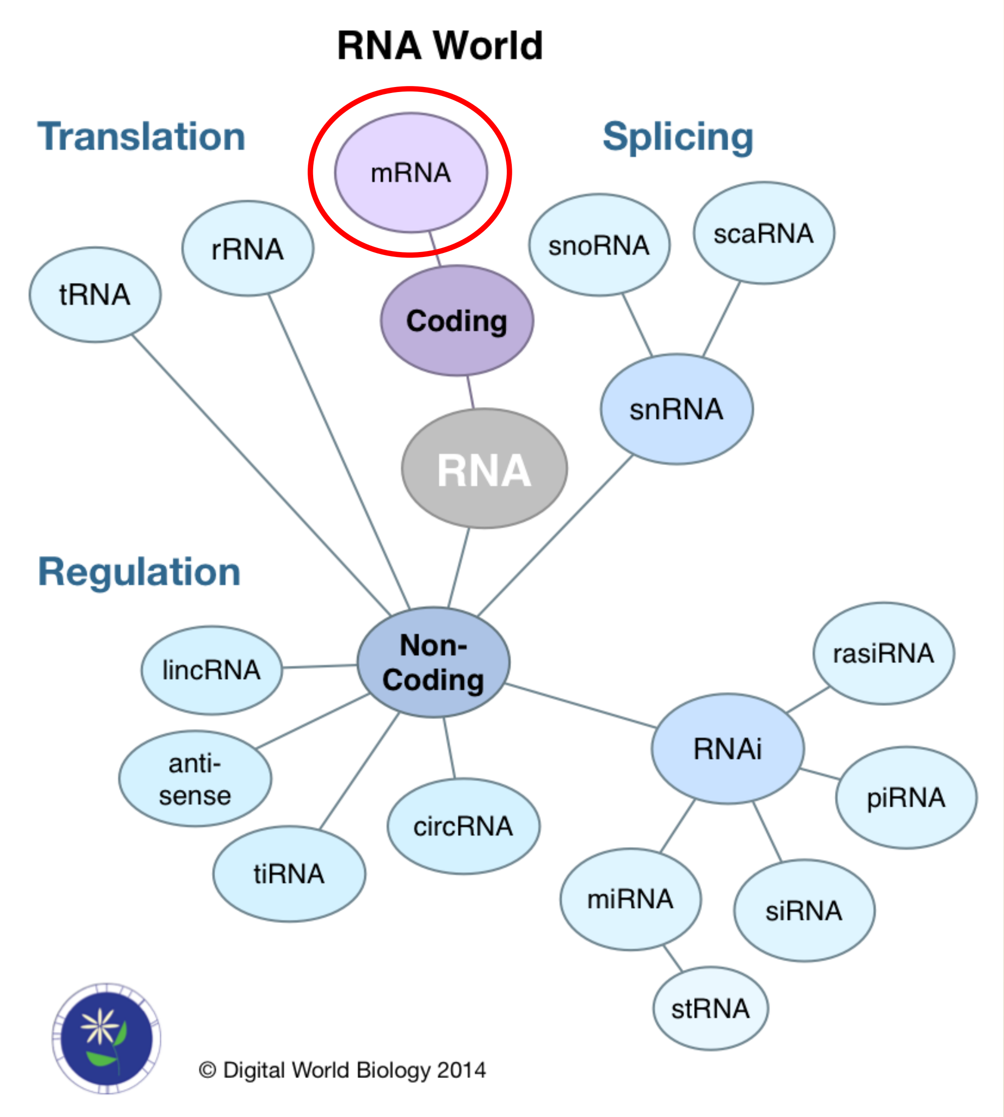
UiO : Centre for Ecological and Evolutionary Synthesis
University of Oslo

Outline II

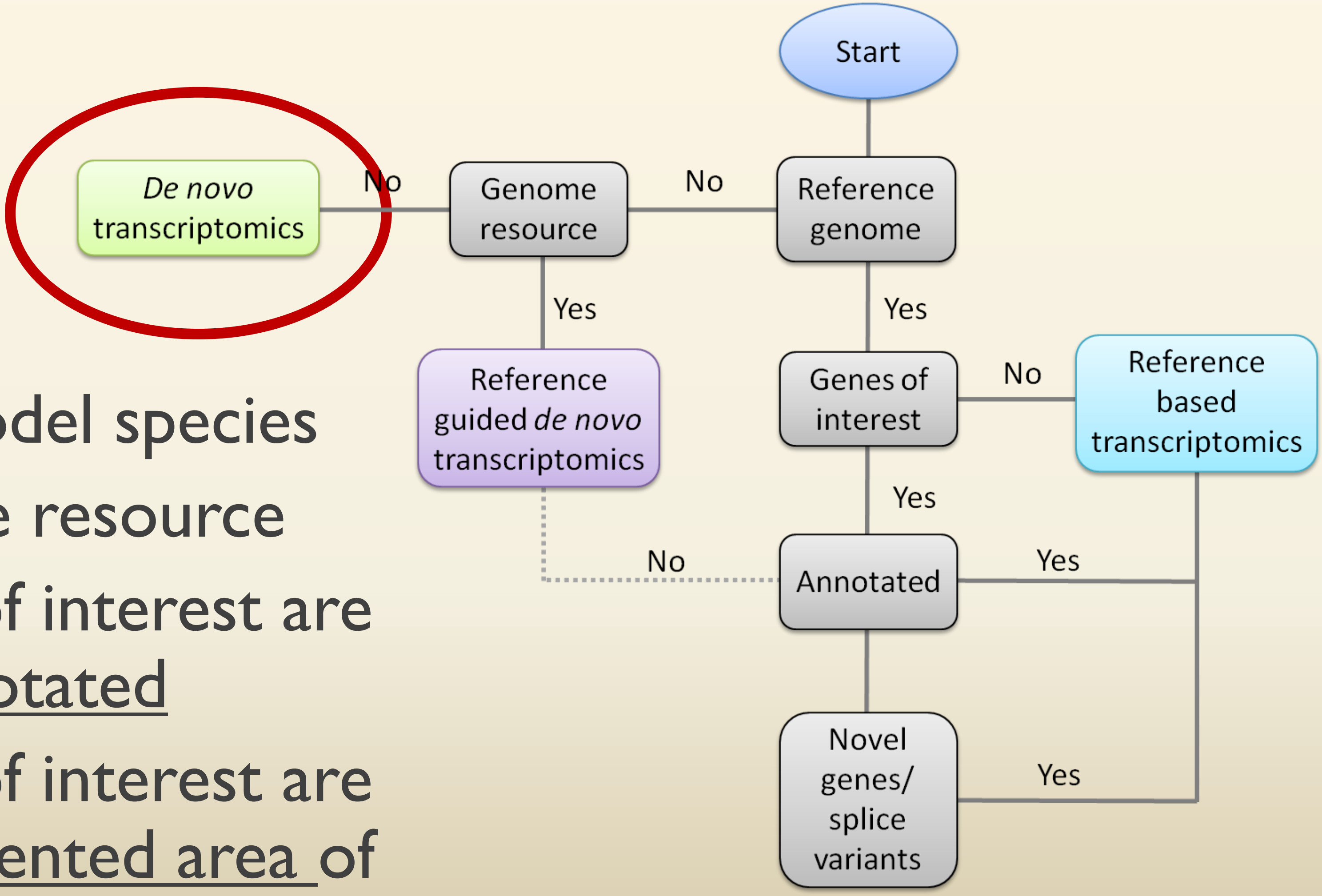
- The RNAseq module day II
  - Recap assembly
  - Assembly annotation
  - Differential expression
- Today we will cover
  - The Trinotate annotation pipeline
  - Mapping of individual samples
  - Evaluation of mapping
  - Extraction of raw expression counts
  - Start on differential expression analysis

Focusing on mRNA

Our transcriptome is mRNA selected thus we focus on protein coding genes



We chose *de novo*


Start
Reference genome
No
Genome resource
No
De novo transcriptomics
Yes
Reference guided *de novo* transcriptomics
Yes
Genes of interest
No
Reference based transcriptomics
Yes
Annotated
No
Yes
Novel genes/ splice variants
Yes

- Non-model species
- Genome resource
- Genes of interest are <u>not annotated</u>
- Genes of interest are <u>in fragmented area </u>of genome

De novo made with Trinity

- Trinity is the best single parameter *de novo* RNA assembly pipeline available
- Good on splice variants, full length transcripts and resolution of lowly expressed transcripts
- Contains tools to help with visualizations

You compared stats for all Trinity.fastas

	Complete	GG	Alternative	Mini
Total trinity 'genes'	320 520	342 099	380 658	98 930
Total trinity transcripts	468 626	454 484	569 062	117 062
Percent GC	47.31	47.64	47.41	49.38
Stats based on ALL transcript contigs				
Contig N10	3 657	5 607	4 648	3 545
Contig N20	2 645	3 962	3 330	2 628
Contig N30	2 042	2 986	2 524	2 059
Contig N40	1 597	2 276	1 930	1 625
Contig N50	1 235	1 716	1 463	1 268
Median contig length	459	505	472	459
Average contig	784.28	972.96	873.39	799.82
Total assembled bases	367 534 825	442 195 867	497 015 429	93 627 954

You compared stats for all Trinity.fastas

- Today we use `Trinity_complete.fasta`
  - Has abundance estimation (2 days computing)
  - Has full length estimation (7 days computing)
  - Has annotation (1-2 weeks computing)

Further transcriptome evaluation I

- Full length estimation is a BLAST based approach
- Atlantic cod has ~22 000 genes
- The drawback of *de novo* on a more complex eukaryote

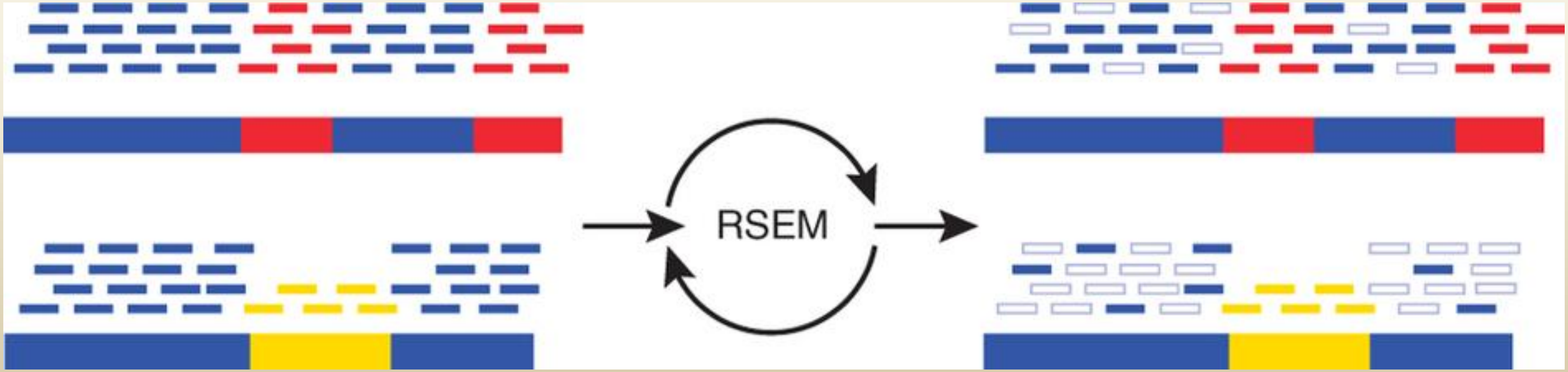
hit_pct cov bin	count_in bin	>bin below
100	5027	5027
90	2008	7035
80	1841	8876
70	1915	10791
60	2189	12980
50	2491	15471
40	2793	18264
30	3213	21477
20	2961	24438
10	906	25344

Further transcriptome evaluation II

- Abundance estimation maps all samples to the transcriptome for a simple expression estimation of all isoforms
- «Detects» artefacts
- May be used for filtering
- Mapback results also indicate read quality

Abundance estimation

- We use the RSEM mapper
- RSEM uses a likelihood based alogrithm to place multimapping reads
- Same method for extracting differential expression counts



doi:10.1038/nprot.2013.084

RSEM results

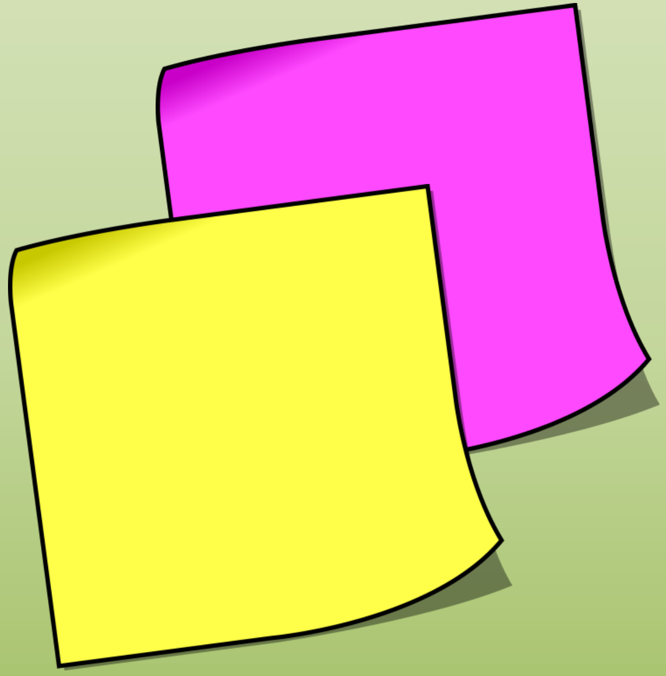
- Comparison of the genome guided (GG) Trinity and *de novo* (DN) Trinity in the assembly folder
- FPKM -2/-3 reflects the predicted 22 000 genes in Atlantic cod in `Trinity_complete.fasta`

GG		DN
num features	neg_min fpkm	num features
-83	1	-65033
-69	2	-18941
	3	-16920
-66	4	-15806
-65	5	-11258
	6	-11201
…	…	…
73371	-6	16072
86545	-5	18267
104769	-4	21055
135418	-3	24869
199789	-2	30779
315811	-1	46015
342099	0	320520

Working with RSEM files

The sticky notes!

