What we did yesterday

(Good job, everybody!)

Start with fastq files
(like you'd get from the sequencer)

@HWI-D00220:61:C2RBCACXX:3:1101:1473:1951 1:N:0:TAGCTT

NTTCAACTTGAACTGTTACCTGTAATGTCAGTTTGTATCAATTTTTGTTCC

+

#0<FFFFFFFFFFIIIIIIIIIIFIIIFIFFIIIIFIFIFIIIIIIIIIIF

Start with fastq files
(like you'd get from the sequencer)

Exact position on flowcell of read

Barcode of sample

@HWI-D00220:61:C2RBCACXX:3:1101:1473:1951 1:N:0:TAGCTT

Sequence of read (50 bases here)

NTTCAACTTGAACTGTTACCTGTAATGTCAGTTTGTATCAATTTTTGTTCC

\+ ← they put an empty line in the format just in case

#0<FFFFFFFFFIIIIIIIIIIFIIIFIFFIIIIFIFIFIIIIIIIIIIF

Sanger quality scores:

worst

best

!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~

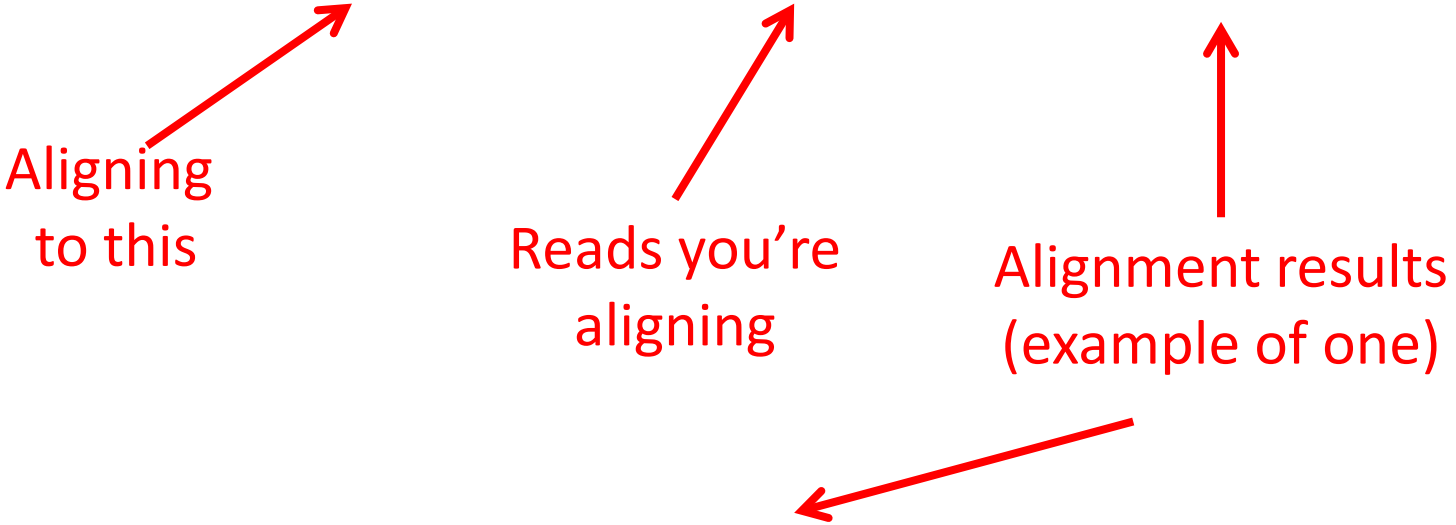
Make an index from a fasta file

$ bowtie2-build name_of.fasta choose_an_index_prefix

>Scaffold1000
TGTCCAATGTGAAAGTATCACATGTACTGGTGGGACTATAGGAGAGGC
CTTTCTGAAGTACATTCAACTGCAGGGGAGTC
AATACATTGTCAGAAATGTTATATACTAACGTTTCGATTGAGTCCTCCAT
GCTGATCGTGGCGGTTGCTGC
>Scaffold10000
GTTGTAAGTGCATCATCTTTTCTTGCTGGAATTACTTCCACCCACTTAG
AGAATACATCTATTACCACTAAGGCATAACG
TAGGCCATACCTACCAGATGGTAGTGGTCCTATGTAGTCAATTTGGAGT

Align reads in fastq file to index

$ bowtie2 –X index_prefix –U reads.fastq –S my_align.sam



LAMARCK:3118:C067BACXX:2:2304:1278:103673 0
MCIDAS|ENSG00000234602|c.Chung201110_X003550|JGIv7b.000052441_12586078-12589763- 608 42 51M * 0 0
CAAAAATCAAAATGCATGGAGCATTTCACGGACTGAAAACATCCACAGGGC
@@@BFDFFHHHHGJEGIIIII>GCHGIIGGIIIGIJGIGGGCEFHIIGGIE AS:i:0
XN:i:0 XM:i:0 XO:i:0 XG:i:0 NM:i:0 MD:Z:51 YT:Z:UU

Count reads with eXpress

$ express name_of.fasta my_align.sam

Original fasta file

Where each read aligns to in fasta

output

bundle_id target_id length eff_length tot_counts uniq_counts est_counts eff_counts ambig_distr_alpha ambig_distr_beta fpkm fpkm_conf_low fpkm_conf_high solvable tpm

1 ATP5SL|ENSG00000105341|c.Amin201106_X009956|JGIv7b.000039723_9205370-9214135+ 1099 999.274749 102 102 102.000000 112.179358 0.000000e+00 0.000000e+00 2.940272e+00 2.940272e+00 2.940272e+00 T

Count reads with eXpress

$ express name_of.fasta my_align.sam

Original fasta file

Where each read aligns to in fasta

Counts we want

bundle_id target_id length eff_length tot_counts uniq_counts est_counts eff_counts ambig_distr_alpha ambig_distr_beta fpkm fpkm_conf_low fpkm_conf_high solvable tpm

