

brk-DESeq2

Upload packages

```
library(DESeq2)
library(apeglm)
library(dplyr)
library(tibble)
library(data.table)
library(ggplot2)
library(ggfortify)
library(ggrepel)
library(RColorBrewer)
library(pheatmap)
library(plotly)
library(magrittr)
library(AnnotationDbi)
library(org.Dm.eg.db)
```

Load data

```
counts <- fread(file = "rsem.merged.gene_counts.tsv", header=TRUE)
sample_info <- read.delim(file = "Data info/RN23010-sampleinfo_full.txt")
```

Prepare a matrix for DESeq:

Make first column row names, switch to integers, create a matrix object

```
info<-sample_info %>% column_to_rownames("Sample")
gcounts <- counts[, -2]
```

```
dt <- as.data.frame(gcounts) %>%
  column_to_rownames("gene_id")

dt[,1:9] <- lapply(dt[,1:9], as.integer)

mcountdata <- as.matrix(dt)

has_rownames(dt)
```

```
[1] TRUE
```

```
has_rownames(info)
```

```
[1] TRUE
```

Matching column order between the tables

```
all(rownames(info) == colnames(mcountdata))
```

```
[1] FALSE
```

```
mcountdata_s <- mcountdata[, rownames(info)]
all(rownames(info) == colnames(mcountdata_s))
```

```
[1] TRUE
```

Creating levels for samples.

It will read the first level to be compared with the rest.

```
info$Group <- factor(info$Group, levels = c("control", "sal-brk", "ubi-brk"))
info$Repeat <- as.factor(info$Repeat)
```

Run `deseq.`

The object class used by the DESeq2 package to store the read counts and the intermediate estimated quantities during statistical analysis is the **DESeqDataSet**, which will usually be represented in the code here as an object **dds**.

```
dds <- DESeqDataSetFromMatrix(countData = mcountdata_s,  
                              colData = info,  
                              design = ~Group)
```

Note: levels of factors in the design contain characters other than letters, numbers, '_' and '.'. It is recommended (but not required) to use only letters, numbers, and delimiters '_' or '.', as these are safe characters for column names in R. [This is a message, not a warning or an error]

```
dds <- DESeq(dds)
```

estimating size factors

Note: levels of factors in the design contain characters other than letters, numbers, '_' and '.'. It is recommended (but not required) to use only letters, numbers, and delimiters '_' or '.', as these are safe characters for column names in R. [This is a message, not a warning or an error]

estimating dispersions

gene-wise dispersion estimates

mean-dispersion relationship

Note: levels of factors in the design contain characters other than letters, numbers, '_' and '.'. It is recommended (but not required) to use only letters, numbers, and delimiters '_' or '.', as these are safe characters for column names in R. [This is a message, not a warning or an error]

final dispersion estimates

fitting model and testing

```
res <- results(dds)
sum(res$padj < 0.01, na.rm=TRUE)
```

```
[1] 1173
```

```
resultsNames(dds)
```

```
[1] "Intercept"                "Group_sal.brk_vs_control"
[3] "Group_ubi.brk_vs_control"
```

```
res %>%
  as.data.frame() %>%
  filter(padj<0.01) ->dres
```

```
dres$gene = mapIds(
  org.Dm.eg.db,
  keys = rownames(dres),
  column = "SYMBOL",
  keytype = "FLYBASE",
  multiVals = "first"
)
```

'select()' returned 1:1 mapping between keys and columns

Data shrinkage

```
ubiLFC <- lfcShrink(dds, coef="Group_ubi.brk_vs_control", type="apeglm")
```

using 'apeglm' for LFC shrinkage. If used in published research, please cite:
 Zhu, A., Ibrahim, J.G., Love, M.I. (2018) Heavy-tailed prior distributions for
 sequence count data: removing the noise and preserving large differences.
 Bioinformatics. <https://doi.org/10.1093/bioinformatics/bty895>

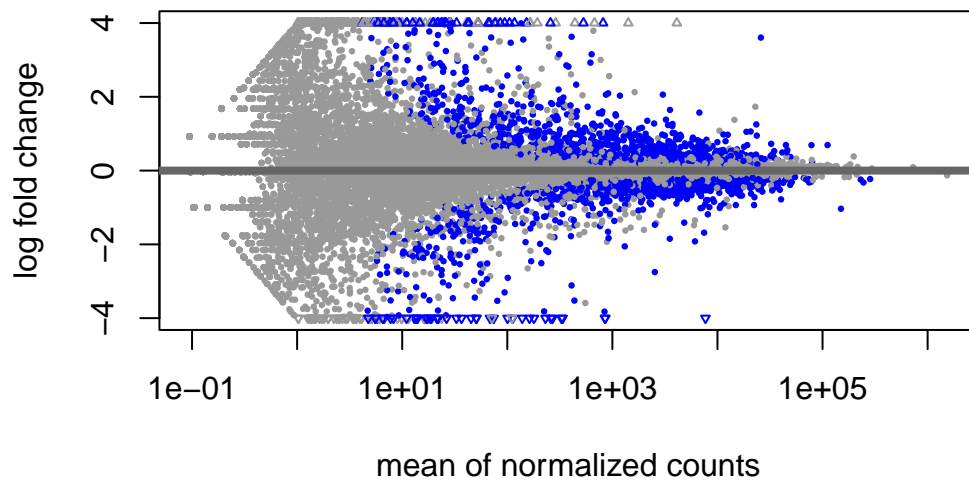
```
salLFC <- lfcShrink(dds, coef="Group_sal.brk_vs_control", type = "apeglm")
```

using 'apeglm' for LFC shrinkage. If used in published research, please cite:

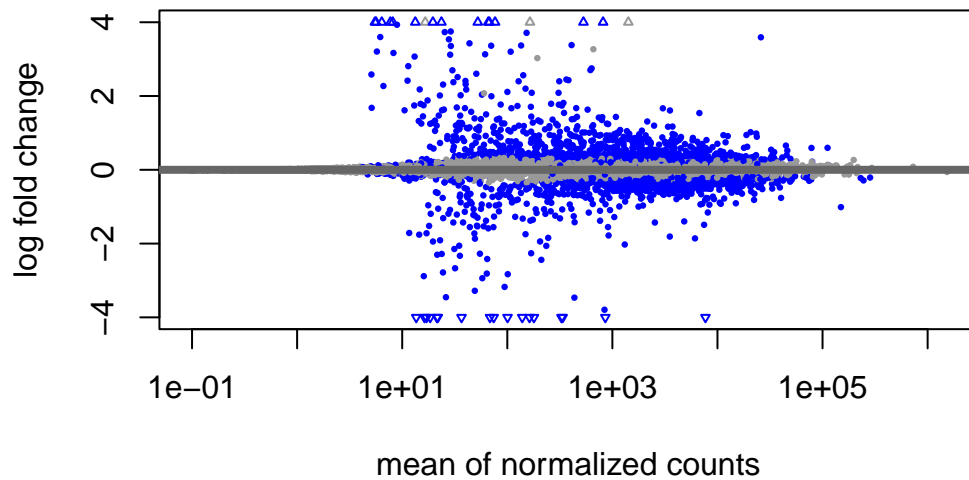
Zhu, A., Ibrahim, J.G., Love, M.I. (2018) Heavy-tailed prior distributions for sequence count data: removing the noise and preserving large differences. Bioinformatics. <https://doi.org/10.1093/bioinformatics/bty895>