- Put up YELLOW if command is running nicely
- Put up PINK if error or other issues



Support

- In /data/RNAseq2 there is a file called READ_ME
- This file contains commands so you can copy/paste to save time

Kill any old running jobs on cod 1,3,4

- The Trinity we started was very suboptimal (Mini)
- Retrieve screen and kill running command(s)

```
screen -ls
screen -rd <number>
ctrl c
exit
```

- Alternatively

```
top
kill:pid
```

Home area clean up

- Delete any non-usable files in your home area

```
    rm file
    rm -r directory
```

Today you will need these files:

```
Trinity_complete.fasta
1 trimmed sample set
RSEM.genes.results
RSEM.isoforms.results
Trinity_complete.fasta index (bowtie)
4 different .matrix files
```

Mapping back I

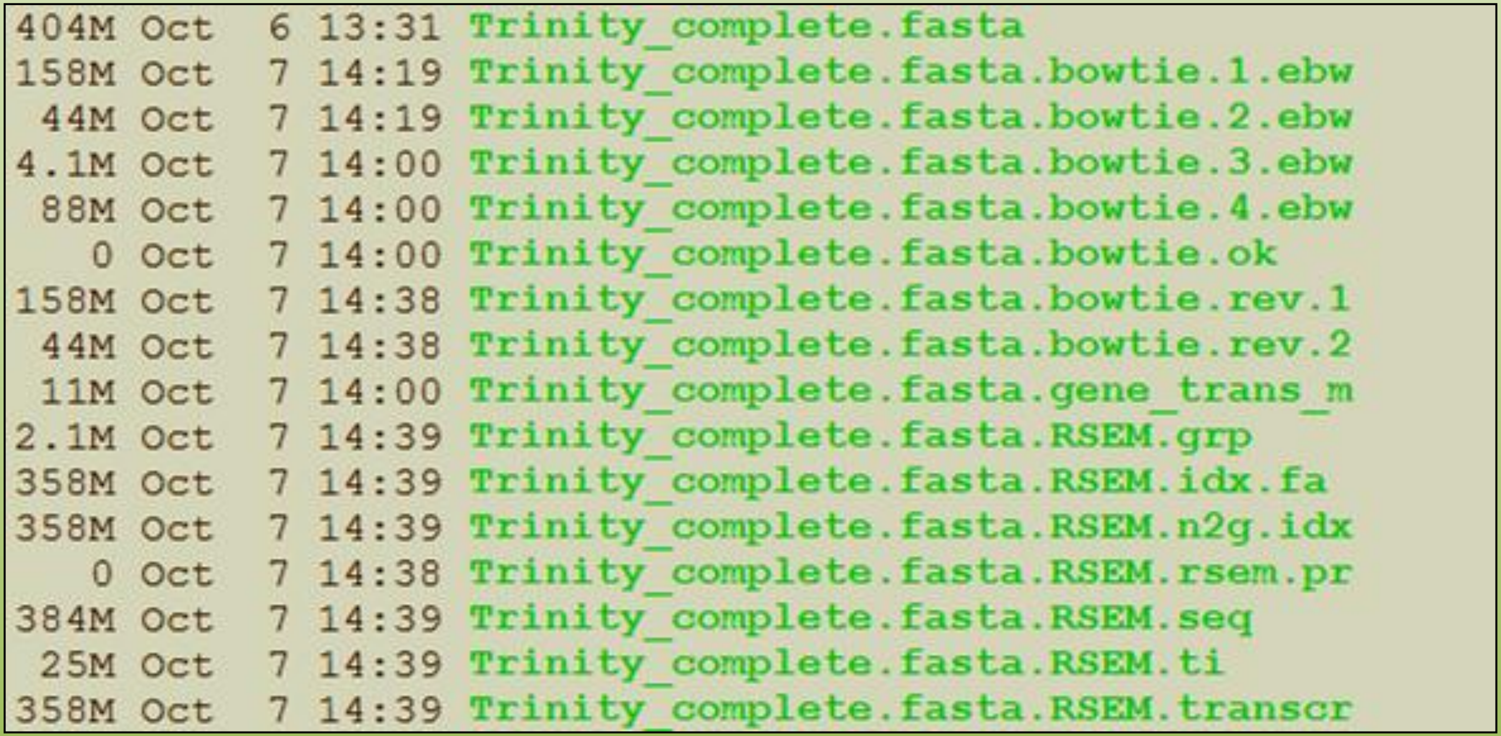
- Set up your environment like so:

```
module load samtools/1.1
module load trinityrnaseq
module load perlmodules/5.10_2
module load gcc/5.2.0
ulimit -s unlimited
```

Mapping back II

Make sure you have Trinity_complete.fasta AND its index in your home directory

If not, copy it from /data/RNAseq2/assembly



```
404M Oct  6 13:31 Trinity_complete.fasta
158M Oct  7 14:19 Trinity_complete.fasta.bowtie.1.ebw
 44M Oct  7 14:19 Trinity_complete.fasta.bowtie.2.ebw
4.1M Oct  7 14:00 Trinity_complete.fasta.bowtie.3.ebw
 88M Oct  7 14:00 Trinity_complete.fasta.bowtie.4.ebw
   0 Oct  7 14:00 Trinity_complete.fasta.bowtie.ok
158M Oct  7 14:38 Trinity_complete.fasta.bowtie.rev.1
 44M Oct  7 14:38 Trinity_complete.fasta.bowtie.rev.2
 11M Oct  7 14:00 Trinity_complete.fasta.gene_trans_m
2.1M Oct  7 14:39 Trinity_complete.fasta.RSEM.grp
358M Oct  7 14:39 Trinity_complete.fasta.RSEM.idx.fa
358M Oct  7 14:39 Trinity_complete.fasta.RSEM.n2g.idx
   0 Oct  7 14:38 Trinity_complete.fasta.RSEM.rsem.pr
384M Oct  7 14:39 Trinity_complete.fasta.RSEM.seq
 25M Oct  7 14:39 Trinity_complete.fasta.RSEM.ti
358M Oct  7 14:39 Trinity_complete.fasta.RSEM.transcr
```

Mapping back III

- The script you will be using is located here:

```
/cluster/software/VERSIONS/\
trinityrnaseq/trinityrnaseq-2.0.6/util/\
align_and_estimate_abundance.pl

#parameters
--transcripts <path to Trinity_complete.fasta>
--seqType fq
--left <path to R1 trimmed>
--right <path to R2 trimmed>
--est_method RSEM
--thread_count 2
--output_dir <name>
--aln_method bowtie
--trinity_mode
1>rsem_trimmed_default.out
2>rsem_trimmed_default.err
```

You can use either .fq or .fq.gz

Cod 4 can be used.

Give output directory the same name as the sample

Screen! Should take ~1.5 hrs

Look at some RSEM.genes.results I

- Use: `21dayK_F4/RSEM.genes.results` in `/data/RNAseq2/trimmed_data/mapping`
- Copy this file to ~

Gene id	Transcript id(s)	length	Effective length	Expected count	TPM	FPKM
c100000_g1	c100000_g1_i1	668	476	0	0	0
c100001_g1	c100001_g1_i1	201	34,59	0	0	0
c100001_g2	c100001_g2_i1,c100001_g2_i2	283	99,61	0	0	0
c100002_g1	c100002_g1_i1,c100002_g1_i2, c100002_g1_i3,c100002_g1_i4	441,43	250,63	12	3,53	3,58
c100003_g1	c100003_g1_i1	1206	1013,96	0	0	0

Look at some RSEM.genes.results II

- Length: transcript length without poly A tail
- Effective length: transcript positions that can generate a valid fragment
- Expected count: sum of the posterior probability of each read comes from this transcript over all reads
- TPM: Transcripts Per Million (relative measure of transcript abundance)
- FPKM: Fragments Per Kilobase of transcript per Million mapped reads (another relative measure of transcript abundance)

Gene id	Transcript id(s)	length	Effective length	Expected count	TPM	FPKM
c100002_g1	c100002_g1_i1,c100002_g1_i2, c100002_g1_i3,c100002_g1_i4	441,43	250,63	12	3,53	3,58

Look at some RSEM.genes.results III

- Use: `21dayK_F4/RSEM.genes.results`

- Can you find the gene with FPKM of 274.36?
  - Hint: grep
- Which gene has the highest FPKM value and what is this value?
  - Hint: awk / UNIX

Look at some RSEM.genes.results IV

```
awk '{print $7}' RSEM.genes.results \
 | sort -n | tail
```

```
>65354.39
```

```
grep '65354.39' RSEM.genes.results
```

```
>c131771_g6…
```

Filtering Trinity.fasta by RSEM

Try filtering Trinity_complete.fasta I

- Make sure you have RSEM.isoforms.results
  - (derived from Trinity_complete.fasta)
- The script you will be using is located here:

```
/cluster/software/VERSIONS/\
trinityrnaseq/trinityrnaseq-2.0.6/util/\
filter_fasta_by_rsem_values.pl

#parameters
--rsem_output RSEM.isoforms.results
--fasta Trinity_complete.fasta
--output Trinity_complete_filtered.fasta
--fpkm_cutoff 1
```

Try filtering Trinity_complete.fasta II

- Rerun trinity stats when you are finished

```
/cluster/software/VERSIONS/\
trinityrnaseq/trinityrnaseq-2.0.6/util/\
TrinityStats.pl \
Trinity_complete_filtered.fasta > \
Trinity_complete_filtered.fasta.stats.txt
```

Compare complete and filtered

	Complete	**Filtered**
Total trinity 'genes'	320 520	
Total trinity transcripts	468 626	
Percent GC	47.31	
Stats based on ALL transcript contigs		
Contig N10	3 657,00	
Contig N20	2 645	
Contig N30	2 042	
Contig N40	1 597	
Contig N50	1 235	
Median contig length	459	
Average contig	784.28	
Total assembled bases	367 534 825	

Compare complete and filtered

	Complete	Filtered
Total trinity 'genes'	320 520	33 488
Total trinity transcripts	468 626	46 810
Percent GC	47.31	48.72
Stats based on ALL transcript contigs		
Contig N10	3 657	4 497
Contig N20	2 645	3 435
Contig N30	2 042	2 799
Contig N40	1 597	2 330
Contig N50	1 235	1 953
Median contig length	459	1002
Average contig	784.28	1341.31
Total assembled bases	367 534 825	627 86 646

Filtering Trinity.fasta

- Why should you be careful when filtering Trinity.fasta?

Filtering Trinity.fasta

- Why should you be careful when filtering Trinity.fasta?

Risk of loosing rare transcripts and/or lowly expressed transcripts

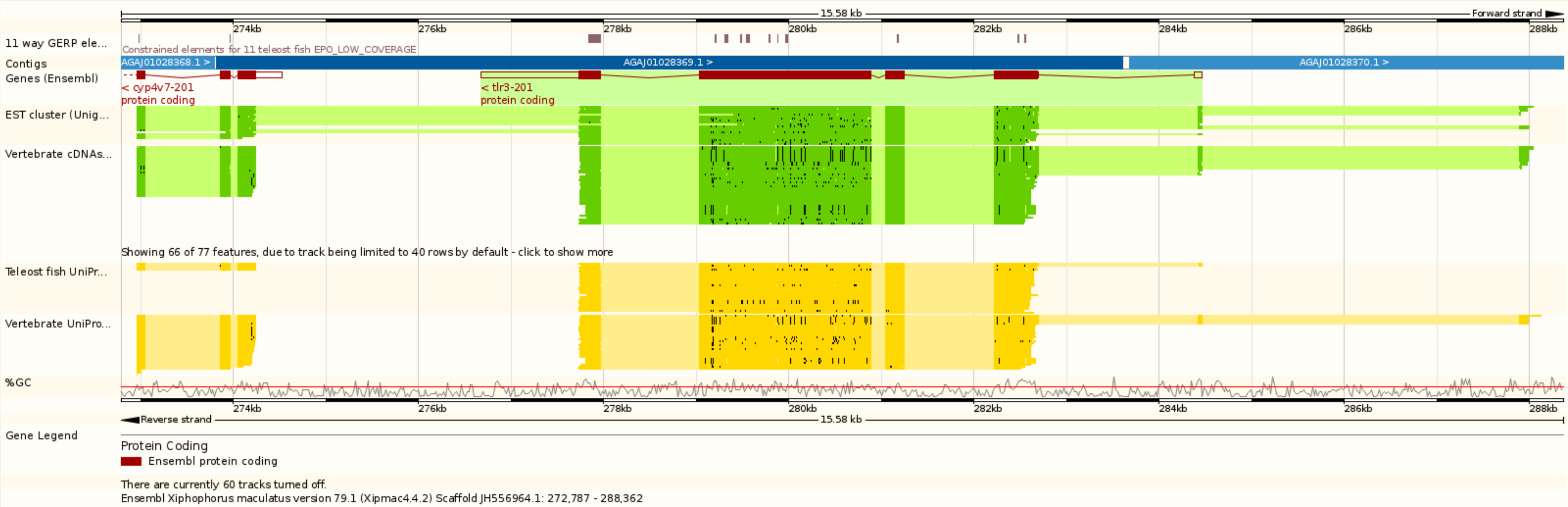
Short lecture
– Assembly annotation pipelines

Annotation

- Annotation = metadata to your assembly
- Prediction of protein coding regions, non-coding RNAs, ribosomal RNAs…
- Often based on sequence homology (BLAST) and reading frame investigation (finding likely protein coding regions)

Why annotate?