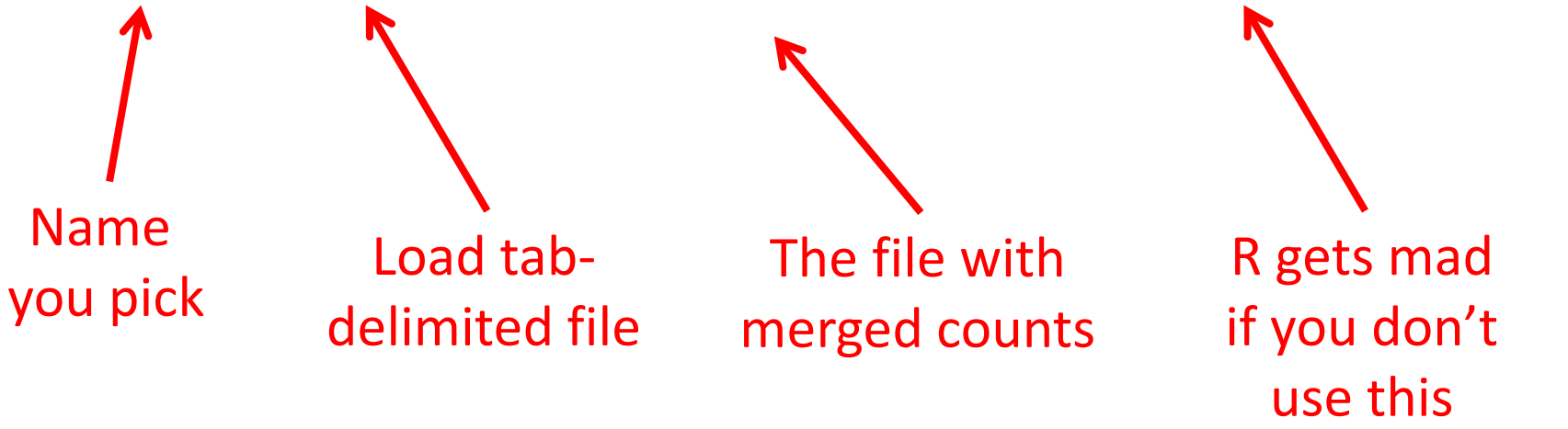
1 ATP5SL|ENSG00000105341|c.Amin201106_X009956|JGIv7b.000039723_9205370-9214135+ 1099 999.274749 102 102 102.000000 112.179358 0.000000e+00 0.000000e+00 2.940272e+00 2.940272e+00 2.940272e+00 T

Merge counts (‘eff_counts’) from experiments into single table

- I gave you a file from a bunch of weird scripts I have
- I also gave you a janky approach using “cut”, “sort”, and “join”
- Leon generated an awk script
- Virginia generated an R script
- If you’re impatient you can use excel, but other instructors will yell/laugh at me (also might not work)
- I’ll figure out the best one today

Fire up R

```
>mydata = read.delim("merged_data.txt",stringsAsFactors=FALSE)
```



Run DESeq in R

>mydata = read.delim("merged_data.txt",stringsAsFactors=FALSE)

To check:
> head(mydata)

Condition 1 counts

Condition 2 counts

LLGL2	725.869645	519.156491	621.864761	531.770848
DYNC2LI1	448.907891	299.038396	225.772704	258.203116
PPHLN1	635.317731	580.306441	572.638483	444.313247
PCSK9	0	0	16.07931	11.739452

Run DESeq in R

>rownames(mydata) = mydata[,1]

Name the rows of my data ("mydata")

… after the first column of the R object "mydata"

LLGL2	725.869645	519.156491	621.864761	531.770848
DYNC2LI1	448.907891	299.038396	225.772704	258.203116
PPHLN1	635.317731	580.306441	572.638483	444.313247
PCSK9	0	0	16.07931	11.739452

Run DESeq in R

```
>rownames(mydata) = mydata[,1]
```

Name the rows of my data ("mydata")

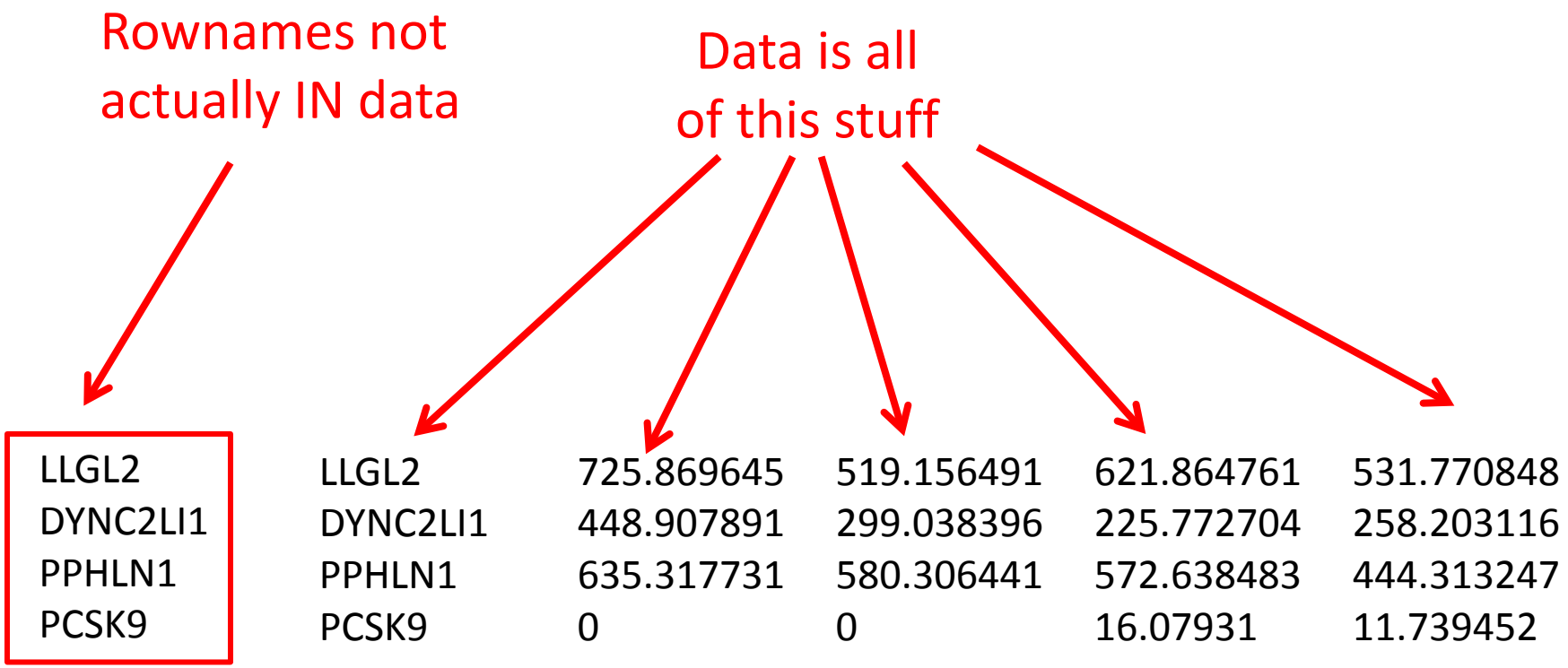
… after the first column of the R object "mydata"