Warning in nbinomGLM(x = x, Y = YNZ, size = size, weights = weightsNZ, offset = offsetNZ, : the line search routine failed, unable to sufficiently decrease the function value

```
plotMA(res, ylim=c(-4,4))
```



```
plotMA(ubiLFC, ylim=c(-4,4))
```



```
ubiLFC %>%
  as.data.frame() %>%
  filter(padj<0.01) %>%
  arrange(log2FoldChange) -> ubiLFC_s
```

Adding gene names to all data for ubiLFC

```
ubiLFC %>%
  as.data.frame() %>%
  arrange(log2FoldChange) -> ubiLFC_all
```

```
ubiLFC_all$gene = mapIds(
  org.Dm.eg.db,
  keys = rownames(ubiLFC_all),
  column = "SYMBOL",
  keytype = "FLYBASE",
  multiVals = "first"
)
```

'select()' returned 1:1 mapping between keys and columns

Adding gene names to all data for salLFC

```
salLFC %>%  
  as.data.frame() %>%  
  arrange(log2FoldChange) -> salLFC_all
```

```
salLFC_all$gene = mapIds(  
  org.Dm.eg.db,  
  keys = rownames(salLFC_all),  
  column = "SYMBOL",  
  keytype = "FLYBASE",  
  multiVals = "first"  
)
```

'select()' returned 1:1 mapping between keys and columns

```
salLFC_all %>%  
  filter(padj<0.1) %>%  
  arrange(log2FoldChange)
```

	baseMean	log2FoldChange	lfcSE	pvalue	padj
FBgn0013679	7170.627837	-7.254626e-01	0.22065818	3.443624e-05	2.402112e-02
FBgn0003996	841.515644	-7.124847e-01	0.24615945	1.196960e-04	4.630463e-02
FBgn0035480	1955.341925	-6.230745e-01	0.24592925	3.249310e-04	9.726134e-02
FBgn0020493	4826.397871	-5.691916e-01	0.09284659	3.984609e-11	8.323848e-08
FBgn0259932	378.675990	-4.049670e-01	0.15527259	2.744043e-04	9.544069e-02
FBgn0037796	8009.063987	-3.432772e-01	0.13456638	3.259116e-04	9.726134e-02
FBgn0261648	7877.685535	-2.836156e-01	0.09066954	6.354353e-05	3.318561e-02
FBgn0028789	825.587955	-4.592897e-03	0.01572004	3.449659e-05	2.402112e-02
FBti0019102	13.600512	-9.383137e-04	0.01470714	2.648346e-08	4.610328e-05
FBti0018937	7.758252	-4.982460e-04	0.01467519	9.567574e-06	8.327776e-03
FBgn0010295	11.359199	-2.820077e-04	0.01466677	1.133996e-04	4.555609e-02
FBti0102102	213.086870	-6.887191e-05	0.01466304	2.934387e-14	1.021656e-10
FBti0019471	52.406155	2.971626e-03	0.01510991	3.264419e-08	4.870979e-05
FBgn0262139	4409.851710	2.526059e-01	0.07964731	5.376690e-05	3.231513e-02
FBgn0033159	3790.127816	3.334542e-01	0.13053239	3.208938e-04	9.726134e-02
FBgn0263256	1446.974131	3.798371e-01	0.09865126	4.545280e-06	4.315950e-03
FBgn0003130	506.502226	4.580553e-01	0.17652278	2.891249e-04	9.544069e-02
FBgn0033232	2148.372696	5.514052e-01	0.18717627	1.068988e-04	4.466231e-02

FBgn0038610	6728.223984	5.955645e-01	0.19282199	8.081247e-05	3.550085e-02
FBgn0003137	20679.712869	6.274068e-01	0.15099963	1.283519e-06	1.489595e-03
FBgn0034045	197.417698	6.323162e-01	0.20828762	8.157209e-05	3.550085e-02
FBgn0037000	261.212738	6.883280e-01	0.21837135	5.693636e-05	3.231513e-02
FBgn0033520	269.466774	7.107480e-01	0.27123674	2.633113e-04	9.483748e-02
FBgn0261649	3533.114443	7.193516e-01	0.18531333	4.102217e-06	4.284765e-03
FBgn0265575	102.341037	7.254217e-01	0.22924983	5.519523e-05	3.231513e-02
FBgn0038180	581.146809	9.783684e-01	0.32135617	7.723785e-05	3.550085e-02
FBgn0263038	307.369467	1.028502e+00	0.39134462	2.558714e-04	9.483748e-02
FBgn0039078	342.075300	1.334324e+00	0.43880051	7.424787e-05	3.550085e-02
FBgn0039749	170.770248	1.720919e+00	0.22587494	1.418838e-15	7.409880e-12
FBgn0024250	25897.896274	2.220270e+00	0.16223169	5.778255e-44	6.035387e-40
FBti0102111	76.482349	3.444483e+00	1.11165860	5.878290e-05	3.231513e-02
FBgn0035300	67.468663	3.701320e+00	0.82898100	3.489683e-07	4.556217e-04
FBgn0029804	815.499625	4.967346e+00	0.81123836	3.641589e-11	8.323848e-08
FBti0019371	7.547635	6.583595e+00	3.24100147	2.923985e-04	9.544069e-02
FBti0059691	12.581896	6.835518e+00	2.80287473	2.570772e-05	2.065517e-02
	gene				
FBgn0013679	ND1				
FBgn0003996	w				
FBgn0035480	CG14984				
FBgn0020493	Dad				
FBgn0259932	CG42455				
FBgn0037796	CG12814				
FBgn0261648	salm				
FBgn0028789	Doc1				
FBti0019102	<NA>				
FBti0018937	<NA>				
FBgn0010295	ng3				
FBti0102102	<NA>				
FBti0019471	<NA>				
FBgn0262139	trh				
FBgn0033159	Dscam1				
FBgn0263256	CG43394				
FBgn0003130	Poxn				
FBgn0033232	CG12159				
FBgn0038610	CG7675				
FBgn0003137	Ppn				
FBgn0034045	CG8249				
FBgn0037000	ZnT77C				
FBgn0033520	Prx2540-1				
FBgn0261649	tinc				
FBgn0265575	yin				