- Obtain a general overview over your assembly
- Use annotation as a quality measure



Showing 66 of 77 features, due to track being limited to 40 rows by default - click to show more

Protein Coding

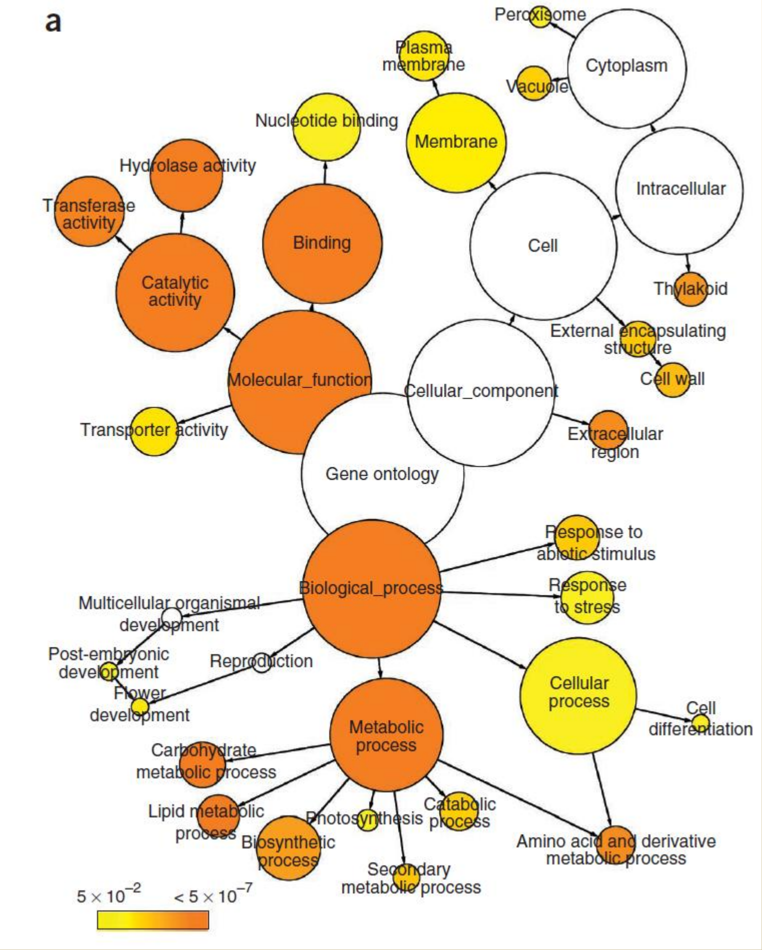
🟥 Ensembl protein coding

There are currently 60 tracks turned off.

Ensembl Xiphophorus maculatus version 79.1 (Xipmac4.4.2) Scaffold JH556964.1: 272,787 - 288,362

What to annotate

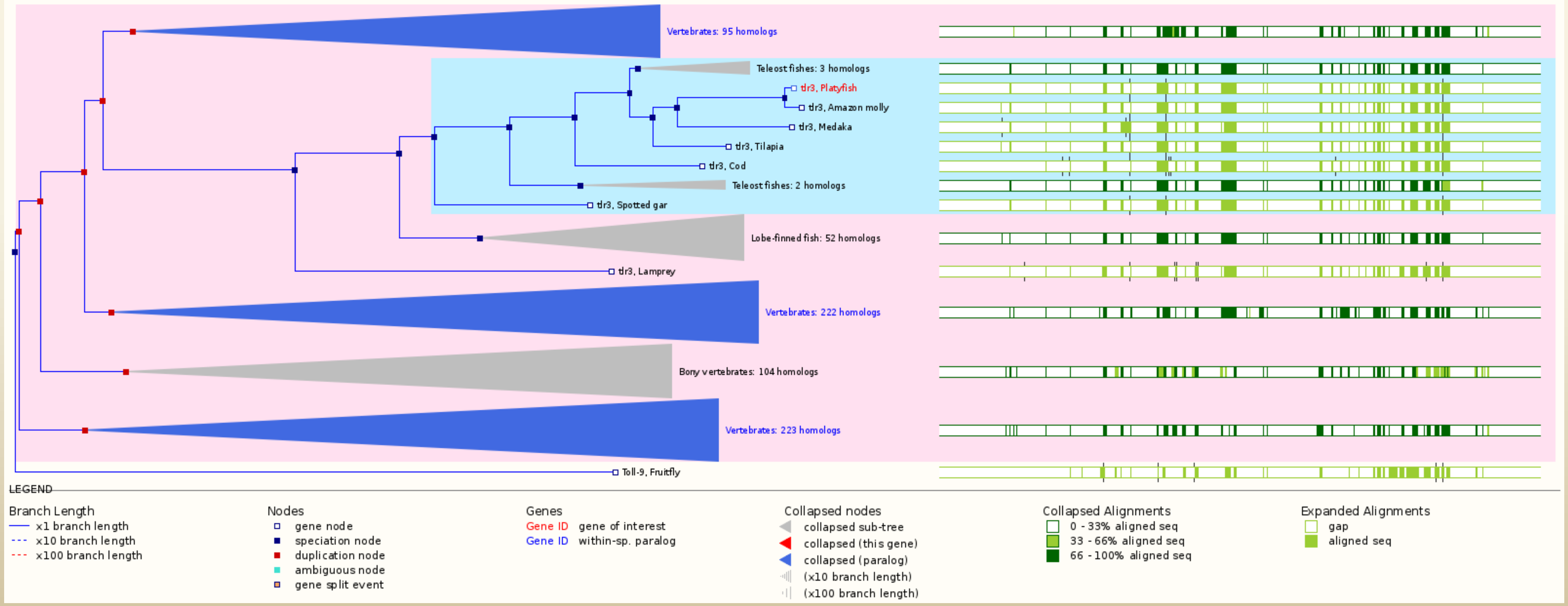
- Most common: protein coding genes
- Others:
  - – Gene ontology (group by function)
  - – Non-coding RNA
  - – Ribosomal RNA
  - – Small RNAs
  - – Repeat elements



The genome of woodland strawberry (Fragaria vesca) – Nature genetics

Usage of an RNA assembly annotation

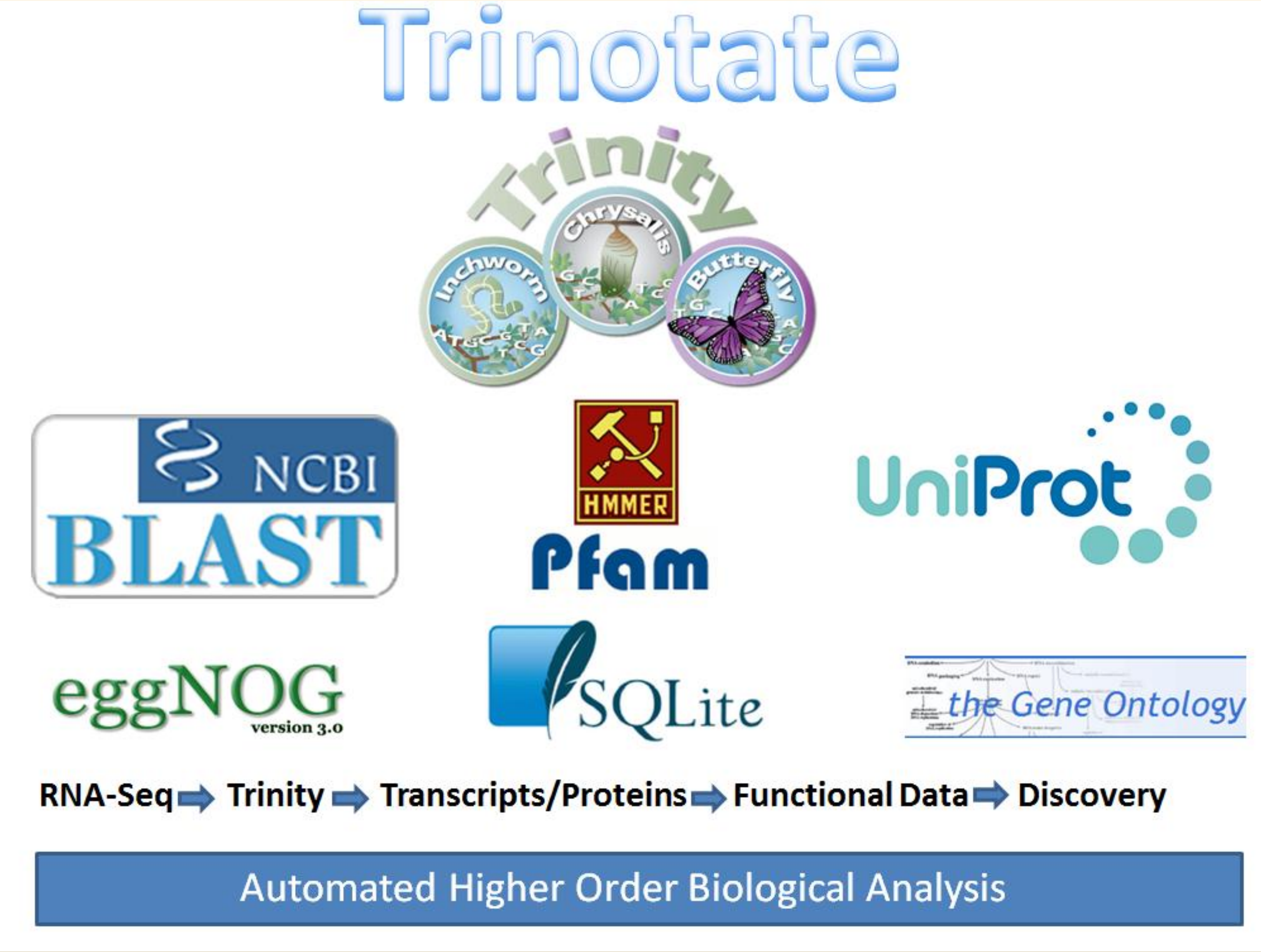
- Comparative analyses
- Append it to a differential expression analysis
    - – Tissue expression profiling



Annotation pipelines

- Basic BLASTX towards a (curated) protein database
- Basic BLASTN towards a (curated) nucleotide database
- Mapping transcripts to a reference genome
- Blast2GO for gene ontology annotations only
- Extensive pipelines utilizing BLAST, protein stucture databases, signal sequences etc:
  - Pendant
  - Annocript
  - Trinotate

Trinotate



RNA-Seq → Trinity → Transcripts/Proteins → Functional Data → Discovery

Automated Higher Order Biological Analysis

Trinotate

- One of the most comprehensive annotation pipelines
- Combines BLAST, protein domain, protein structure, signal peptide and transmembrane domain searches
- Makes a SQLite database of the combined annotations
- Also available as a web version!
- A 1-2 week job using moderate resources on Abel

BLAST

- Homology search in several steps
  - Trinity transcripts towards SwissProt
    - Only top hit reported
  - Longest ORF reported from each Trinity transcript towards SwissProt
    - Only top hit reported
  - Optional: redo the same searches as above but towards the extensive Uniref90 database

Swissprot - manually annotated and reviewed section of the UniProt Knowledgebase (UniProtKB)
Uniref90 - The UniProt Reference Clusters: combines identical sequences and sub-fragments with 11 or more residues from any organism into a single UniRef entry

HMMER

- Searches for protein domains
- Utilizes a profile hidden Markov model instead for BLAST
- Great for detecting distant homologs
- Uses the PFAM database describing protein families in multiple sequence alignments and protein structures

Signal peptides

- SignalP4 predicts the presence and location of signal peptide cleavage sites
- Based on articifical neural networks
- Focus: only N-terminal -> ER secretory pathway signals

Transmembrane regions and RNA families

- Tmhmm: searches for transmembrane helices in your data
  - Hidden Markov model based approach
- RNAmmer: predicts 5s/8s and 23s/28s ribosomal RNA
  - Hidden Markov model based approach

The output of Trinotate

- Trinotate makes a searchable and filterable database

#gene_id	sprot_Top_BLASTX_hit	Pfam	gene_ontology_pfam
c10_g1	.	.	.
c81329_g1	EFTU_FRAP2^^Q:5586-4405,H:1-394^98.98%ID^E:0^.^.	PF00009.22^GTP_EFTU^Elongation factor Tu GTP binding domain^10-201^E:1.8e-61`PF01926.18^MMR_HSR1^50S ribosome-binding GTPase^15-134^E:6.2e-05`PF03144.20^GTP_EFTU_D2^Elongation factor Tu domain 2^225-294^E:1.3e-18`PF03143.12^GTP_EFTU_D3^Elongation factor Tu C-terminal domain^298-392^E:1.3e-34	GO:0003924^molecular_function^GTPase activity`GO:0005525^molecular_function^GTP binding
c81329_g2	DMRT2_HUMAN^^Q:14-343,H:110-219^80%ID^E:9e-57^.^.	PF00751.13^DM^DM DNA binding domain^14-60^E:5.6e-23	GO:0043565^molecular_function^sequence-specific DNA binding`GO:0006355^biological_process^regulation of transcription, DNA-templated
c81329_g3	RPOB_FRAP2^^Q:2-472,H:232-388^98.09%ID^E:4e-93^.^.	.	.

The Trinotate annotation report

- Can be a huge file!
  - Sometimes Excel can handle it….
  - Use UNIX or R to handle it, extract data, sort etc.