First digit = column
Second digit = row

LLGL2	725.869645	519.156491	621.864761	531.770848
DYNC2LI1	448.907891	299.038396	225.772704	258.203116
PPHLN1	635.317731	580.306441	572.638483	444.313247
PCSK9	0	0	16.07931	11.739452

Run DESeq in R

>rownames(mydata) = mydata[,1]



Run DESeq in R

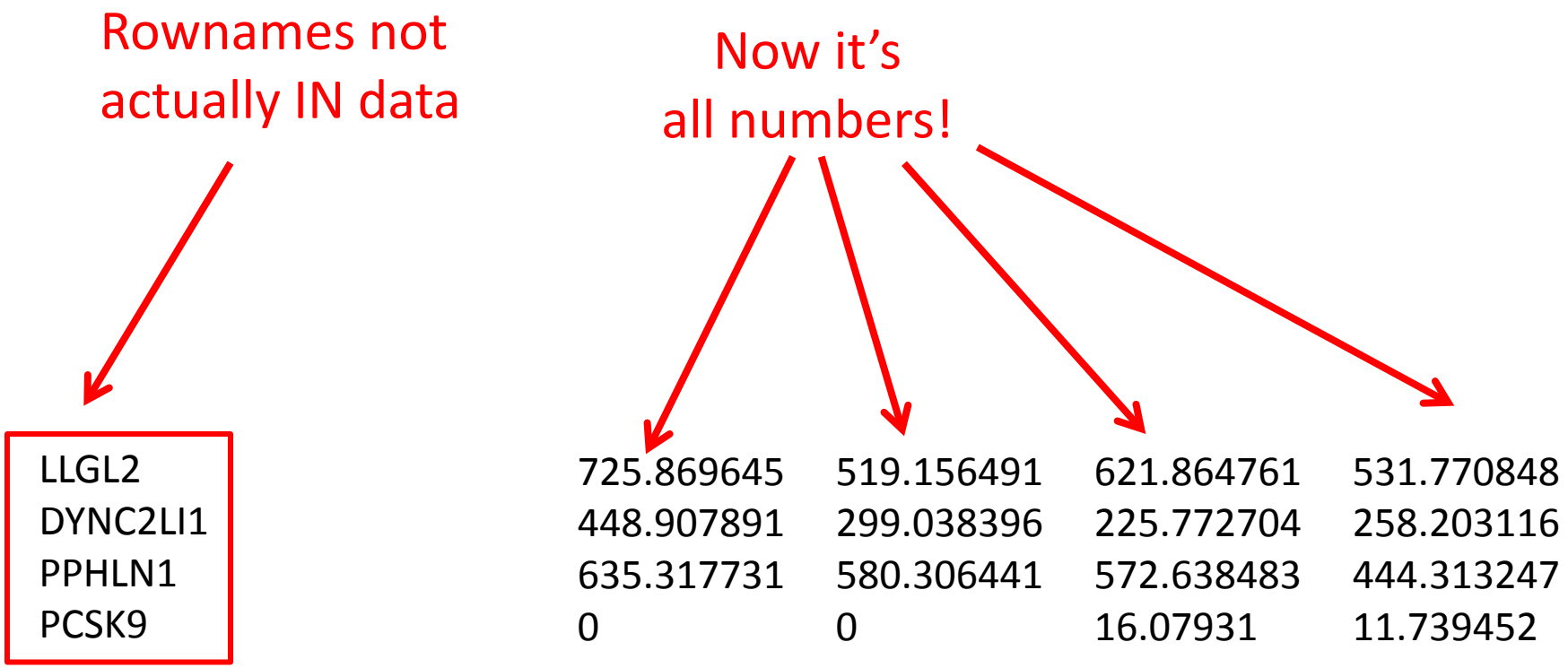
```
>mydata = mydata[,-1]
```

Get rid of
Leftmost column,
Leave rows alone

LLGL2	LLGL2	725.869645	519.156491	621.864761	531.770848
DYNC2LI1	DYNC2LI1	448.907891	299.038396	225.772704	258.203116
PPHLN1	PPHLN1	635.317731	580.306441	572.638483	444.313247
PCSK9	PCSK9	0	0	16.07931	11.739452