FBgn0038180	Cht5
FBgn0263038	CG43333
FBgn0039078	CG4374
FBgn0039749	CG11498
FBgn0024250	brk
FBti0102111	<NA>
FBgn0035300	CG1139
FBgn0029804	CG3097
FBti0019371	<NA>
FBti0059691	<NA>

Adding gene names to ubiLFC with padj<0.01

```
ubiLFC_s$gene = mapIds(
  org.Dm.eg.db,
  keys = rownames(ubiLFC_s),
  column = "SYMBOL",
  keytype = "FLYBASE",
  multiVals = "first"
)
```

'select()' returned 1:1 mapping between keys and columns

```
ubiLFC_s %>%
  filter(baseMean>100, log2FoldChange<(-0.2)) -> ubi_down001
fwrite(ubi_down001, file = "./ubi_down001.xlsx")
```

```
ubiLFC_s %>%
  filter(baseMean>100, abs(log2FoldChange)>1 & abs(log2FoldChange)<2)
```

	baseMean	log2FoldChange	lfcSE	pvalue	padj
FBgn0026061	205.1062	-1.942218	0.21228910	3.789008e-21	2.395099e-18
FBti0019522	6108.3701	-1.859857	0.26311392	9.835620e-14	2.457990e-11
FBgn0287423	245.0981	-1.845286	0.64761762	1.388194e-04	1.944792e-03
FBgn0037181	3518.5431	-1.809787	0.18658344	1.556041e-23	1.194372e-20
FBgn0259226	920.0422	-1.780230	0.21742795	1.420713e-17	5.654436e-15
FBgn0036690	149.1412	-1.746162	0.43876937	3.068703e-06	7.832845e-05
FBgn0040279	908.6699	-1.552488	0.29997127	1.113078e-08	6.657712e-07
FBti0019633	7461.7578	-1.490685	0.41428382	1.387849e-05	2.829947e-04

FBgn0037127	2536.6073	-1.429363	0.16806968	1.017091e-18	4.752025e-16
FBgn0033483	436.9711	-1.421129	0.21717652	2.975889e-12	5.076016e-10
FBgn0020493	4826.3979	-1.399497	0.09148941	4.901729e-54	1.316849e-50
FBgn0038405	763.3332	-1.381673	0.30867054	3.587209e-07	1.227648e-05
FBgn0027586	165.8752	-1.378629	0.39381928	1.959630e-05	3.773868e-04
FBgn0031745	175.0911	-1.353802	0.24931587	3.198580e-09	2.291462e-07
FBgn0038761	140.1666	-1.324333	0.27534123	7.627962e-08	3.488089e-06
FBgn0032803	317.5330	-1.316588	0.25555054	1.405364e-08	8.297823e-07
FBgn0045800	1042.6826	-1.281596	0.24126480	5.549525e-09	3.704049e-07
FBgn0053542	118.8783	-1.274766	0.36158310	2.028071e-05	3.884785e-04
FBgn0039297	428.6658	-1.259147	0.23715010	5.867840e-09	3.892334e-07
FBgn0031630	286.0658	-1.243588	0.26949519	2.055745e-07	7.889655e-06
FBgn0085354	230.5502	-1.193532	0.17754977	1.048157e-12	1.976051e-10
FBgn0034861	190.0041	-1.171022	0.28688154	2.522658e-06	6.676966e-05
FBgn0036821	595.9706	-1.160352	0.17532931	2.116037e-12	3.727693e-10
FBgn0000179	8091.0312	-1.157927	0.15082046	1.773501e-15	5.955639e-13
FBgn0035379	258.9265	-1.125517	0.16627459	7.586843e-13	1.482331e-10
FBgn0283741	762.6886	-1.111945	0.19016228	2.964396e-10	2.895945e-08
FBgn0004956	949.7247	-1.092814	0.21949899	3.567285e-08	1.888377e-06
FBti0019154	333.0366	-1.074841	0.25622167	1.420585e-06	4.103658e-05
FBgn0262563	186.1847	-1.071498	0.26334350	2.621720e-06	6.854744e-05
FBgn0002868	223.6053	-1.067770	0.35146273	1.074136e-04	1.561930e-03
FBgn0036589	407.2687	-1.056655	0.43742491	6.020702e-04	6.203112e-03
FBgn0001233	149617.9827	-1.011731	0.11090185	4.785855e-21	2.857156e-18
FBgn0030183	603.7885	1.003480	0.17276517	3.967499e-10	3.739890e-08
FBgn0004512	1114.2821	1.006214	0.35978110	2.359167e-04	2.972433e-03
FBgn0262353	517.3457	1.010078	0.20540256	5.174442e-08	2.504710e-06
FBgn0260660	547.3397	1.012120	0.30425099	4.490106e-05	7.676662e-04
FBgn0003137	20679.7129	1.019412	0.14520305	1.203113e-13	2.873034e-11
FBgn0037487	316.6203	1.030165	0.21955257	1.607457e-07	6.643745e-06
FBgn0032402	168.3651	1.034581	0.44404971	7.886897e-04	7.805136e-03
FBgn0033574	663.4265	1.035453	0.23107999	4.148413e-07	1.397456e-05
FBgn0027843	107.2237	1.039181	0.37054373	2.316345e-04	2.924964e-03
FBgn0034253	1375.2643	1.047778	0.23424161	4.492989e-07	1.499430e-05
FBgn0003285	1909.3386	1.061409	0.24431937	7.520919e-07	2.384065e-05
FBgn0265575	102.3410	1.074793	0.21413154	3.209246e-08	1.732993e-06
FBgn0002633	1185.9644	1.079947	0.19673432	2.391704e-09	1.822784e-07
FBgn0267041	228.4824	1.097837	0.25256393	7.681349e-07	2.413561e-05
FBgn0052843	250.9056	1.132434	0.29575335	6.674907e-06	1.513261e-04
FBgn0000108	117.0334	1.134353	0.32016530	1.943672e-05	3.763368e-04
FBgn0039938	641.2609	1.164764	0.24100876	7.360743e-08	3.394787e-06
FBgn0000180	3384.4735	1.167253	0.21012053	1.591738e-09	1.305711e-07
FBgn0003326	3734.1292	1.172314	0.25523486	2.810433e-07	1.006697e-05

