

```

#EdgeR analysis

# 1 reading input

library("edgeR")

x <- read.delim("bodyraw.txt",row.names="ENSEMBL_ID")

group <- factor(c(1,1,1,1,2,2,2,2)) #for BRGC, 3 and 4 were concatenated

y <- DGEList(counts=x,group=group)

design <- model.matrix(~group)

#2. Filtering out non-expressed genes

keep <- rowSums(cpm(y)>1) >= 1
y <- y[keep, , keep.lib.sizes=FALSE]
table(keep)

#3. Normalization

y <- calcNormFactors(y)
y$samples

#4. exporting normalized values with annotations

data.cds <- calcNormFactors(y, method='TMM')
data.norm <- cpm(data.cds, prior.count=2, log=TRUE)

library("AnnotationDbi")

library("org.Dm.eg.db")

res <- as.data.frame(data.norm)

res$symbol <- mapIds(org.Dm.eg.db,
                    keys=row.names(res),
                    column="SYMBOL",
                    keytype="ENSEMBL",
                    multiVals="first")

res$entrez <- mapIds(org.Dm.eg.db,
                    keys=row.names(res),
                    column="ENTREZID",
                    keytype="ENSEMBL",
                    multiVals="first")

res$name = mapIds(org.Dm.eg.db,
                  keys=row.names(res),
                  column="GENENAME",

```



```
p1 <- p1 + geom_point( aes(plot_X, plot_Y, size = plot_size), shape = 21, fill = "transparent", c
olour = I (alpha ("black", 0.6) )) + scale_size_area();
p1 <- p1 + scale_size( range=c(5, 30)) + theme_bw(); # + scale_fill_gradientn(colours = heat_hcl(
7), limits = c(-300, 0) );
ex <- one.data [ one.data$dispensability < 0.15, ];
p1 <- p1 + labs (y = "semantic space x", x = "semantic space y");
p1 <- p1 + theme(legend.key = element_blank() );
one.x_range = max(one.data$plot_X) - min(one.data$plot_X);
one.y_range = max(one.data$plot_Y) - min(one.data$plot_Y);
p1 <- p1 + xlim(min(one.data$plot_X)-one.x_range/10,max(one.data$plot_X)+one.x_range/10);
p1 <- p1 + ylim(min(one.data$plot_Y)-one.y_range/10,max(one.data$plot_Y)+one.y_range/10);
p1 <- p1 + theme(
  axis.line = element_line(size = 1.5),
  axis.text = element_text(size = rel(3)),
  legend.text = element_text(size= rel(1)),
  axis.title = element_text(size = rel(3)))
```