cat `ll \*.Reads\* | sed 's/ \*/ /g' | cut -d' ' -f 9 | head -1` | tail -n+5 | cut -f1 > tmp.tsv

for x in \*.Reads\*; do paste tmp.tsv <(tail -n+5 $x | cut -f2) > tmp ; mv tmp tmp.tsv; done

cat <(for x in \*.Reads\*; do echo -en "\t${x%%.\*}"; done; echo "") tmp.tsv > counts.tsv; rm tmp.tsv

----------------------------------------------------------------------------------

suppressPackageStartupMessages(library(DESeq2))

suppressPackageStartupMessages(library(gplots))

suppressPackageStartupMessages(library(ggplot2))

suppressPackageStartupMessages(library(RColorBrewer))

suppressPackageStartupMessages(library(biomaRt))

suppressPackageStartupMessages(library(sva))

suppressPackageStartupMessages(source('~/scripts/make.html.report.R'))

hmcol <- colorRampPalette(brewer.pal(9,"GnBu"))(100)

data <- read.table("counts.tsv", row.names=1, header=T, sep="\t")

colnames(data) <- gsub("X", "", colnames(data))

colnames(data) <- gsub('[.]', "-", colnames(data))

col <- unique(grepl("[\_]",colnames(data)))

if(length(col) == 1 & col) {col <- as.data.frame(t(sapply((strsplit(colnames(data),"[\_]")),unlist)))} else {col <- as.data.frame(colnames(data))}

rownames(col) <- colnames(data); col

#col$V3 <- factor(paste0(col$V1, '\_', col$V3))

col$V5 <- factor(paste0(col$V3, '\_', col$V5))

col$V6 <- factor(paste0(col$V1,"\_",col$V2,"\_",col$V4))

col[,1:4] <- NULL

colnames(col) <- intgroup <- c('type','rep')

dds <- DESeqDataSetFromMatrix(countData=data, colData=col, design=~type)

dds <- DESeq(dds); rld <- rlog(dds)

rlogMat <- assay(rld); distsRL <- dist(t(rlogMat)); mat <- as.matrix(distsRL)

hc <- hclust(distsRL)

if(length(colnames(col)) > 1) {rownames(mat) <- colnames(mat) <- do.call(paste,c(col[,intgroup]))} else rownames(mat) <- colnames(mat) <- paste(col[,"type"])

pca\_data <- plotPCA(rld,intgroup=intgroup, returnData=T)

perc <- round(100 \* attr(pca\_data, "percentVar"))

p <- ggplot(pca\_data, aes(PC1, PC2, color=paste(type))) + geom\_point(size=2) + geom\_text(aes(label=paste(rep)),size=3,colour="black",hjust='inward',vjust='inward') + xlab(paste0("PC1: ",perc[1],"% variance")) + ylab(paste0("PC2: ",perc[2],"% variance")); p$labels$colour <- ""

plot(p)

pdf("sample2sample.pdf",title="sample2sample")

heatmap.2(mat, Rowv=as.dendrogram(hc), symm=T, trace="none", col = rev(hmcol), margin=c(13,13), cexRow=0.8, cexCol=0.8)

dev.off()

pdf("PCA.pdf",title="PCA", height=10,width=10); p; dev.off()

write.table(counts(dds, normalized=T), sep="\t", file="counts\_norm.tsv", quote=F, col.names=NA)

### get results ###

contrasts <- 'type'

conditions <- character()

for(x in contrasts){conditions <- c(conditions,as.vector(unique(col[,x])))}

res <- list()

for(z in contrasts){for(x in conditions){for(y in conditions){

if(x==y) next; res[[paste0(y,"\_vs\_",x)]] <- results(dds,contrast=c(z,y,x))}}}

names(res)

### make tsv ###

mart <- useMart("ensembl",dataset="hsapiens\_gene\_ensembl")

for(x in names(res)){

res\_tmp <- add.anns(as.data.frame(res[[x]]),mart)

res\_tmp <- cbind(res\_tmp[,1:4],rownames(res\_tmp),res\_tmp[,5:11])

colnames(res\_tmp)[5] <- "EnsemblID"

colnames(res\_tmp)[12] <- "FDR"

write.table(res\_tmp, file=paste0(x,".tsv"), quote=F, sep="\t",row.names=F)}

### batch effect ###

batch <- col$batch

mod <- model.matrix(~type, data=col)

rldfilt <- subset(assay(rld), rowMeans(assay(rld)) > 0)

rldbatch <- ComBat(dat=rldfilt, batch=batch, mod=mod,par.prior=TRUE, prior.plots=FALSE); rv <- rowVars(rldbatch)

select <- order(rv, decreasing = TRUE)[seq\_len(min(500,length(rv)))]

pca <- prcomp(t(rldbatch[select, ])); percentVar <- pca$sdev^2/sum(pca$sdev^2)

intgroup.df <- as.data.frame(colData(rld)[, intgroup, drop = FALSE])

group <- factor(apply(intgroup.df, 1, paste, collapse = " : "))

pca\_data <- data.frame(PC1 = pca$x[, 1], PC2 = pca$x[, 2], group = group,intgroup.df, names = colnames(rld))

attr(pca\_data, "percentVar") <- percentVar[1:2]

distsRL <- dist(t(rldbatch)); mat <- as.matrix(distsRL); hc <- hclust(distsRL)

if(length(colnames(col)) > 1) {rownames(mat) <- colnames(mat) <- do.call(paste,c(col[,intgroup]))} else rownames(mat) <- colnames(mat) <- paste(col[,"type"])