FBgn0032022	464.7526	1.180003	0.16892179	1.773026e-13	4.053816e-11
FBgn0015011	251.4749	1.182992	0.21789944	3.164646e-09	2.282368e-07
FBgn0261526	143.6292	1.186881	0.36948094	6.031426e-05	9.746422e-04
FBgn0267029	128.5551	1.196769	0.40065901	1.181379e-04	1.697206e-03
FBgn0031678	2494.5680	1.197039	0.22598535	6.103990e-09	3.999602e-07
FBgn0263380	302.0143	1.201263	0.35654538	3.431470e-05	6.084914e-04
FBgn0000022	655.9389	1.208257	0.20977526	4.713626e-10	4.329284e-08
FBgn0037126	291.4756	1.248545	0.29828177	1.447571e-06	4.159249e-05
FBgn0004170	887.4590	1.251567	0.23651670	6.500282e-09	4.182756e-07
FBgn0003499	174.9473	1.278754	0.44117997	1.528842e-04	2.090196e-03
FBgn0033232	2148.3727	1.309289	0.17087259	1.075884e-15	3.729501e-13
FBgn0003089	196.6421	1.338278	0.61911598	9.430356e-04	8.950271e-03
FBgn0004003	794.1917	1.349053	0.40179848	3.350211e-05	5.983006e-04
FBgn0033520	269.4668	1.360697	0.24375408	1.271335e-09	1.050905e-07
FBgn0027844	411.2421	1.370269	0.22632413	7.640040e-11	8.463904e-09
FBgn0002632	430.5362	1.378386	0.33232980	1.584434e-06	4.516268e-05
FBgn0063127	742.4218	1.416048	0.33952984	1.439318e-06	4.146625e-05
FBgn0032283	801.5836	1.426283	0.38897704	1.090335e-05	2.275096e-04
FBgn0025525	1080.2904	1.433208	0.20870379	3.615181e-13	7.617399e-11
FBgn0284244	460.3946	1.435434	0.38782749	9.482128e-06	2.036241e-04
FBti0019364	169.1228	1.441761	0.59595576	5.090547e-04	5.448508e-03
FBgn0038610	6728.2240	1.538489	0.17723248	3.143271e-19	1.535345e-16
FBgn0032405	346.5463	1.542115	0.38876445	3.189513e-06	8.102720e-05
FBgn0261649	3533.1144	1.607768	0.17561147	2.940838e-21	1.975141e-18
FBgn0038180	581.1468	1.630686	0.29954334	2.594611e-09	1.909705e-07
FBti0019523	3021.7647	1.666853	0.66409373	3.753684e-04	4.384467e-03
FBgn0033518	109.3588	1.669167	0.45549589	1.064750e-05	2.226031e-04
FBgn0035522	167.6700	1.706850	0.55684909	8.082457e-05	1.246113e-03
FBgn0014469	124.9687	1.730263	0.24501846	8.572126e-14	2.302902e-11
FBgn0285970	330.4999	1.740045	0.35508045	4.428318e-08	2.223678e-06
FBgn0033405	309.4204	1.765642	0.17537091	4.135903e-25	3.703701e-22
FBgn0037000	261.2127	1.789185	0.19794165	9.478692e-21	5.360949e-18
FBgn0039078	342.0753	1.811680	0.41582428	5.832618e-07	1.910894e-05
	gene				
FBgn0026061	Mipp1				
FBti0019522	<NA>				
FBgn0287423	Nplp2				
FBgn0037181	CG11370				
FBgn0259226	CG42326				
FBgn0036690	Ilp8				
FBgn0040279	Osi14				
FBti0019633	<NA>				
FBgn0037127	CG14566				

FBgn0033483	egr
FBgn0020493	Dad
FBgn0038405	CG8927
FBgn0027586	CG5867
FBgn0031745	rau
FBgn0038761	CG17190
FBgn0032803	CG13082
FBgn0045800	Uhg1
FBgn0053542	upd3
FBgn0039297	CG11852
FBgn0031630	CG15629
FBgn0085354	CG34325
FBgn0034861	CG9815
FBgn0036821	CG3961
FBgn0000179	bi
FBgn0035379	spz5
FBgn0283741	prage
FBgn0004956	upd1
FBti0019154	<NA>
FBgn0262563	CG43103
FBgn0002868	MtnA
FBgn0036589	CG13067
FBgn0001233	Hsp83
FBgn0030183	Ypel
FBgn0004512	Mdr49
FBgn0262353	CG43051
FBgn0260660	Mp
FBgn0003137	Ppn
FBgn0037487	thw
FBgn0032402	PLCXD
FBgn0033574	Spn47C
FBgn0027843	CAH2
FBgn0034253	CG10936
FBgn0003285	rst
FBgn0265575	yin
FBgn0002633	E(spl)m7-HLH
FBgn0267041	asRNA:CR45485
FBgn0052843	Dh31-R
FBgn0000108	Appl
FBgn0039938	Sox102F
FBgn0000180	bib
FBgn0003326	sca
FBgn0032022	CG14275

FBgn0015011	AhcyL2
FBgn0261526	NT1
FBgn0267029	lncRNA:CR45473
FBgn0031678	Nep13
FBgn0263380	lncRNA:CR43432
FBgn0000022	ac
FBgn0037126	CG14567
FBgn0004170	sc
FBgn0003499	sr
FBgn0033232	CG12159
FBgn0003089	pip
FBgn0004003	wbl
FBgn0033520	Prx2540-1
FBgn0027844	CAH1
FBgn0002632	E(spl)m6-BFM
FBgn0063127	lncRNA:CR33938
FBgn0032283	CG7296
FBgn0025525	bab2
FBgn0284244	l(2)k05911
FBti0019364	<NA>
FBgn0038610	CG7675
FBgn0032405	firl
FBgn0261649	tinc
FBgn0038180	Cht5
FBti0019523	<NA>
FBgn0033518	Prx2540-2
FBgn0035522	CG1273
FBgn0014469	Cyp4e2
FBgn0285970	CG32500
FBgn0033405	CG13954
FBgn0037000	ZnT77C
FBgn0039078	CG4374