Lets check the annotation

- Make sure you have the Trinotate_report.xls
- Find a gene in the report – maybe your favorite gene?
- Take note of the Trinity isoform ID
- Extract that gene's sequence using the tool fastagrep in `/data/bin`

```
/data/bin/fastagrep -p <geneid> \
Trinity_complete.fasta > results.txt
```

- Go to: http://blast.ncbi.nlm.nih.gov and perform a blastx towards the nr database
- Does it match?

Can you trust the annotation?

- Discussion…

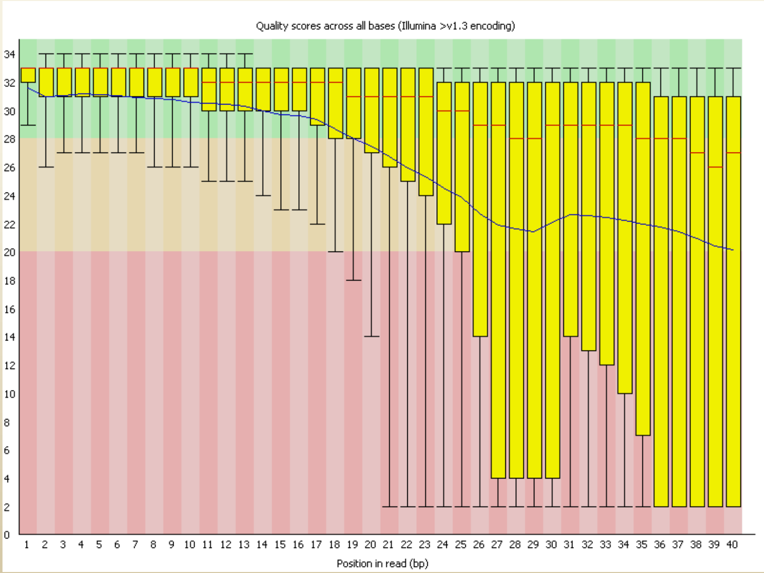
Can you trust the annotation?

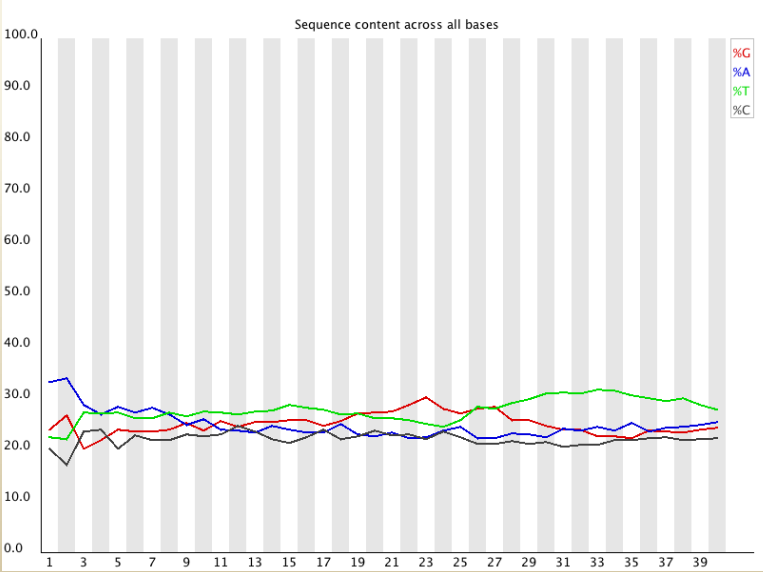
- BLAST – usually correct gene family, might miss the correct family member
- Sequence trait searches – traits may be different in non-model(non-mammalian) species
- You never know if it’s truly correct unless functionally tested *in vivo* (*in vitro*)

Differential expression

Recap

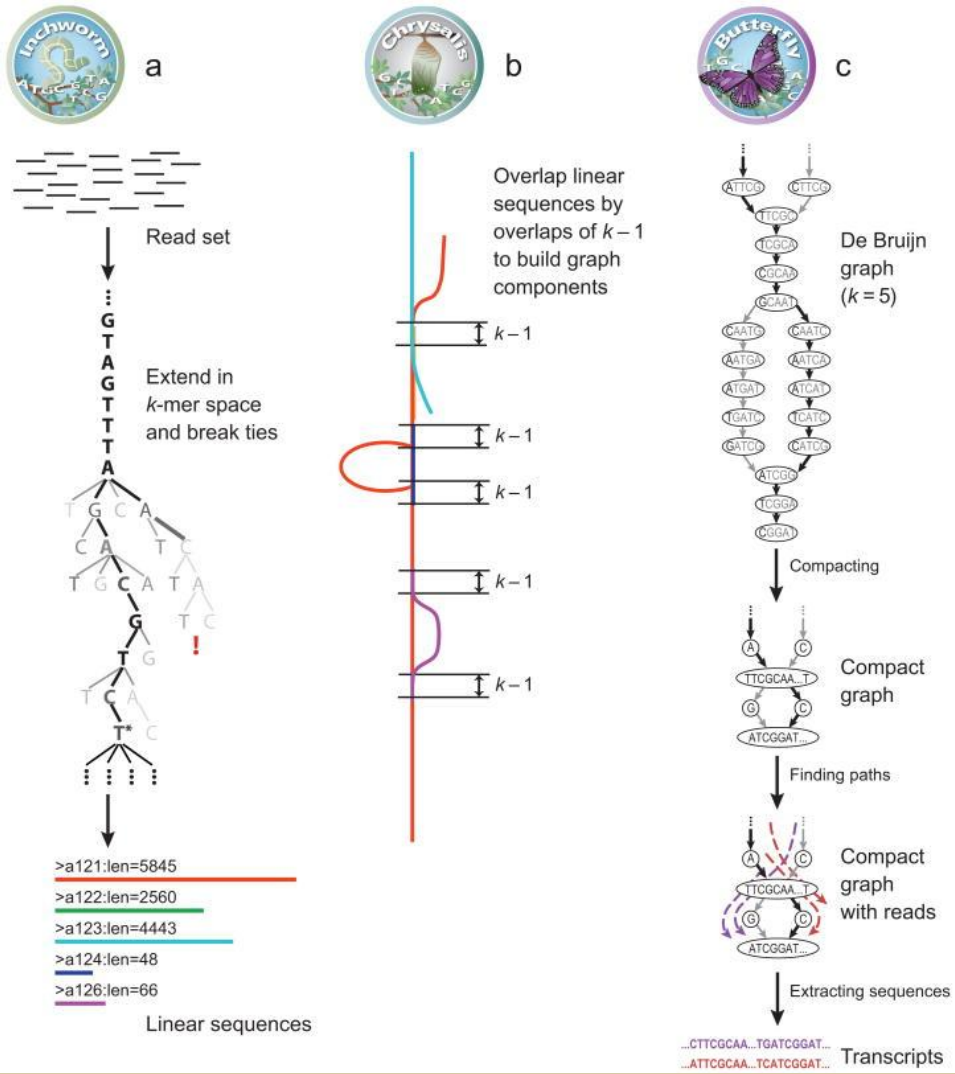
- You have:
  - Evaluated your sequences
  - Trimmed sequences
  - Normalized sequence input for assembly





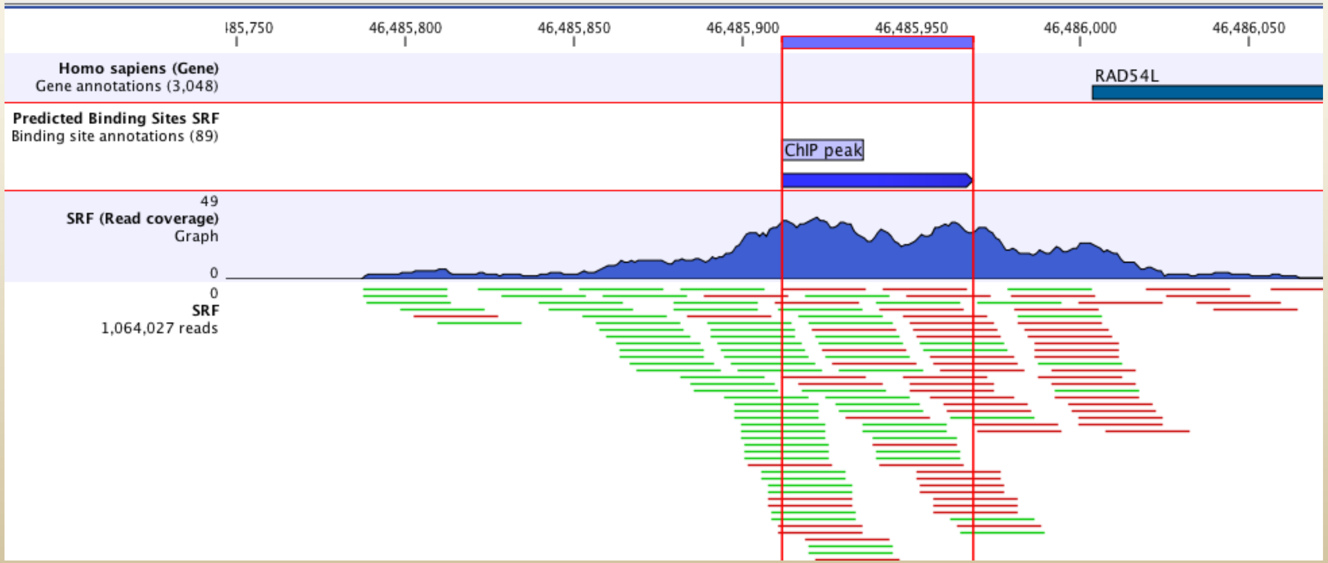
Recap

- You have:
  - Made (were given) an assembly(s)
  - Evaluated the assembly(s)
  - Looked at the annotation of Trinity_complete.fasta



Recap

- You have:
  - Mapped individual samples back to the assembly (RSEM/Bowtie)
  - Mapping provided raw read-pair sequence count per transcript



Differential expression analysis "software"

- Mostly performed in R (handles big datasets well)
- Mostly open source
- Several available through Bioconductor
- Can be performed locally on your laptop as well as on the cluster

Consider your experimental setup

- Before / after treatment
- With / without mutation
- Time-series
- Sample from different locations
- Descriptive focus only
- Controls
- +++

RNAseq sequence bias

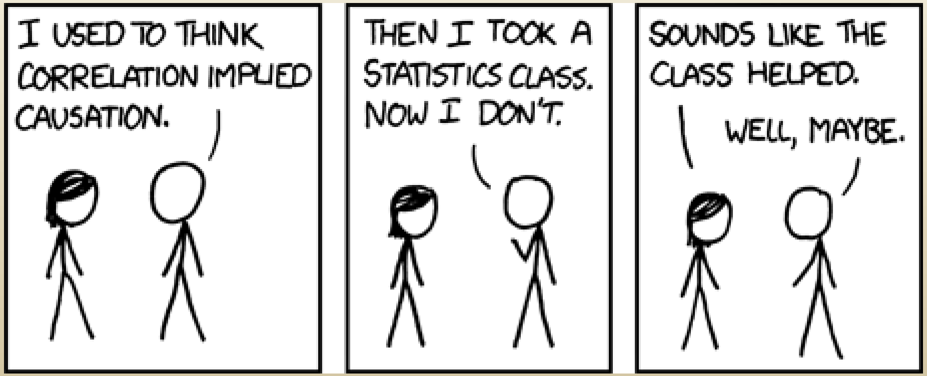
- Different technologies -> different bias
- Illumina example:
  - Bias in sequence is random = no homopolymer problem
  - Signs of hexamers in polyA-protocol
  - 5’ end or 3’ end bias in stranded protocol in relation to coverage
  - Lane bias
  - Batch effects
  - Inter-instrument bias
  - ...

RNAseq transcript bias

- High abundance transcripts
  - over-sampeled
- Low abundance transcripts
  - under-sampeled
- Library sequence bias
  - some libraries may become ”repeptitive” due to the PCR amplification step when too little RNA is used for prep
- Studies show that high-throughput RNAseq bias fit the **Negative Binominal Distribution** best

Statistics

- Be careful when concluding on your results!
- RNAseq is expensive -> often the number of samples / replicates is minimal
- Assumptions are made
- The less data you have the more assumptions are made!


I USED TO THINK CORRELATION IMPLIED CAUSATION.
THEN I TOOK A STATISTICS CLASS. NOW I DON'T.
SOUNDS LIKE THE CLASS HELPED.
WELL, MAYBE.

Data input

- For ALL analyses demonstrated in this course raw counts are used
- Make sure that you do not use FPKM, TMM, RPKM and similar normalized values as input
- The data is in the form of a matrix
  - Genes = rows
  - Samples = columns

Differential expression analysis

- Enter the world of R
- Perform several DE analyses
  - Simple no-replicate comparison DESeq
  - Simple replicate comparison DESeq
  - Simple replicate comparison edgeR

R ”syntax”

- #
  - comment. All after # will not be executed and you don’t have to type it
- \
  - means that the command is too long to fit on the slide in one line and continues on the next line
- >
  - the output that you can expect on screen

Avoid copying from slides when using R!