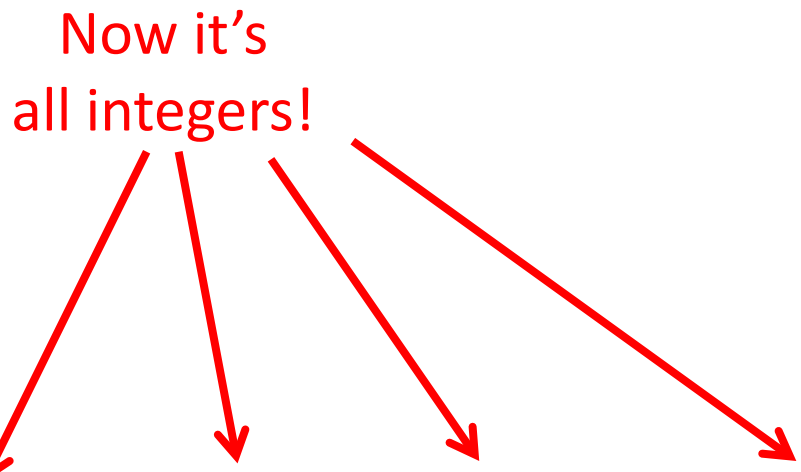
Run DESeq in R

>rownames(mydata) = mydata[,1]



Round the counts

>mydata = round(mydata)



LLGL2	726	519	622	532
DYNC2LI1	449	299	226	258
PPHLN1	635	580	572	444
PCSK9	0	0	16	11

Get rid of the empties

```
>mydata = mydata[rowSums(mydata) > 0,]
```

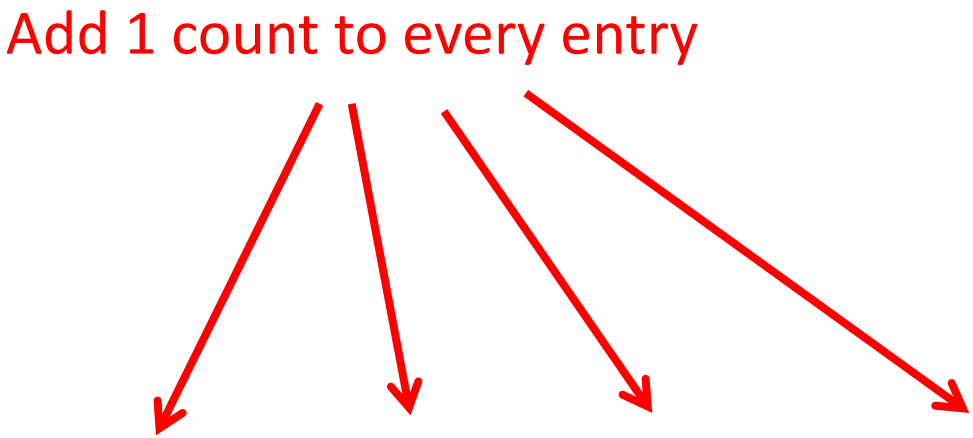
Keep all rows
Whose sum across row > 0

note that we
ignore column

LLGL2	726	519	622	532
DYNC2LI1	449	299	226	258
PPHLN1	635	580	572	444
PCSK9	0	0	16	11

Run DESeq in R

>mydata = mydata+1

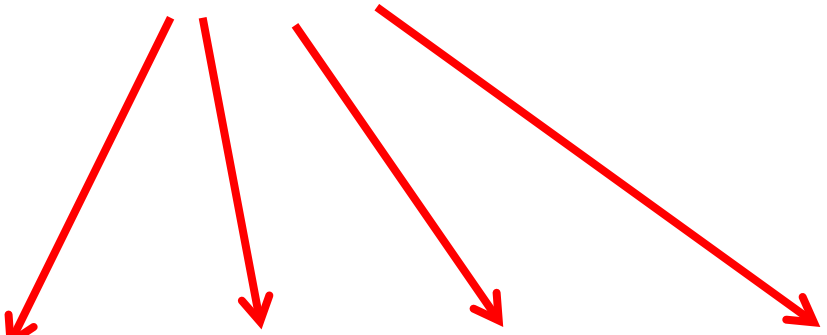


LLGL2	727	520	623	533
DYNC2LI1	450	300	227	259
PPHLN1	636	581	573	445
PCSK9	1	1	17	12

Run DESeq in R

>mydata = mydata+1

Now rows with a few zeros
Don't have any!



LLGL2	727	520	623	533
DYNC2LI1	450	300	227	259
PPHLN1	636	581	573	445
PCSK9	1	1	17	12

Won't have *infinity* fold-change

Run DESeq in R

```
>library(DESeq)
```

Tell DESeq you want to make new dataset

Name you pick

Data from

```
>cds = newCountDataset(mydata,c
("cond1","cond1","cond2","cond2"))
```

LLGL2	727	520	623	533
DYNC2LI1	450	300	227	259
PPHLN1	636	581	573	445
PCSK9	1	1	17	12

Run DESeq in R

```
>library(DESeq)
```

Tell DESeq you want to make new dataset

Name you pick

Data from

```
>cds = newCountDataset(mydata,c
("cond1","cond1","cond2","cond2"))
```

LLGL2	727	520	623	533
DYNC2LI1	450	300	227	259
PPHLN1	636	581	573	445
PCSK9	1	1	17	12

Run DESeq in R

```
>cds = estimateSizeFactors(cds)
```

Normalize libraries to one size

```
>cds = estimateDispersion(cds)
```

Estimate variance between “replicates”