Data transformation

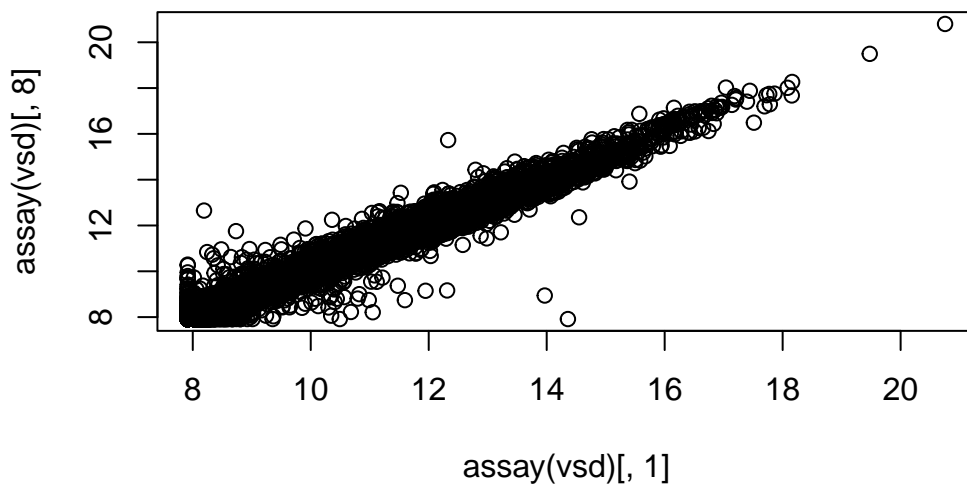
```
vsd <- vst(dds)
head(assay(vsd), 3)
```

	3h_Sal-Brk-dark_1	3h_Sal-Brk-light_1	3h_Ubi-Brk-light_1
FBgn0000003	8.390974	7.914333	8.277769
FBgn0000008	11.656992	11.696456	11.782049

FBgn0000014	8.260044	8.136115	7.914333
	3h_Sal-Brk-light_3	3h_Ubi-Brk-light_3	3h_Sal-Brk-dark_2
FBgn0000003	8.707625	8.511504	8.725605
FBgn0000008	11.737912	11.662655	11.803065
FBgn0000014	8.064303	8.000241	8.135127
	3h_Sal-Brk-light_2	3h_Ubi-Brk-light_2	3h_Sal-Brk-dark_3
FBgn0000003	8.513593	8.584759	8.806269
FBgn0000008	11.661077	11.850823	11.749082
FBgn0000014	8.106995	8.063072	7.914333

QC

```
plot(
  assay(vsd)[,1],
  assay(vsd)[,8]
)
```

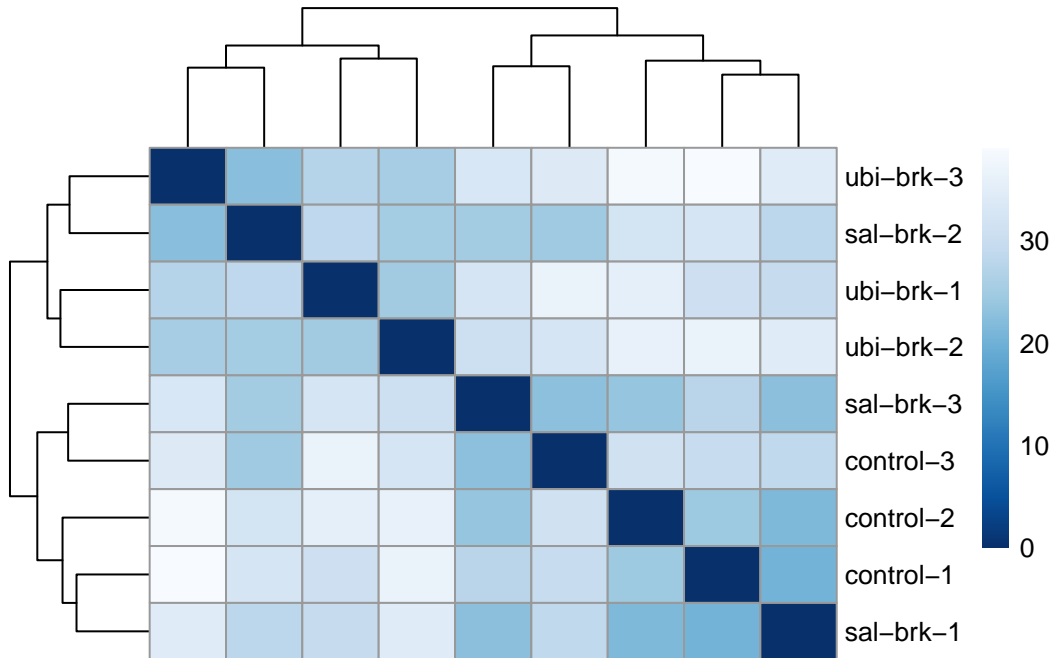


```
sampleDists <- dist(t(assay(vsd)))
sampleDistMatrix <- as.matrix(sampleDists)
rownames(sampleDistMatrix) <- paste(vsd$Group, vsd$Repeat, sep="-")
```

```

colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Blues"))) (255)
pheatmap(sampleDistMatrix,
          clustering_distance_rows=sampleDists,
          clustering_distance_cols=sampleDists,
          col=colors)