Use the read_me file

DESeq

- Can perform:
  - simple no-replicate comparisons
  - simple comparisons with replicates for one/both conditions
  - multi-factorial analyses

DESeq

- Effective library size calculated (normalization) using library size

DESeq

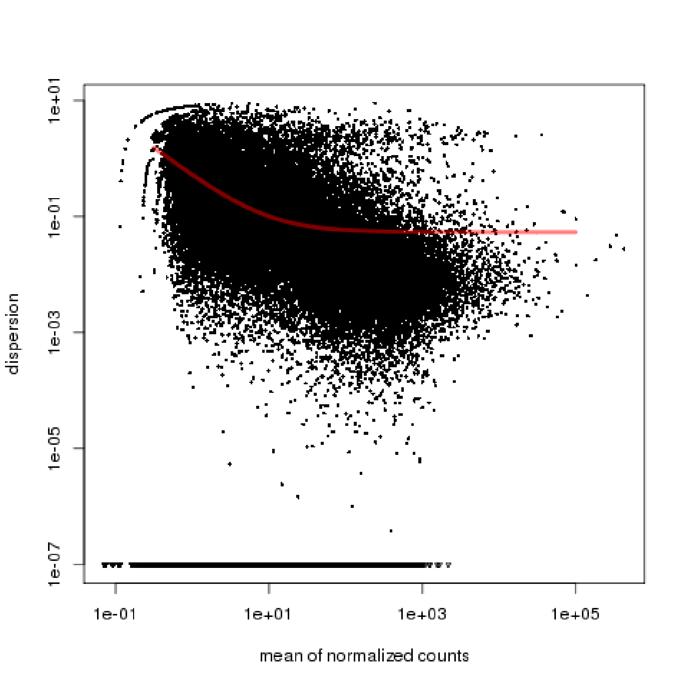
- Variance (dispersion)– the typical relationship between the data's variance and their mean

✓ Estimates dispersion per gene

✓ Fits a curve through the estimates

✓ Assigns a value to each gene.

  - Above line – the per gene estimate. Below line – the fitted estimate.



DESeq

- For DE with replicates:
  - Assuming negative binomial distribution
  - Null hypothesis is condition A = condition B
  - Assumes independent samples
  - Will for lowly expressed genes only report very high log fold changes as significant

DESeq

- How can I increase sensitivity using DESeq?
  - For lowly expressed genes deeper sequencing
  - For highly expressed genes more replicates

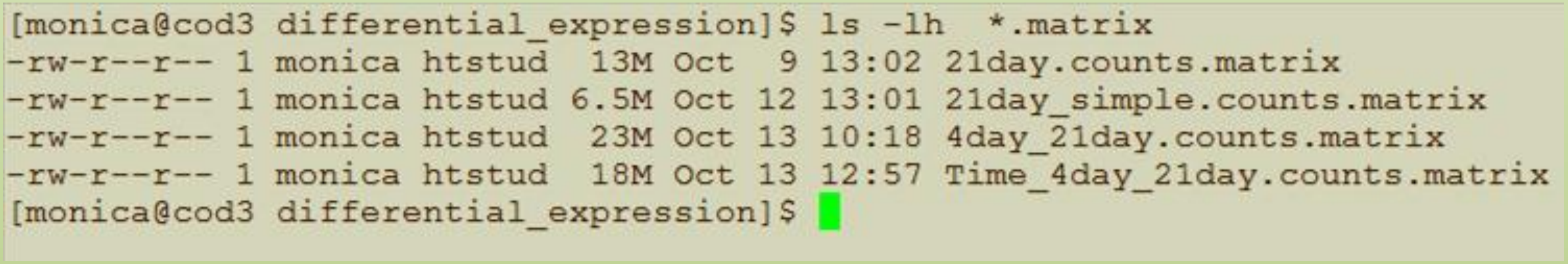
DESeq

- Why not DESeq2?
  - Has become more “black box”

Extract mapping results

Getting matrix file

- Make sure you have all the count matrices we will use today:



```
[monica@cod3 differential_expression]$ ls -lh  *.matrix
-rw-r--r-- 1 monica htstud  13M Oct  9 13:02 21day.counts.matrix
-rw-r--r-- 1 monica htstud 6.5M Oct 12 13:01 21day_simple.counts.matrix
-rw-r--r-- 1 monica htstud  23M Oct 13 10:18 4day_21day.counts.matrix
-rw-r--r-- 1 monica htstud  18M Oct 13 12:57 Time_4day_21day.counts.matrix
[monica@cod3 differential_expression]$
```

(Making matrix file)

- The matrix can be made like this:
- Copy all RSEM.genes.results to your home area and rename them accordingly before:

```
/cluster/software/VERSIONS/trinityrnaseq/\
trinityrnaseq-2.0.6/util/\
abundance_estimates_to_matrix.pl \
--est_method RSEM \
--out_prefix 21day \
21dayK_F4_RSEM.genes.results \
21dayK_F5_RSEM.genes.results \
21dayK_F6_RSEM.genes.results \
21dayV_F1_RSEM.genes.results \
21dayV_F2_RSEM.genes.results \
21dayV_F3_RSEM.genes.results
```

If you make the matrix yourself make sure to the load Trinity modules. Also some R related errors will occur. Delete all files except 21day.counts.matrix

R

- In the terminal write:

```
module load R
which R
>/cluster/software/VERSIONS/R-3.2.1/bin/R
```

- Start R by typing R and pressing enter. Your promt will change to >
- To quit R type q()

Support

- In /data/RNAseq2 there is a file called READ_ME_for_R
- This file contains commands so you can copy/paste to save time

R II

```
getwd()  # to get working directory
setwd("path")  # to change directory
list.files(path = ".")  # to list files in current directory
```

Example:

```
getwd()
>…/homedirs/<username>
```

- Check your working directory and change it if needed
- Make sure that the matrix file(s) is present in this directory

R III

```
library("DESeq")
library("edgeR")
library("gplots")
```

- Load the packages that you need
- Takes a few minutes / package
- Check your session before continuing (next slide)

R IV

- Session info must contain DEseq, EdgeR, gplots, and RColorBrewer

```
sessionInfo()

other attached packages:
[1] gplots_2.17.0        edgeR_3.10.5         limma_3.24.15
[4] DESeq_1.20.0         lattice_0.20-33      locfit_1.5-9.1
[7] Biobase_2.28.0       BiocGenerics_0.14.0

loaded via a namespace (and not attached):
 [1] AnnotationDbi_1.30.1 splines_3.2.1        IRanges_2.2.9
 [4] xtable_1.7-4         GenomeInfoDb_1.4.3   caTools_1.17.1
 [7] grid_3.2.1           KernSmooth_2.23-15   DBI_0.3.1
[10] genefilter_1.50.0    gtools_3.5.0         survival_2.38-3
[13] geneplotter_1.46.0   RColorBrewer_1.1-2   S4Vectors_0.6.6
[16] bitops_1.0-6         RSQLite_1.0.0        gdata_2.17.0
[19] stats4_3.2.1         XML_3.98-1.3         annotate_1.46.1
```

A simple no-replicate comparison

Simple no-replicate comparison DESeq

- Why bother?
  - So you can get familiar with the DESeq package ☺
- Can I actually use this example?
  - In certain cases: yes
  - Strong confidence: no
  - Preliminary overview of data: yes
- What is reported?
  - STRONGLY up- or down-regulated genes that conquer the data's noise

Assumptions

- The mean of both the treated and untreated sample is used as estimate for dispersion (variability)
  - Thus we assume that the change in condition only affects a small number of genes
  - This test is very conservative because DE genes will increase the dispersion estimate and thus "camouflage" lower DE genes

Data read-in

```
data1= ("21day_simple.counts.matrix")
CountTable1 = read.table(data1, \
header=T, row.names=1, com='')
CountTable1 = round(CountTable1)
head(CountTable1)
              X21dayK_F4 X21dayV_F1
c96089_g1             0          0
c164959_g1            0          0

dim CountTable1
     >[1] 320520      2
```

- Read in the 21 day simple comparison matrix
- Make a table object called CountTable1
- Round CountTable1 to remove decimals
- Look at CountTable1
- Make sure dimensions correspond

Experiment factors

```
condition1 = \
factor(c("untreated","treated"))
> condition1
[1] untreated treated
Levels: treated untreated
```

- After read-in we will make a condition object and store the condition of the two samples
- Condition contains the factors DESeq will consider later on
- Controls always has to be first!

Count / factor object

```
cds1 = newCountDataSet (CountTable1,
\ condition1)
head(cds1)

>CountDataSet (storageMode:
environment)
assayData: 1 features, 2 samples
  element names: counts
protocolData: none
phenoData
  sampleNames: X21dayK_F4 X21dayV_F1
…………
```

- Then we will combine the factors and the count table into object cds

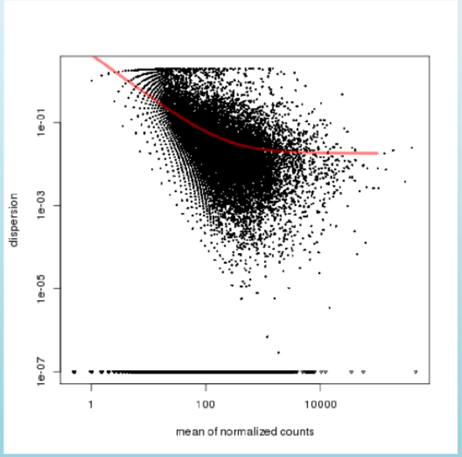
Normalization

```
cds1 =estimateSizeFactors(cds1)
> sizeFactors(cds1)
X21dayK_F4 X21dayV_F1
         1          1
head(counts(cds1, normalized=TRUE))
           X21dayK_F4 X21dayV_F1
c96089_g1           0          0
c164959_g1          0          0
c156204_g1          0          0
c205267_g1          0          0
c125263_g2        866        606
c251429_g1          0          0
```

- Library size estimation for sample normalization
- These samples are almost identical in size
- You can check the effect of the normalization (for samples with factor ≠ 1

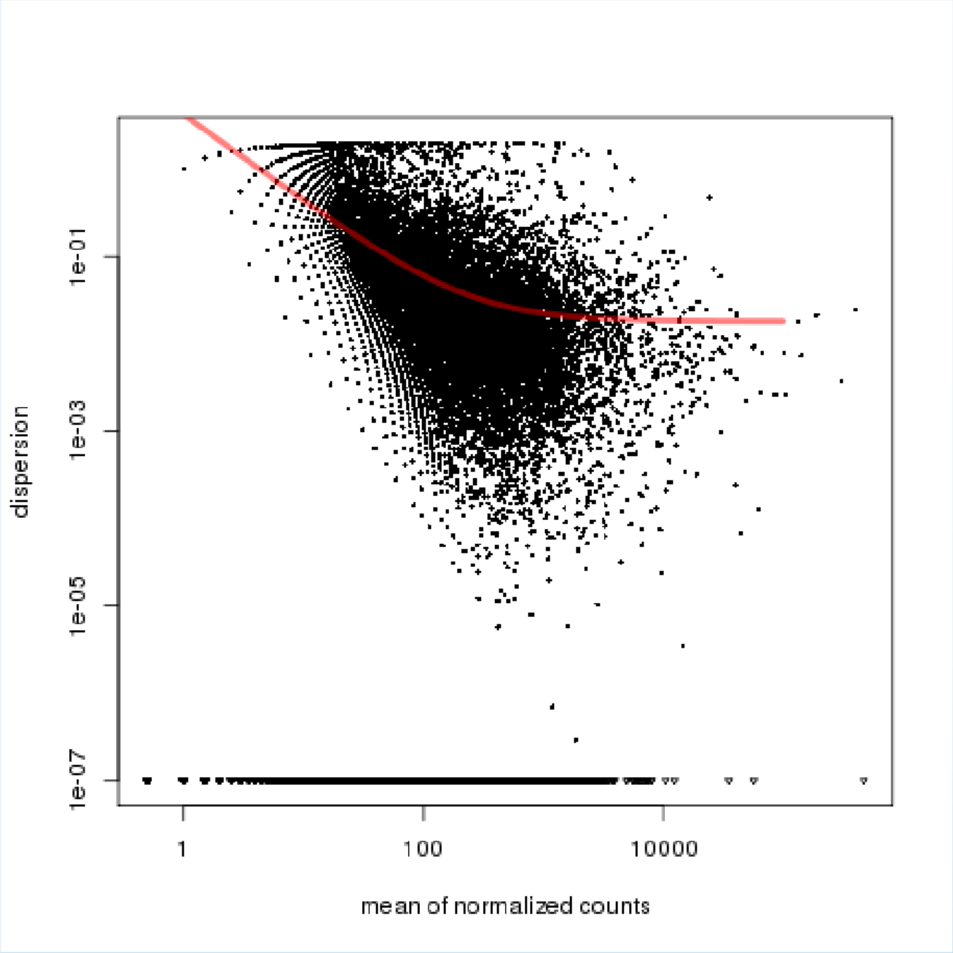
Dispersion estimation

```
cds1 = estimateDispersions \
(cds1, method="blind", \
sharingMode="fit-only")


png
("21day_simple_dispersion.png")
plotDispEsts( cds1 )
dev.off()
```



- Estimate dispersion (variability) across conditions and ignore outliers (fit-only)
- Then plot the dispersion
- Transfer the file to look at the plot