Run DESeq in R

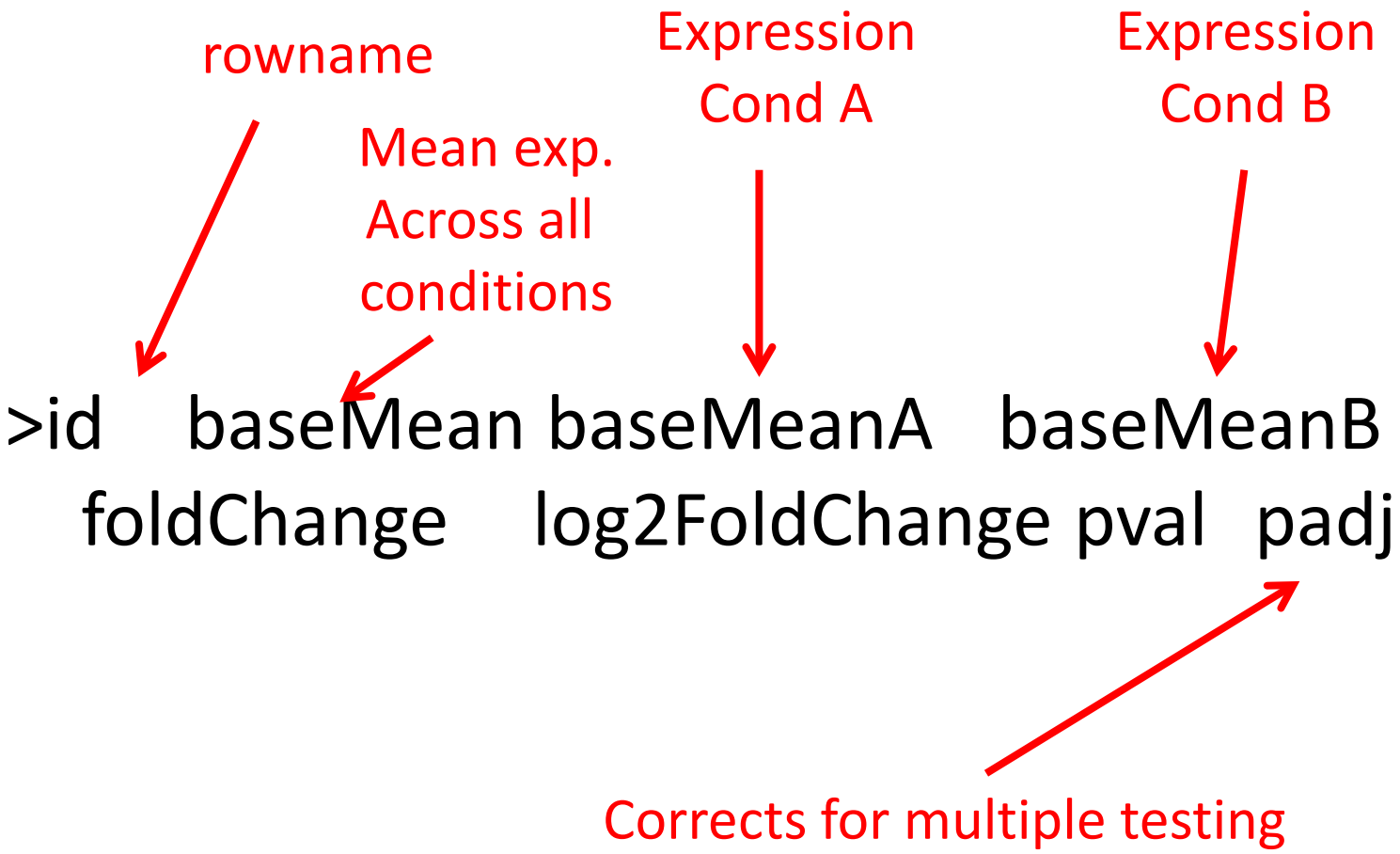
```
>de = nbinomTest(cds,"cond1","cond2")
```

Name you pick

Now normalized and variance estimated

Running differential expression test! (negative binomial distribution)