```

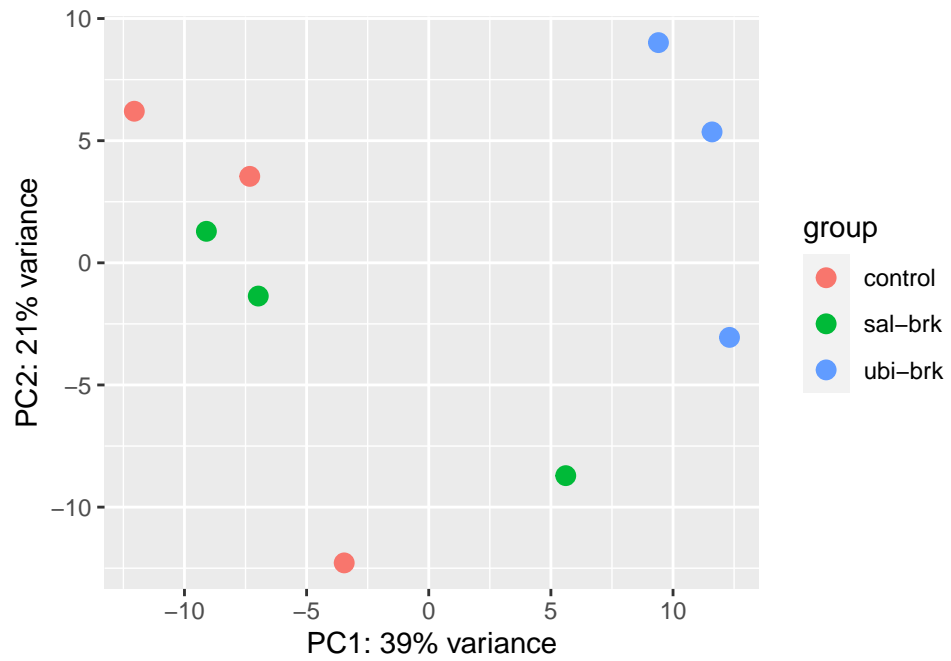


```

plotPCA(vsd, intgroup="Group")

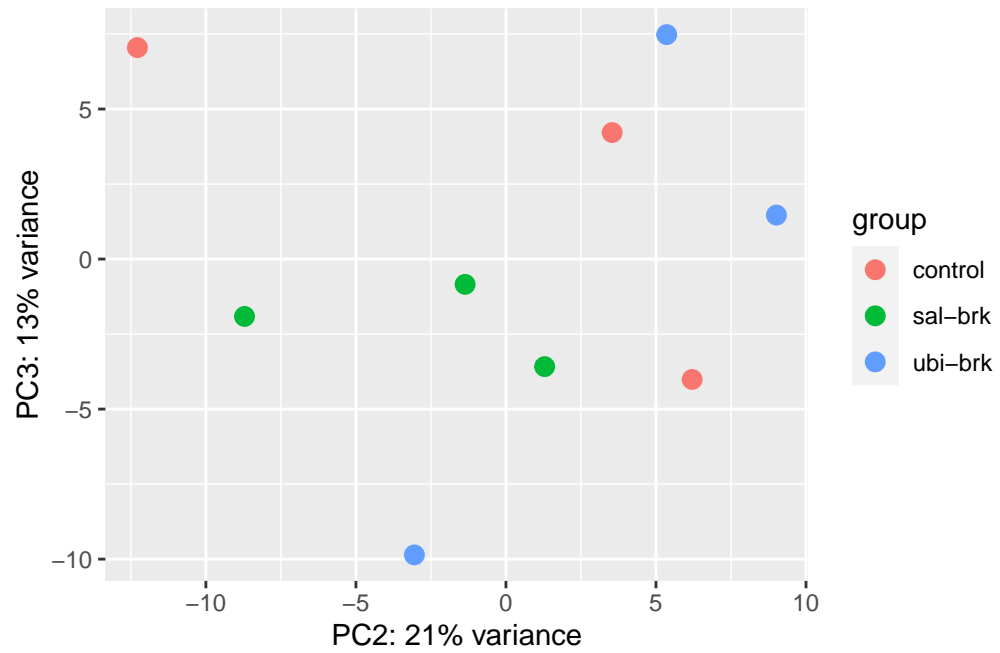
```

using ntop=500 top features by variance



```
plotPCA(vsd, intgroup="Group", pcsToUse=2:3)
```

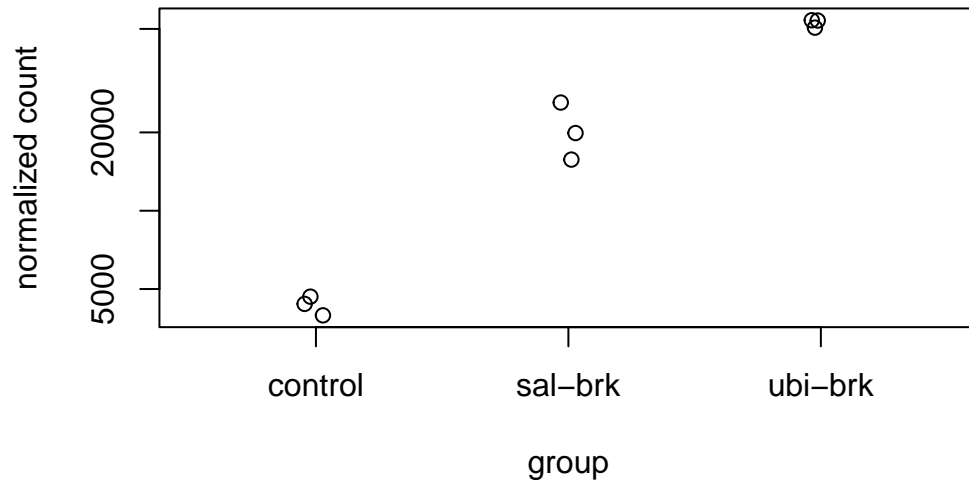
using ntop=500 top features by variance



Normalized counts for some genes

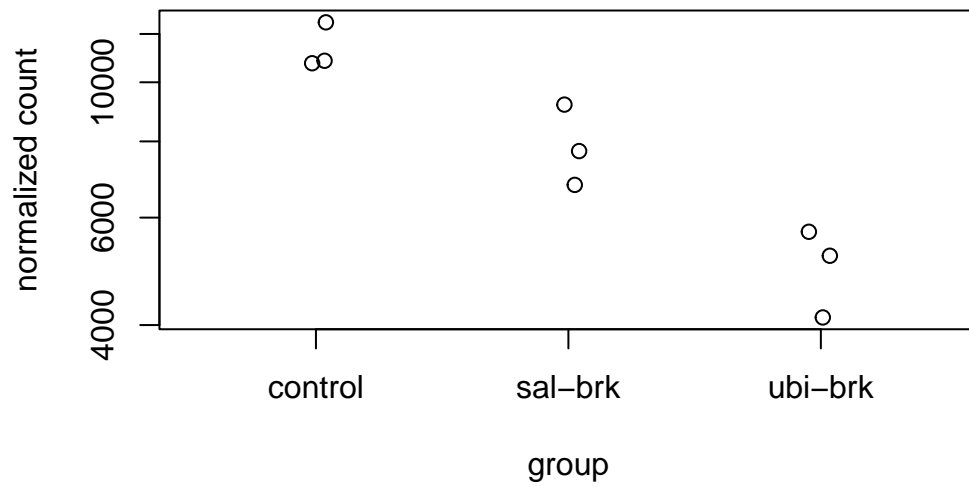
```
plotCounts(dds, gene="FBgn0024250", intgroup="Group") #Brk
```