Dispersion estimation - II



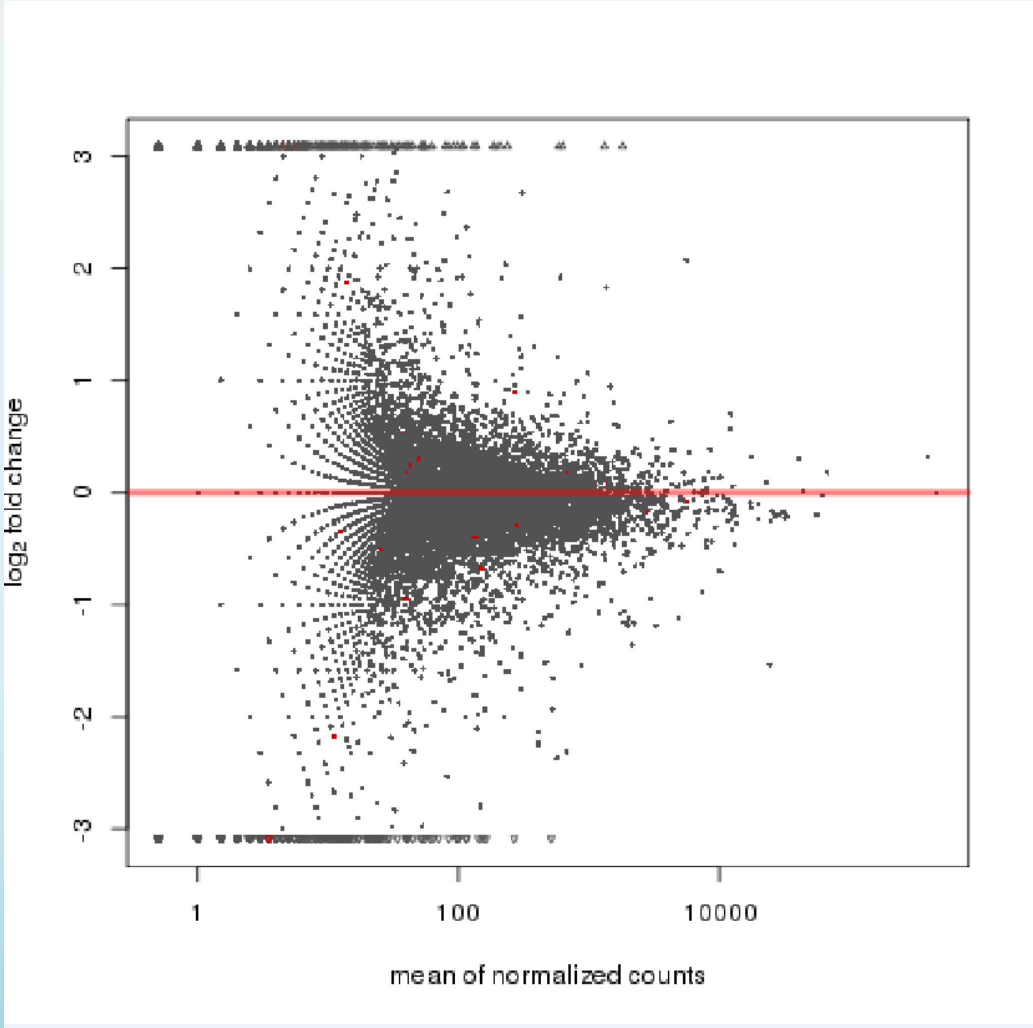
- The dispersion fit line will be skewed for such a simple comparison

Negative binomial test

```
res1 = nbinomTest \
(cds1, "untreated", "treated")

# takes a few minutes


png ("21day_simple_plotMA.png")
plotMA(res1)
dev.off()
```

- Run the simple negative binomial test
- Plot your data
- If plotMA gives an error quit your session and start over using commands from the R_commands_ clean file

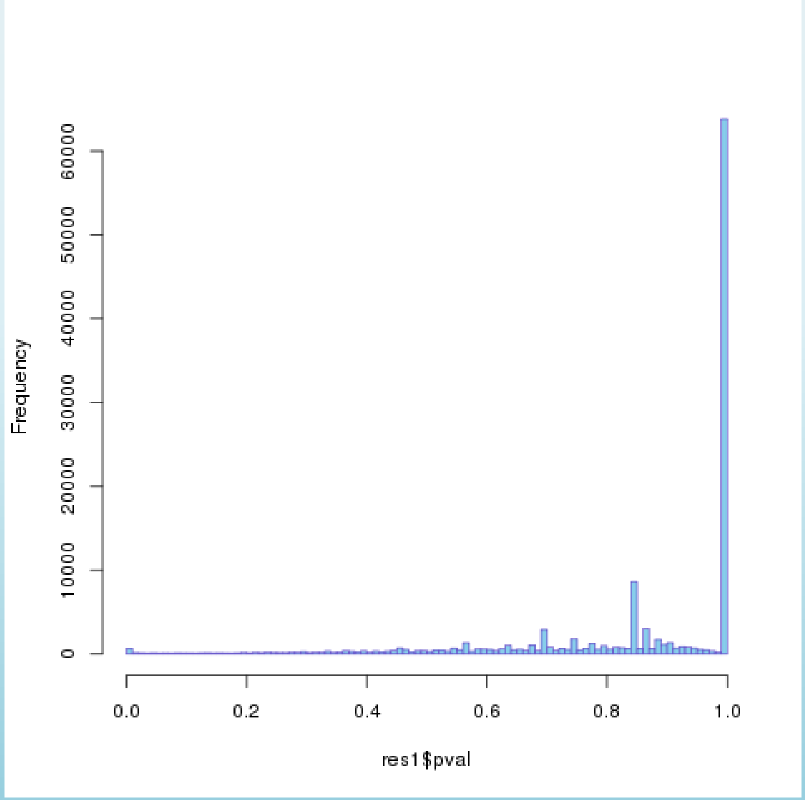
Negative binomial test – II



- A simple comparison will usually yield few significant results

P-value histogram

```
png ("21day_simple_pvalue_histo.png")
hist(res1$pval, breaks=100, col="skyblue", \
border="slateblue", main="")
dev.off()
```



Looking at results

```
head( res1[order(res1$pval), ] )
# sorting just to look

res1Sig = subset(res1, padj<0.1)
# filtering 10 % false discovery

dim(res1Sig)
[1] 276      8
```

- A simple comparison will usually yield few significant results containing several false positives

Save results

```
write.csv( res1, file="21day_simple_DEanalysis.csv”)


write.csv(res1Sig, \
file="21day_simple_DEanalysis_significant.csv" )
```

- Print all results to file
- Write significant results to file

A simple 3x replicate comparison

Data read-in

```
data2 = ("21day.counts.matrix ")
CountTable2 = read.table(data2, \
header=T, row.names=1, com='')
CountTable2 = round(CountTable2)
head(CountTable2)
dim(CountTable2)
>[1] 320520      6
```

- Read in the 21 day comparison matrix
- Make a table object called CountTable2
- Round CountTable2 to remove decimals
- Look at CountTable2
- Make sure dimensions correspond

Experiment factors

```
condition2 = \
factor(c("untreated","untreated“,\
"untreated","treated","treated",\
"treated"))
condition2
>[1] untreated untreated untreated
treated   treated   treated
Levels: treated untreated
```

- After read-in we will make a condition object and store the condition of the two samples
- Condition contains the factors DESeq will consider later on
- Controls always has to be first!

Count / factor object

```
cds2 = newCountDataSet(CountTable2,condition2)
head(cds2)
```

```
CountDataSet (storageMode: environment)
assayData: 1 features, 6 samples
  element names: counts
protocolData: none
phenoData
  sampleNames: X21dayK_F4 X21dayK_F5 ... X21dayV_F3 (6
total)
  varLabels: sizeFactor condition
  varMetadata: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
Annotation:
```

Normalization

```
cds2 = estimateSizeFactors(cds2)
sizeFactors(cds2)
```

```
X21dayK_F4 X21dayK_F5 X21dayK_F6
0.9208898 1.0921015 1.3786340

X21dayV_F1 X21dayV_F2 X21dayV_F3
0.8756682 0.9568349 0.9119786
```

Data visualization

- With replicates you can make various plots to visualize your data
- Initial presentation of data enables you to adjust your statistics and remove outliers

Heat map presentations - I

```
library("RColorBrewer")



cds2Blind = \
estimateDispersions(cds2, \
method="blind")


vsd2 =\
varianceStabilizingTransformation\
(cds2Blind)
```

- Heatmaps and PCA describes the data in a more pleasing visual way
- Not very informative if run on the previous example, but give it a go if you want
- First we make some assumptions and prelim analyses

Heat map presentations - II

```
select =
order(rowMeans(counts(cds2)),decreasin
g=TRUE)[1:30]
hmcol = colorRampPalette(brewer.pal(9,
"GnBu"))(100)

png("21day_heatmap_transformed.png")
heatmap.2(exprs(vsd2)[select,], col=
hmcol, trace="none", margin=c(10,6))
dev.off()

png("21day_heatmap_untransformed.png")
heatmap.2(counts(cds2)[select,], col=
hmcol, trace="none", margin=c(10,6))
dev.off()
```

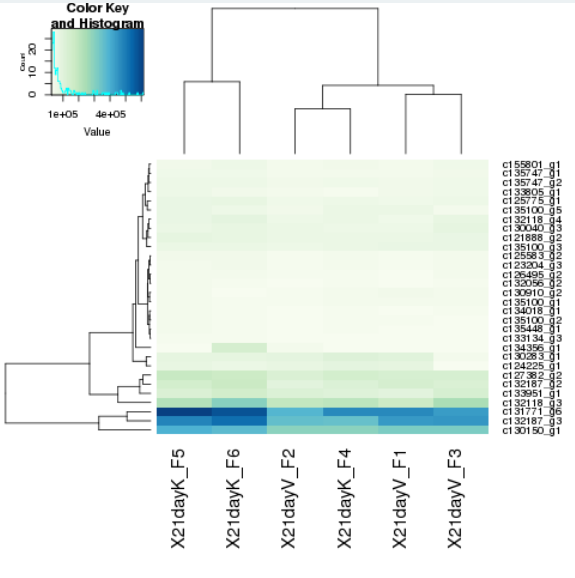
- Adjusting some color settings and making transformed and untransformed heatmaps
- See commands in the R_commands_clean file
- Copy/paste commands

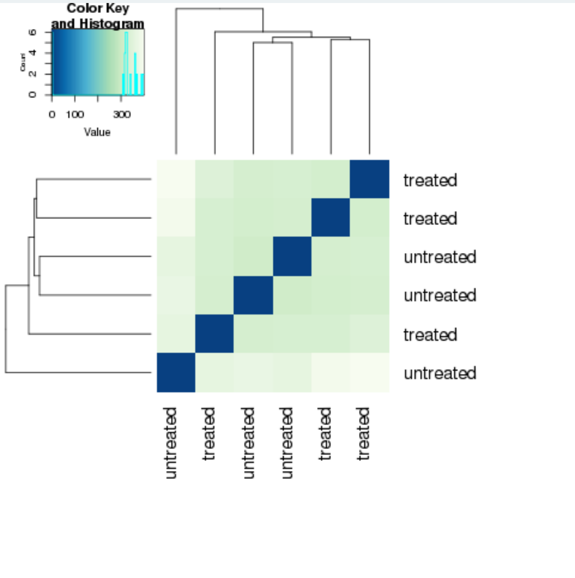
Heat map presentations - III

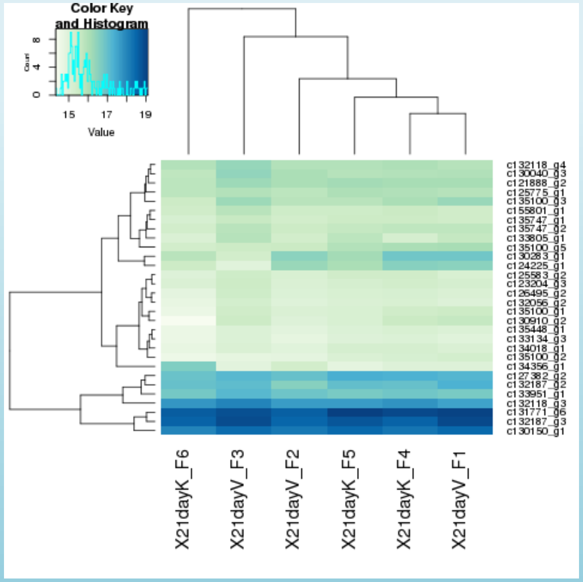
```
dists = dist( t( exprs(vsd2) ) )

mat = as.matrix( dists )
rownames(mat) = colnames(mat) =
with(pData(cds2Blind), paste(condition,
sep=" : "))
heatmap.2(mat, trace="none", col =
rev(hmcol), margin=c(13, 13))


png ("21day_heatmap_distance_matrix.png")
heatmap.2(mat, trace="none", col =
rev(hmcol), margin=c(13, 13))
dev.off()


png ("21day_PCA.png")
print(plotPCA(vsd2,
intgroup=c("condition")))
dev.off()
```

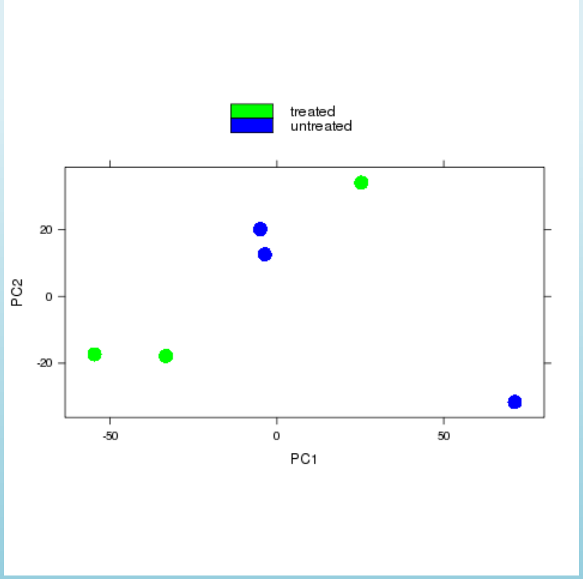
- Distance matrix and PCA plots
- Are there any outliers?
- Consider removing them but take care!