DESeq output



Run DESeq in R

```
>desig = de[de$padj<0.05,]
```

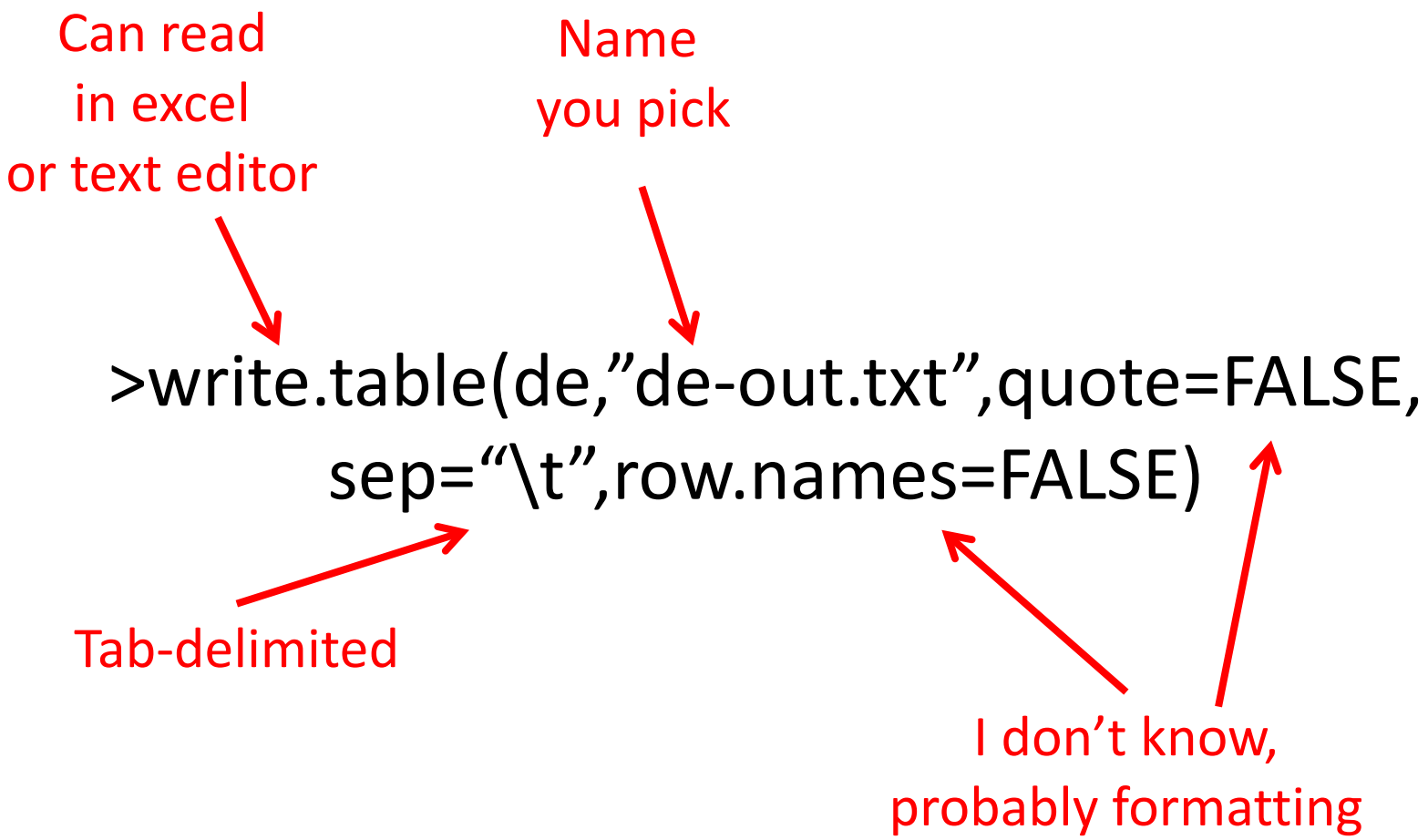
Name you pick

Changes across all genes (you picked this name)

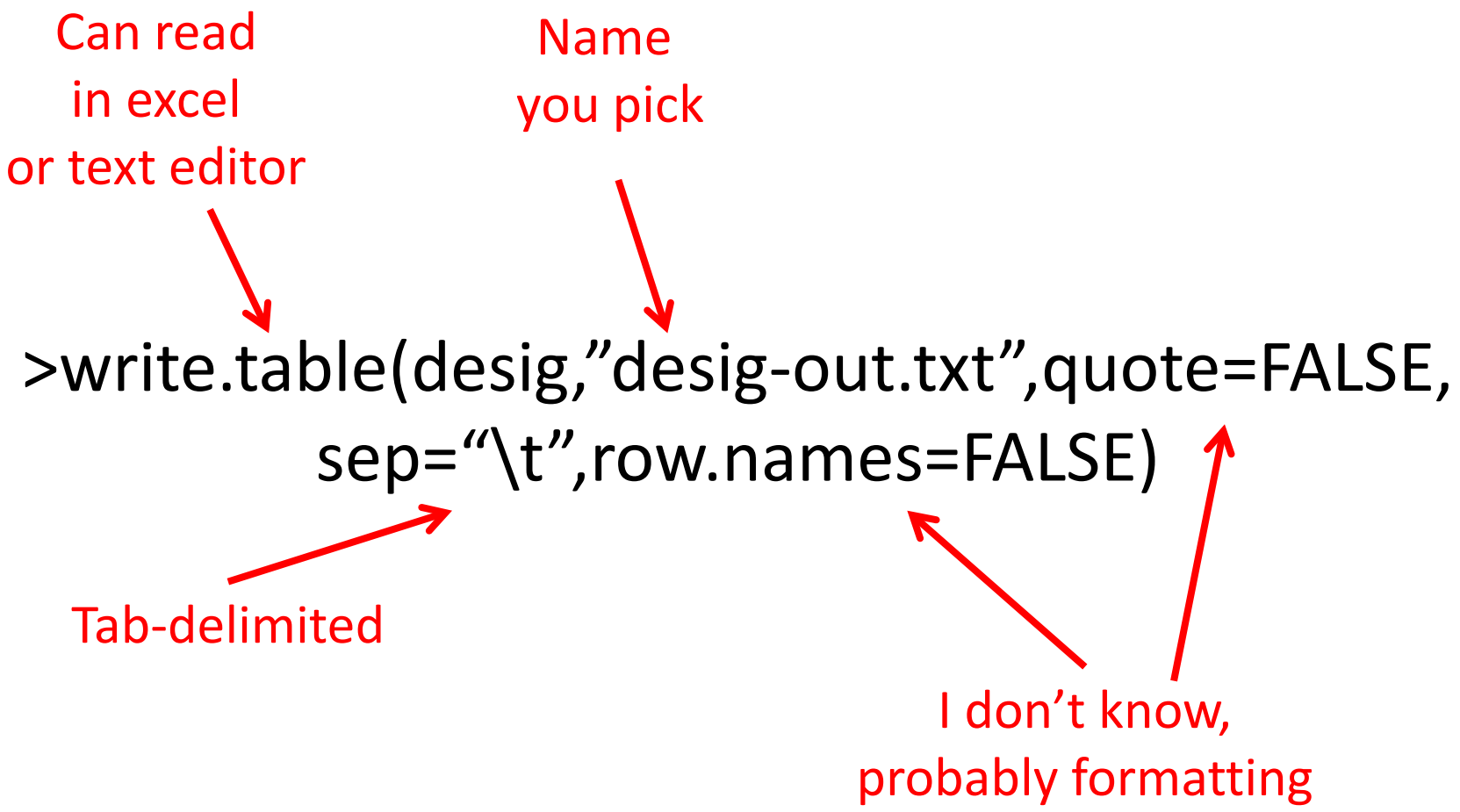
… but only those in column “padj” with value < 0.05