FBgn0024250

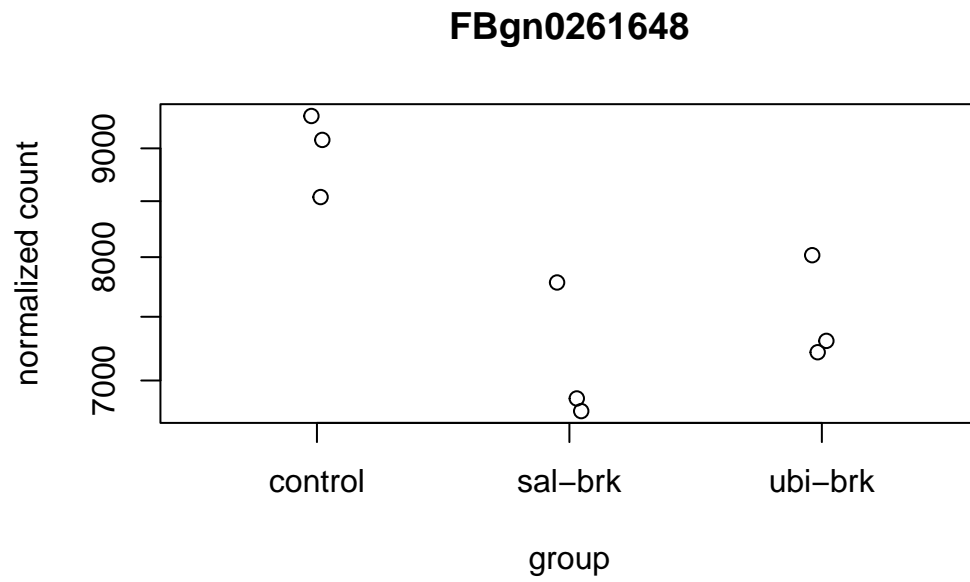


```
plotCounts(dds, gene="FBgn0000179", intgroup="Group") #omb norm counts
```

FBgn0000179

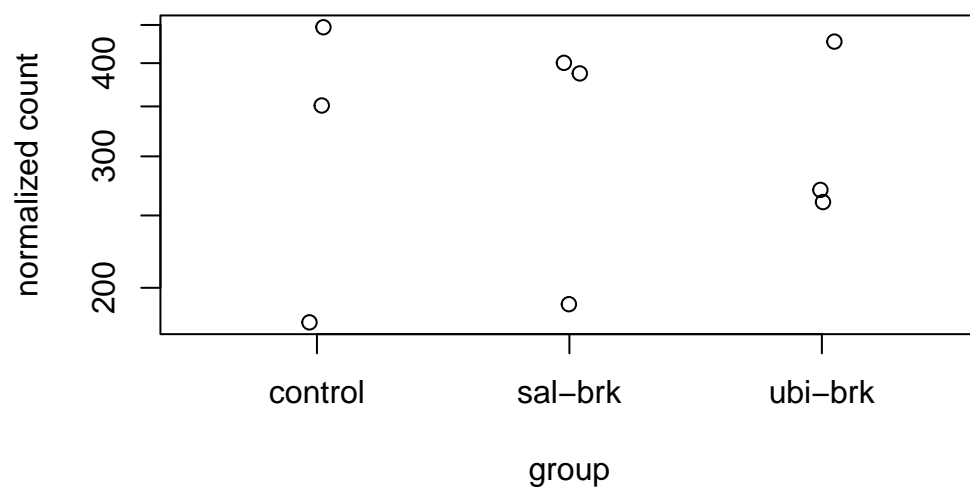


```
plotCounts(dds, gene="FBgn0261648", intgroup="Group") #Sal
```



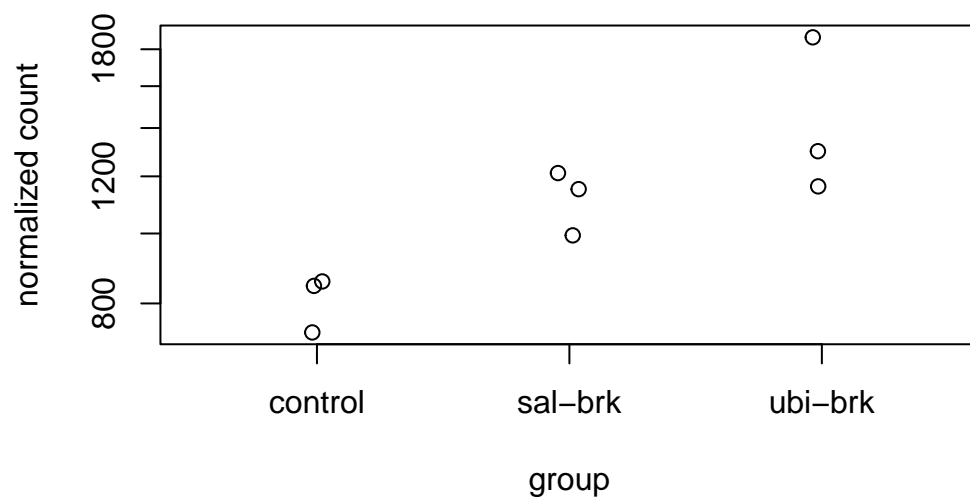
```
plotCounts(dds, gene="FBgn0025360", intgroup="Group") #optix
```

FBgn0025360

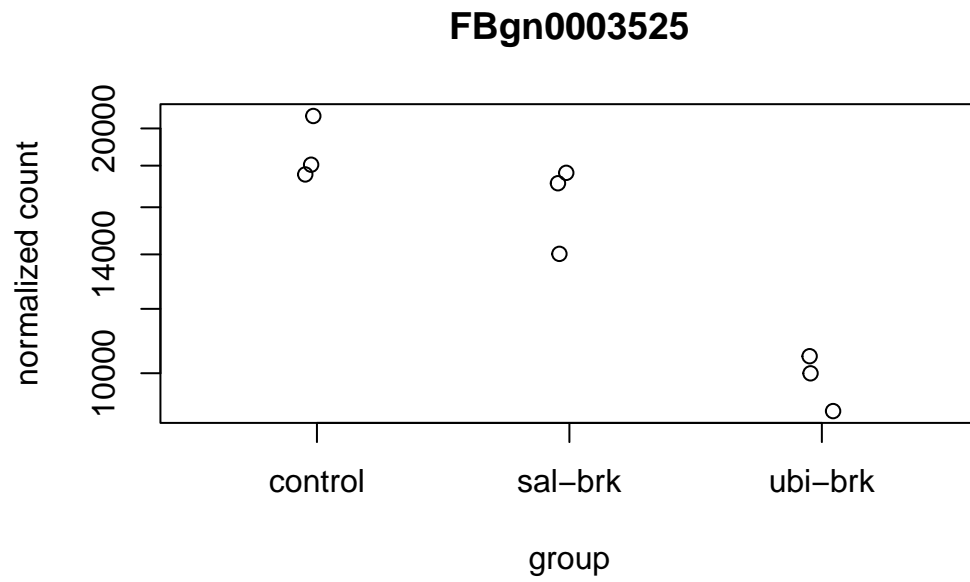


```
plotCounts(dds, gene="FBgn0011706", intgroup="Group") #reaper
```