Heat map presentations - IV









Differential expression analysis

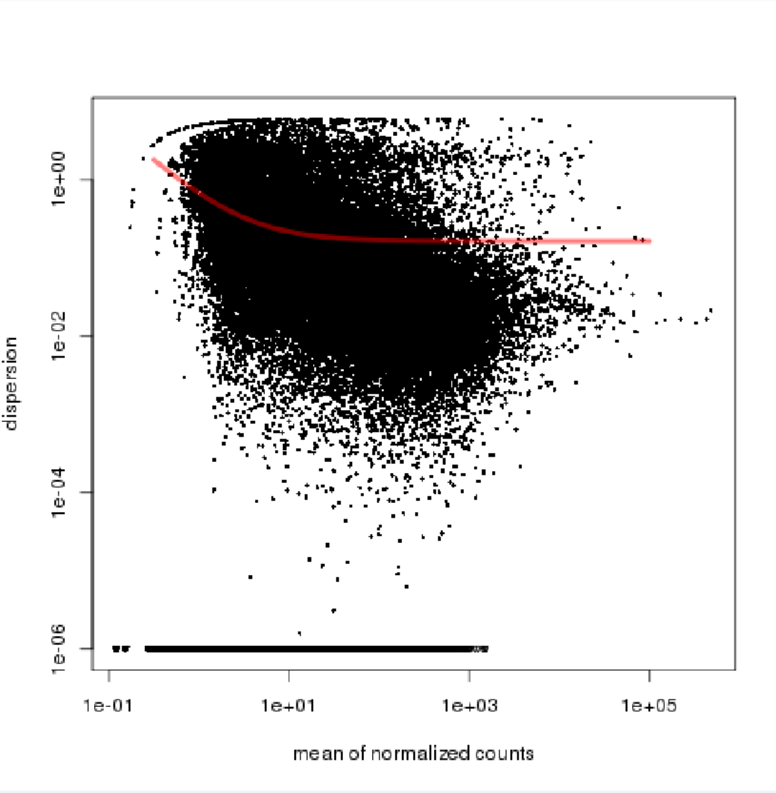
Dispersion estimation

```
cds2 = estimateDispersions(cds2)
png ("21day_dispersion.png")
plotDispEsts( cds2 )
dev.off()
```

Dispersion estimation - II

```
cds2 = estimateDispersions(cds2)
png ("21day_dispersion.png")
plotDispEsts( cds2 )
dev.off()
```

- The dispersion has improved but is still a bit skewed

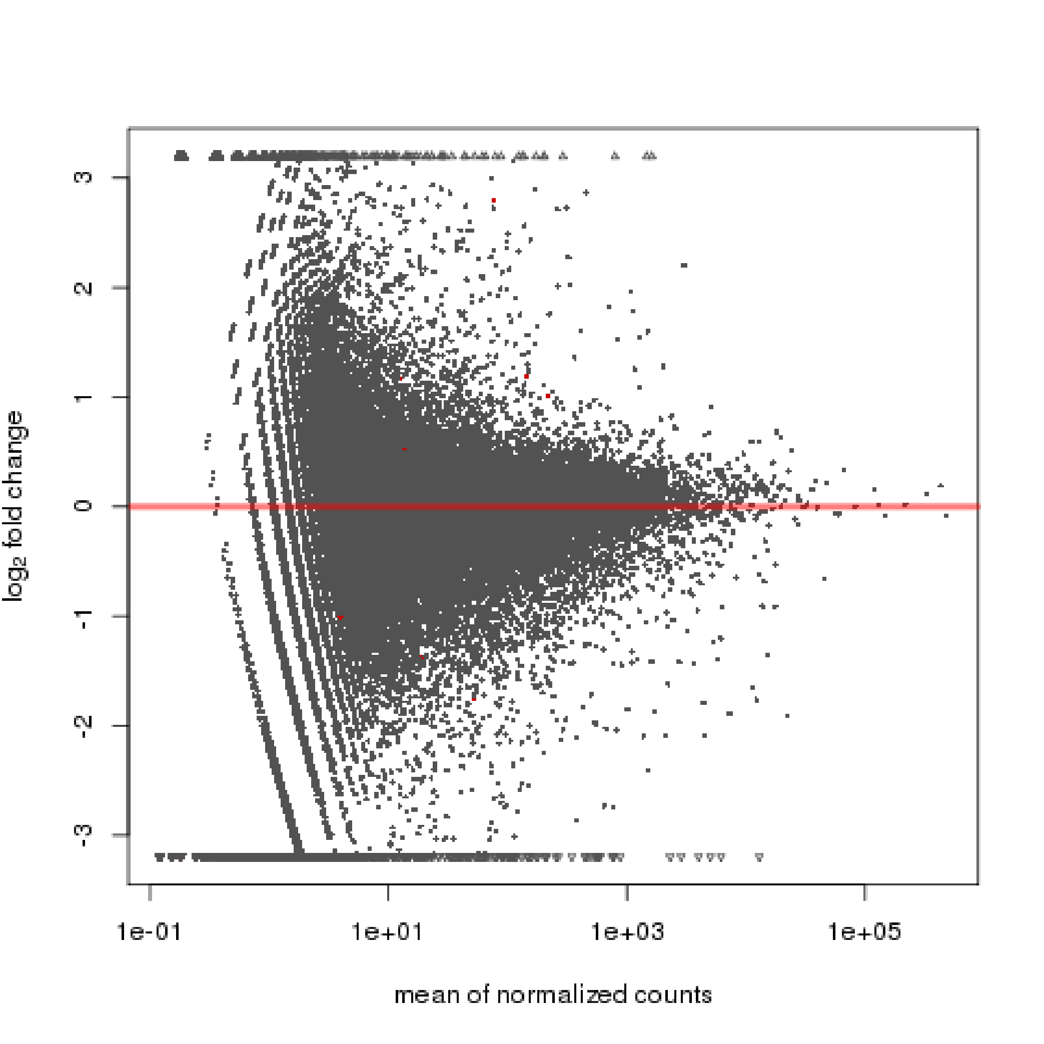


Negative binomial test

```
res2 = nbinomTest(cds2, \
"untreated", "treated")
png ("21day_plotMA.png")
plotMA(res2)
dev.off()
```

- If plotMA gives an error quit your session and start over using commands from the R_commands_clean file

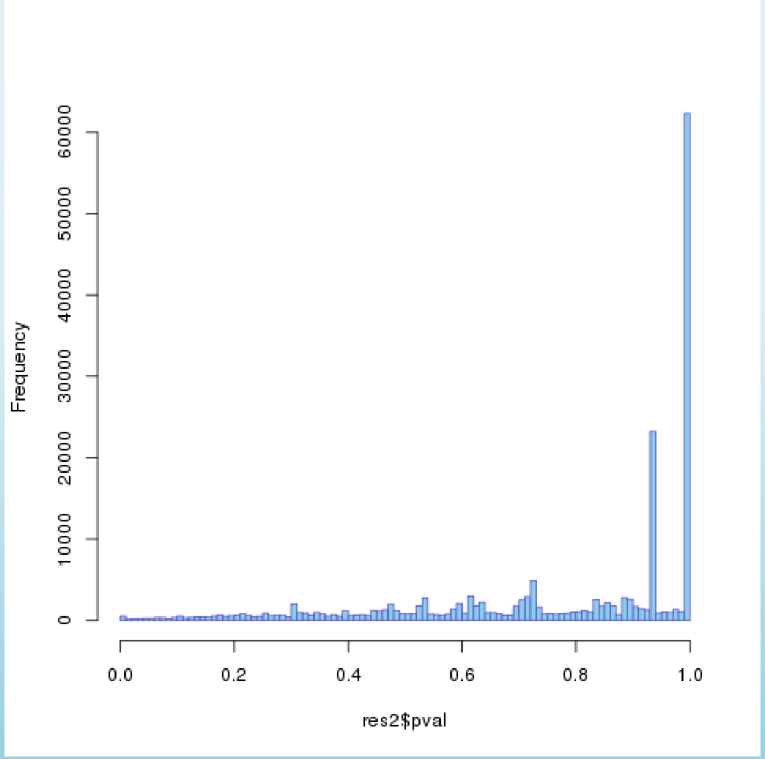
Negative binomial test - II



- The simple comparison gave quite a few false positives
- The replicates adjusts for this

P-value histogram

```
png ("21day_pvalue_histo.png")
hist(res2$pval, breaks=100, col="skyblue", \
border="slateblue", main="")
dev.off()
```



Looking at data

```
head( res2[order(res2$pval),] )
# sorting just to look

res2Sig = subset(res2,padj<0.1)
# filt. 10 % false discovery

dim(res2Sig)
>[1] 133      8
```

- The simple comparison gave quite a few false positives
- The replicates adjusts for this

Save data for later

```
write.csv( res2, \
file="21day_DEanalysis.csv" )

write.csv(res2Sig, \
file="21day_DEanalysis_significant.csv" )
```

Same experiment, different package

Multifactorial designs EdgeR

- Uses the Negative Binominal Distribution
- Exact test (ET) or Generalized linearized model (GLM)
- Dispersion is modelled with a maximum likelihood model
  - ET: quantile-adjusted conditional maximum likelihood method
  - GLM: Cox-Reid profile adjusted likelihood method

Multifactorial designs EdgeR II

- Additional filtering of lowly expressed genes across all samples
- Also handles no-replicate data but not with a detailed method such as DESeq

Same experiment, different package

```
counts <- read.delim("21day.counts.matrix", \
row.names=1, header=TRUE, stringsAsFactors=FALSE)

names(counts) <- c('21dayK1', '21dayK2', '21dayK3',\
'21dayV1', '21dayV2', '21dayV3')

head(counts)

#21dayK1 21dayK2 21dayK3 21dayV1 21dayV2 21dayV3
```

Same experiment, different package II

```
#make grouping factors
group <- c(rep("A", 3) , rep("B", 3))

#make DGEList object called cds
cds <- DGEList (counts , group = group)

names(cds)
#[1] "counts"  "samples"

levels(cds$samples$group)
#[1] "A" "B"
```

Same experiment, different package III

```
#Some filtering and normalization

cds <- cds[rowSums(1e+06 * \
cds$counts/expandAsMatrix(cds$samples$lib.size, \
dim(cds)) > 1) >= 3, ]

cds <- calcNormFactors( cds )
```

Same experiment, different package IV

```
#MDS plot

png( "MDS_21day_edgeR.png" )
plotMDS( cds , main = "MDS_Plot_for_Count_Data", \
labels = colnames( cds$counts ) )
dev.off()
```

Same experiment, different package V

```
# making the design matrix
design <- model.matrix(~0+group, data=cds$samples)

head(design)

colnames(design) <- levels(cds$samples$group)

head(design, n=10L)
```

Same experiment, different package VI

```
# estimating dispersion three ways - will take a few
minutes each

cds <- estimateGLMCommonDisp( cds, design )

cds <- estimateGLMTrendedDisp( cds, design )

cds <- estimateGLMTagwiseDisp( cds, design )


png ("Dispersion_21day_edgeR.png")
plotBCV(cds)
dev.off()
```

Same experiment, different package VII

```
#plotting experiment variance
png ("meanVarPlot_21day_edgeR.png")
meanVarPlot <- plotMeanVar( cds , \
show.raw.vars=TRUE ,
show.tagwise.vars=TRUE ,
show.binned.common.disp.vars=FALSE ,
show.ave.raw.vars=FALSE ,
NBline = TRUE ,
nbins = 100 ,
pch = 16 ,
xlab ="Mean Expression (Log10 Scale)" ,
ylab = "Variance (Log10 Scale)" ,
main = "Mean-Variance Plot" )
dev.off()
```

Same experiment, different package VIII

```
 # Running the DE analysis


fit <- glmFit(cds, design)

my.contrasts <- makeContrasts(BvsA=groupB-groupA, \
levels=design)

head(my.contrasts)
```

Same experiment, different package IX

```
lrt.BvsA <- glmLRT(fit, contrast=c(-1,1))
topTags(lrt.BvsA)

time21 <- topTags(lrt.BvsA, n=nrow(lrt.BvsA$table),\
sort.by = "p.value")

write.table(time21, file = \
"time21_glmFit_adjustedpvalues", quote = FALSE, \
row.names = TRUE, sep = "\t")
```

Same experiment, different package X

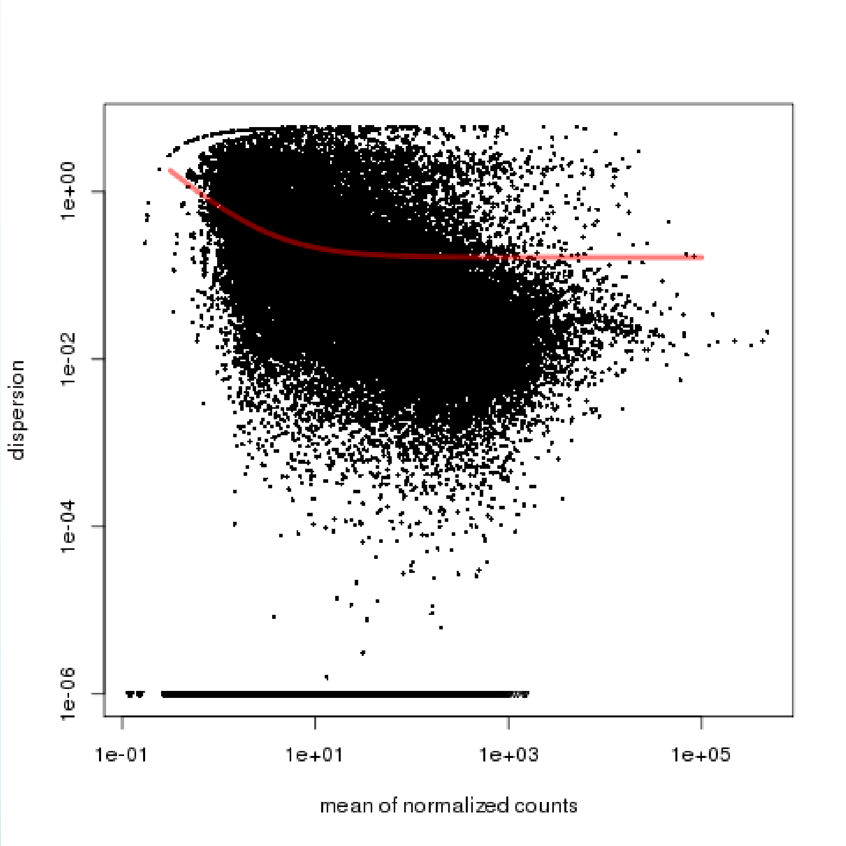
```
time21sign <- topTags(lrt.BvsA, \
n=nrow(lrt.BvsA$table), sort.by = "p.value",  \
p.value=0.05)

write.table(time21sign, file = \
"time21_glmFit_adjustedpvalues_degenes", \
quote = FALSE, row.names = TRUE, sep = "\t")
```