Getting significant hits

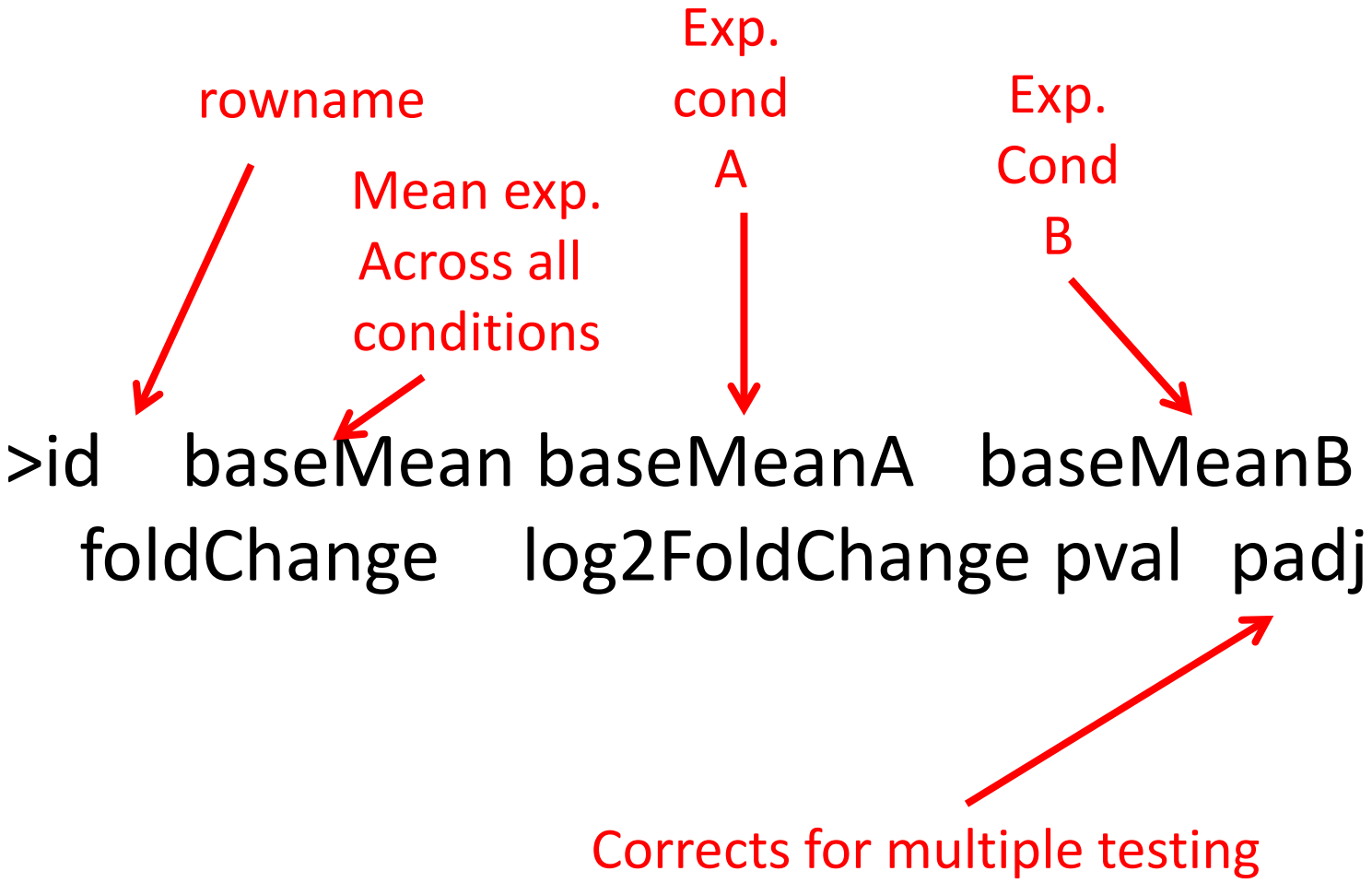
Getting table of genes



Getting significant table of genes



DESeq output



Make pretty pictures
("MA" plot)

X axis is
baseMean column
From "de"