FBgn0011706

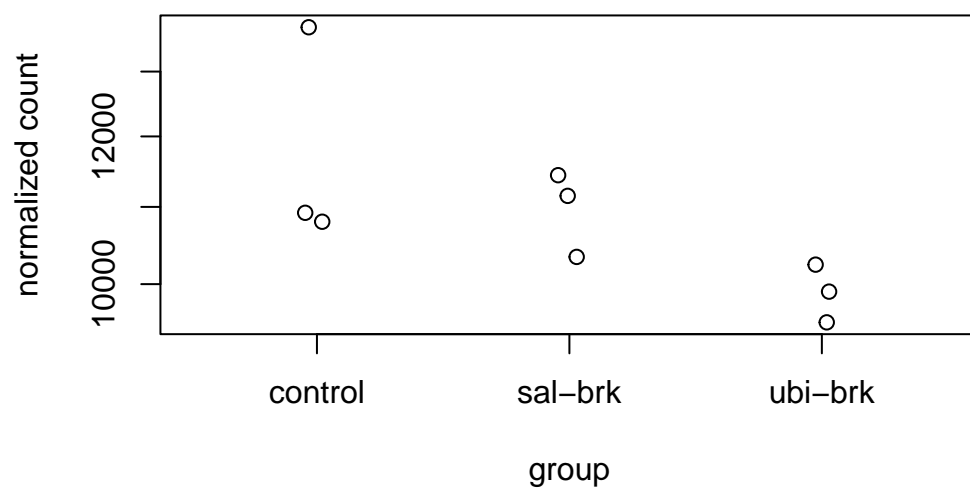


```
plotCounts(dds, gene="FBgn0003525", intgroup="Group") #string
```



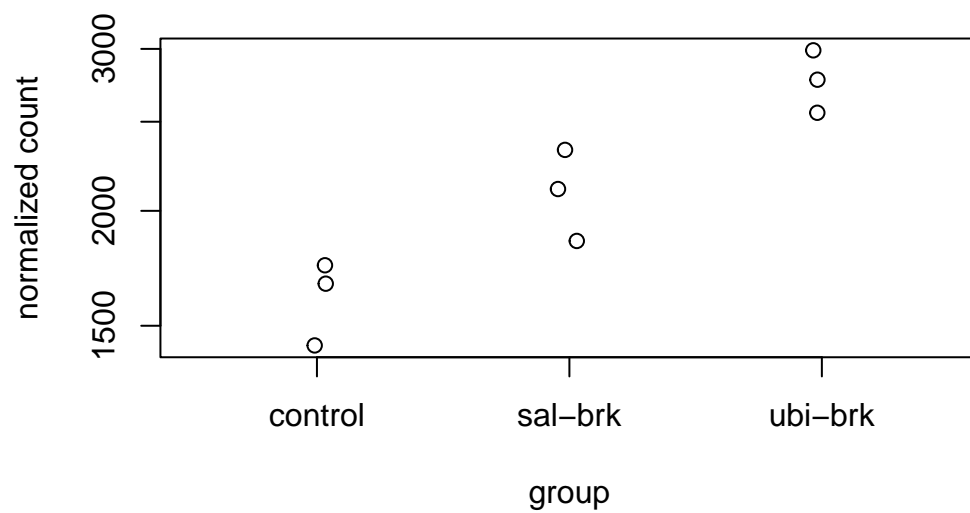
```
plotCounts(dds, gene="FBgn0262656", intgroup="Group") #myc
```

FBgn0262656

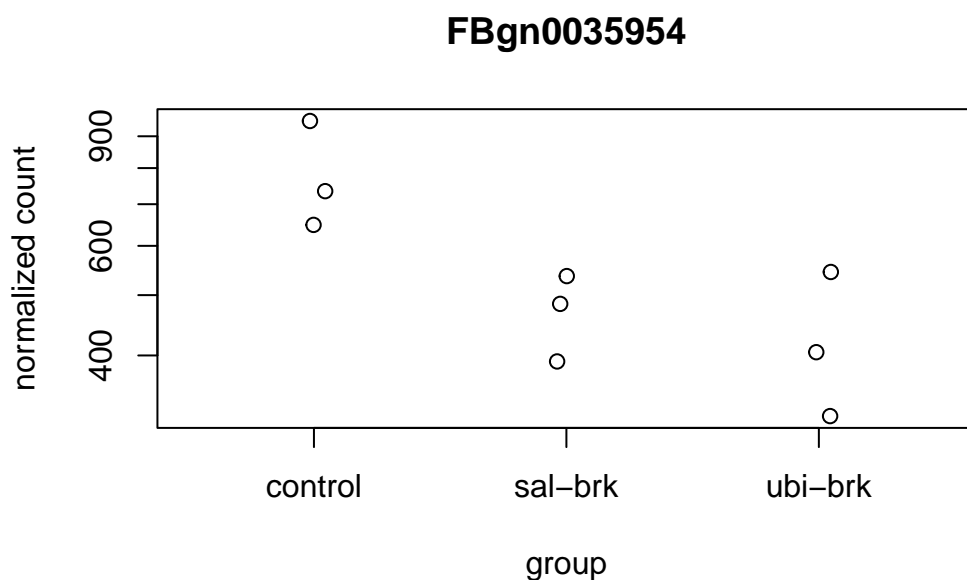


```
plotCounts(dds, gene="FBgn0004394", intgroup="Group") #pdm2
```

FBgn0004394



```
plotCounts(dds, gene="FBgn0035954", intgroup="Group") #doc3
```



Checking cell cycle genes

Upload the table with cell cycle genes first

```
cycle <- fread(file ="cellcycle_genes.csv", header=TRUE)

ubi_cycle<-subset(ubiLFC_all, gene %in% cycle$gene)

merge(ubi_cycle, cycle, by="gene") %>%
  column_to_rownames("gene") %>%
  filter(padj<0.05, log2FoldChange<(-0.3)) %>%
  arrange(log2FoldChange) ->ubi_cycle_005

as.matrix(ubi_cycle_005[,-c(3:5,7)]) -> ubi_cycle_m
```

```
#heatmap(ubi_cycle_m)
```

Yifans data to compare

```
yifan <-read.delim(file = "yifan_bulk_RNAseq_96_VS_104_FlyBase_Fields.txt", header=TRUE)
```