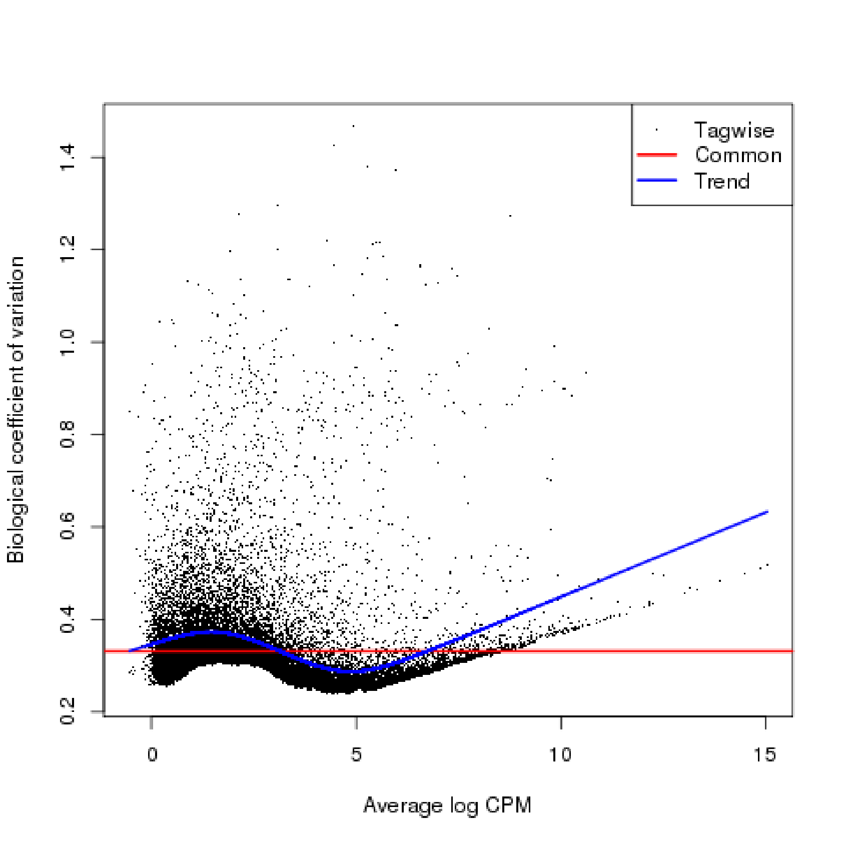
Same experiment, different package XI

```
#The total number of differentially expressed
genes at 5% FDR is given by:

summary(deBvsA <- decideTestsDGE(lrt.BvsA))

png ("DE_21day_glm_edgeR.png")
detagsBvsA <- rownames(cds)[as.logical(deBvsA)]
plotSmear(lrt.BvsA, de.tags=detagsBvsA)
abline(h=c(-1, 1), col="blue")
dev.off()
```

Same experiment, different package XI

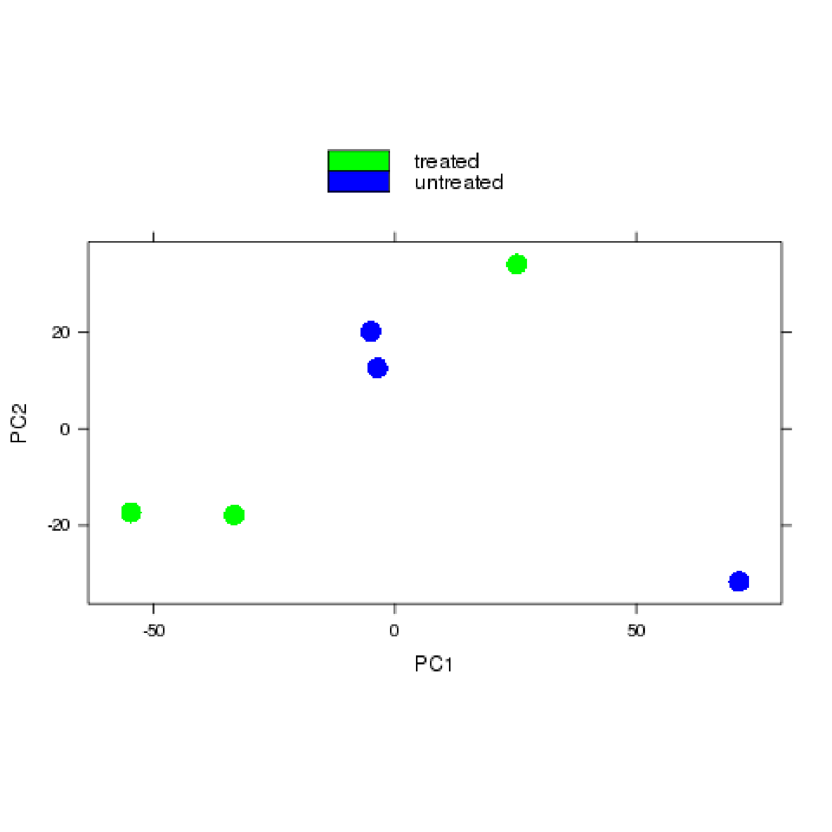
Compare the outputs of the two analyses. Dispersion:

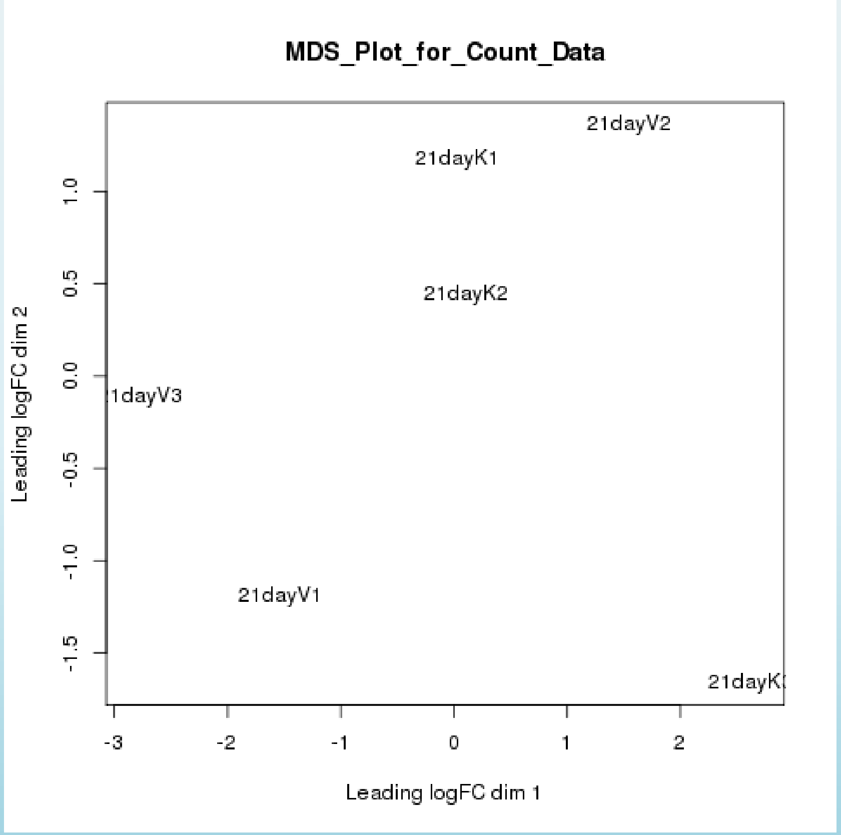




Same experiment, different package XII

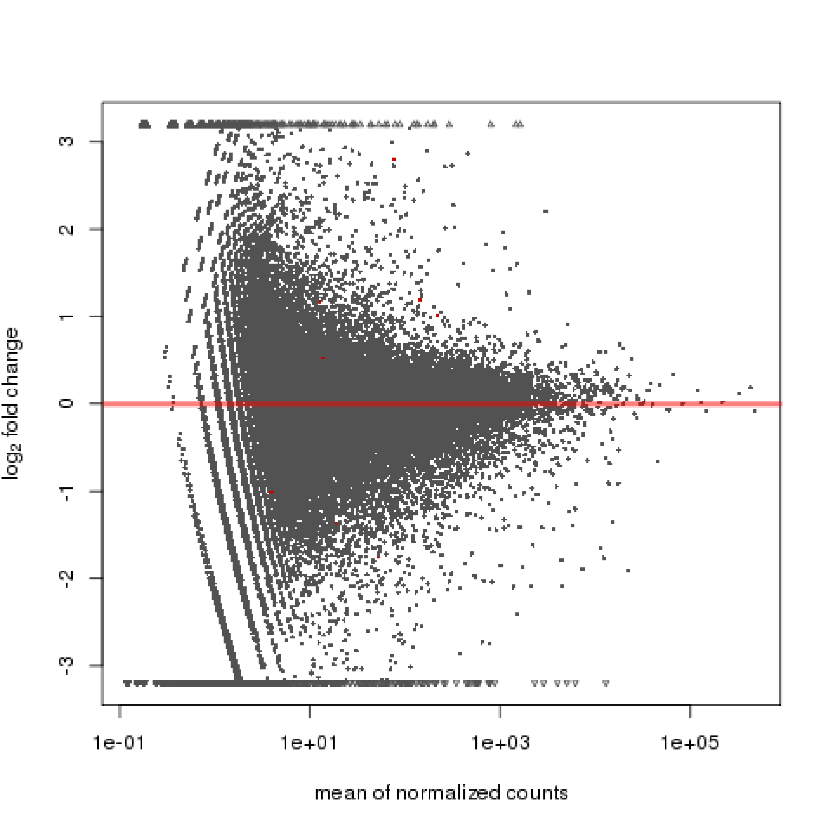
Compare the outputs of the two analyses. PCA vs MDS

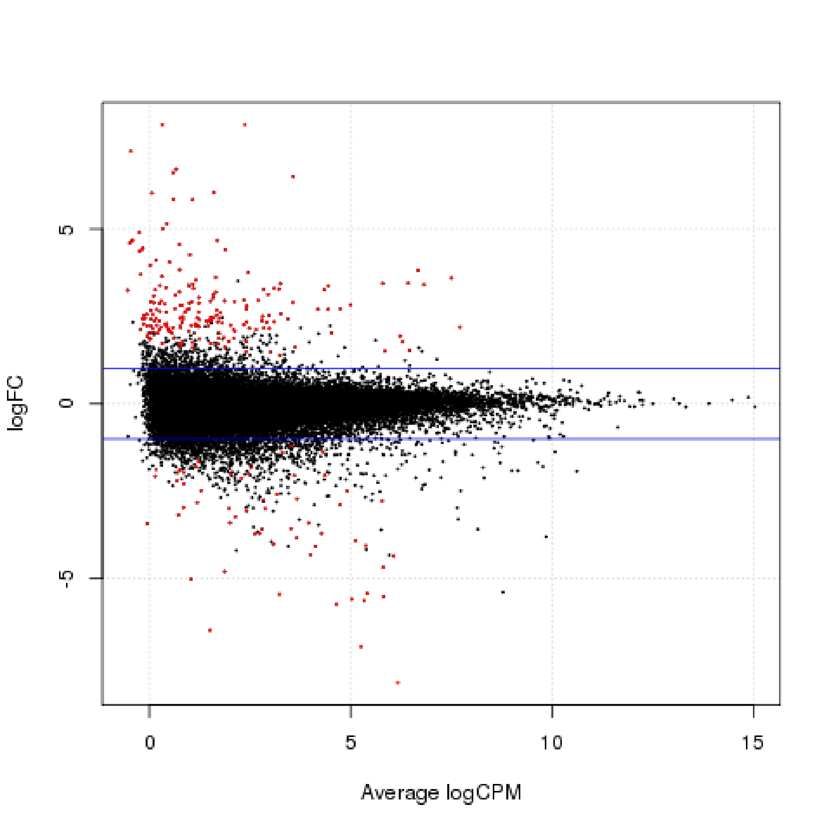




Same experiment, different package XIII

Compare the outputs of the two analyses. DE genes





Same experiment, different package XIII

Compare the outputs of the two analyses. DE genes II

DESeq no replicate: 276

DESeq genes w/ FDR 10 %: 133

EdgeR genes w/ BH FDR: 253

Multifactorial designs EdgeR III

- DESeq vs edgeR
  - edgeR is anti-conservative for lowly expressed genes whereas DESeq is conservative
  - edgeR is conservative for highly expressed genes
  - Similar type-I error control on average
    - Type I: incorrect rejection of null hypothesis (false positive)
    - Type II: failure to reject a false null hypothesis (false negative)