Y axis is
log2FC column
From "de"

```
>plot(de$baseMean,de$log2FoldChange,
```

Dot size

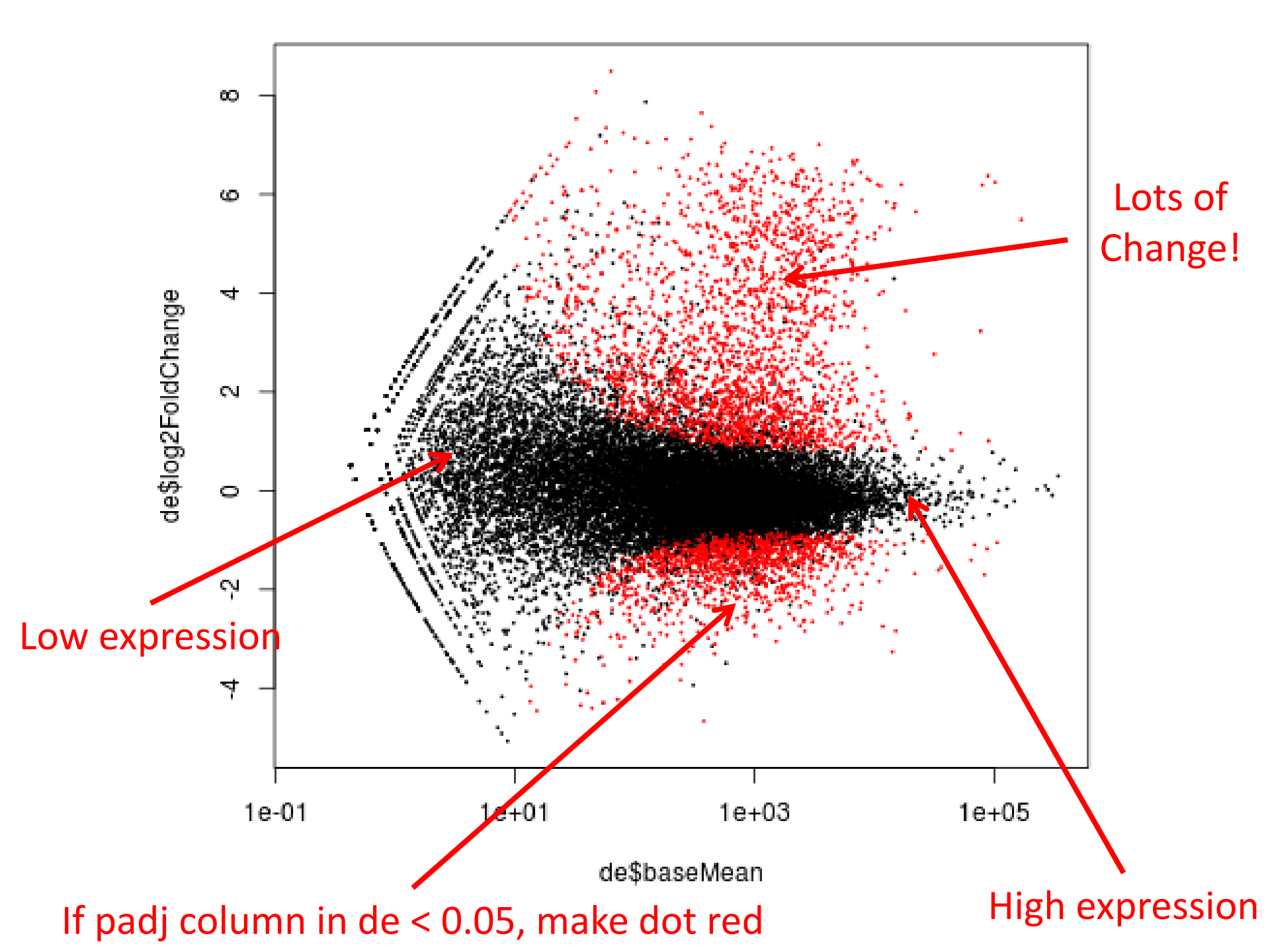
Dot shape

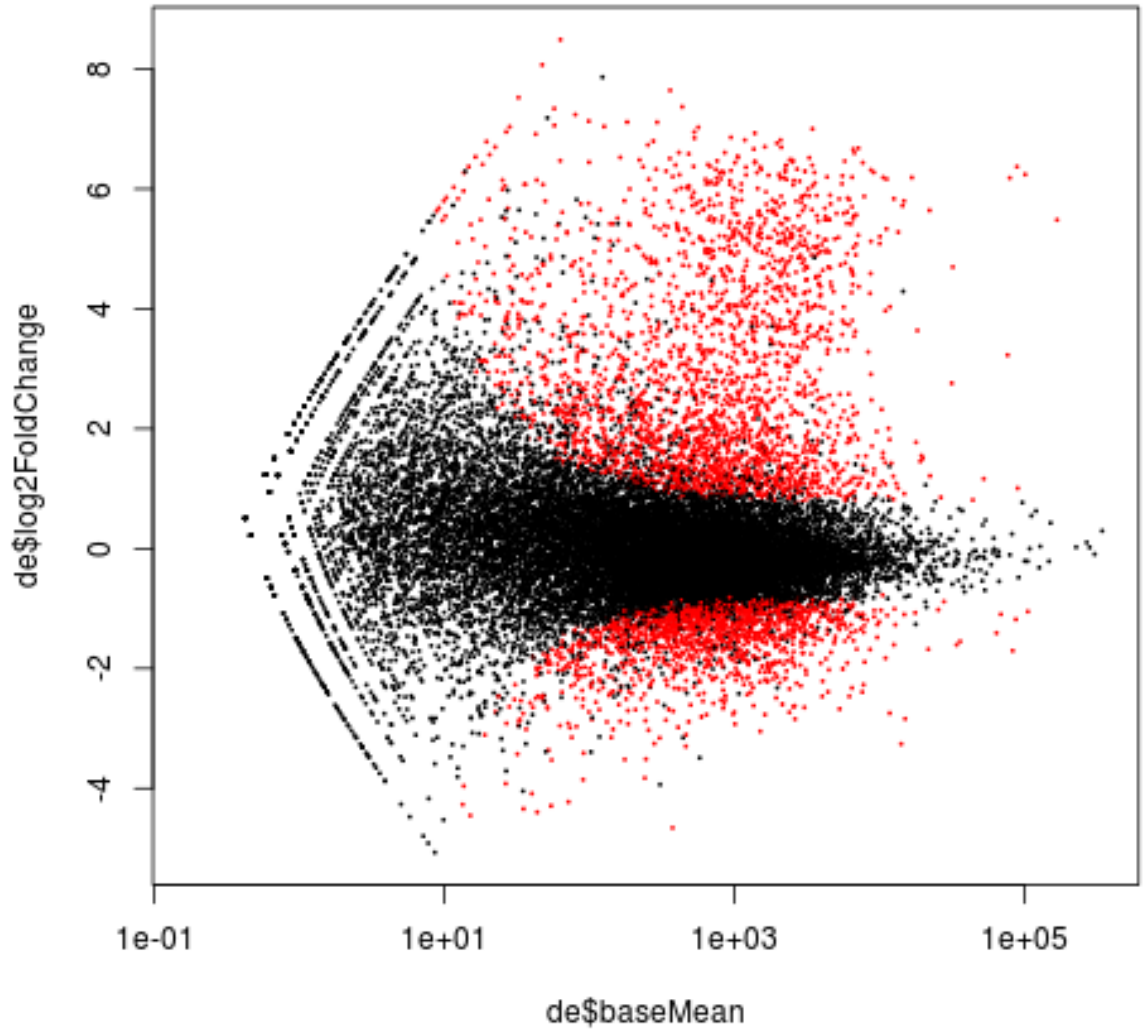
```
cex=0.3,pch=20,
```

```
col=ifelse(de$padj < 0.05,"red","black"),log="x")
```

If padj column in de < 0.05, make dot red

Make X axis log


de$log2FoldChange
8
6
4
2
0
-2
-4
1e-01
1e+01
1e+03
1e+05
de$baseMean
Lots of Change!
Low expression
If padj column in de < 0.05, make dot red
High expression



Point-and-click:

```
> identify(de$baseMean,de$log2FoldChange,cex=0.5,pch=20,labels=row.names(myd))
```