#weiter mit perc <- round(...

### boxplot ###

boxdata <- data[rowSums(data)!=0,]

boxdata <- stack(as.data.frame(log2(boxdata+8)))

boxdata <- cbind(boxdata,col[boxdata[,2],])

colnames(boxdata) <- c('log2Counts','raw samples',colnames(col))

boxdatanorm <- counts(dds,normalized=T)

boxdatanorm <- boxdatanorm[rowSums(boxdatanorm)!=0,]

boxdatanorm <- stack(as.data.frame(log2(boxdatanorm+8)))

boxdatanorm <- cbind(boxdatanorm,col[boxdatanorm[,2],])

colnames(boxdatanorm) <- c('log2Counts','normalized samples',colnames(col))

p1 <- ggplot(boxdata) + geom\_boxplot(aes(x = `raw samples`, y = log2Counts, fill = boxdata$type)) + theme(axis.text.x=element\_blank(), legend.position="none", axis.ticks.x = element\_blank())

p2 <- ggplot(boxdatanorm) + geom\_boxplot(aes(x = `normalized samples`, y = log2Counts, fill = boxdatanorm$type)) + theme(axis.title.y=element\_blank(), axis.text.y=element\_blank(), axis.ticks.y=element\_blank(), axis.text.x=element\_blank(), legend.title=element\_blank(), axis.ticks.x = element\_blank())

pA <- ggplotGrob(p1); pB <- ggplotGrob(p2)

grid.arrange(cbind(pA, pB, size = "last"))

remove(boxdata, boxdatanorm, p1, p2, pA, pB); invisible(gc())

### html-Report ###

suppressPackageStartupMessages(library(ReportingTools))

suppressPackageStartupMessages(library(plotrix))

suppressPackageStartupMessages(library(scales))

suppressPackageStartupMessages(library(Rgraphviz))

suppressPackageStartupMessages(library(biomaRt))

suppressPackageStartupMessages(library(gridExtra))

suppressPackageStartupMessages(library(pathview))

suppressPackageStartupMessages(library(RamiGO))

suppressPackageStartupMessages(library(BioNet))

suppressPackageStartupMessages(library(RpsiXML))

suppressPackageStartupMessages(library(rgl))

suppressPackageStartupMessages(library(DLBCL))

suppressPackageStartupMessages(library(pheatmap))

suppressPackageStartupMessages(library(gage)) ###

suppressPackageStartupMessages(library(npsp)) ###

suppressPackageStartupMessages(library(reshape2)) ###

suppressPackageStartupMessages(library(igraph)) ###

interactome <- as.matrix(read.table("/home/mspohn/skripte/htmlreport/intact/fly/fly",sep="\t",quote="",row.names=1,header=T))

colnames(interactome) <- rownames(interactome)

interactome <- as(interactome,"graphNEL")

kegg <- kegg.gsets(species="hsa")$kg.sets

suppressMessages(go <- go.gsets(species = "human"))

mart <- useMart("ensembl",dataset="hsapiens\_gene\_ensembl")

fdr <- 0.1

lfc <- 1

### report per result ###

x <- names(res[])

res\_tmp <- add.anns(as.data.frame(res[[x]]),mart)

res\_tmp <- cbind(res\_tmp[,1:4],rownames(res\_tmp),res\_tmp[,5:11])

colnames(res\_tmp)[5] <- "EnsemblID"

colnames(res\_tmp)[12] <- "FDR"

### plot counts ###

for(y in which(res\_tmp$FDR < fdr & res\_tmp$log2FoldChange >= lfc | res\_tmp$FDR < fdr & res\_tmp$log2FoldChange <= -lfc)) {

tmp <- plotCounts(dds, gene=y, intgroup="type",returnData = T)

tmp <- tmp[tmp$type %in% unlist(strsplit(x, '\_vs\_')),]

suppressMessages(ggplot(tmp, aes(x=type, y=count, fill = type)) + scale\_x\_discrete(name = "") + ggtitle(rownames(res\_tmp[y,])) + geom\_boxplot(width = 0.2) + ylab("normalized counts") + theme(legend.position="none",

axis.line.x = element\_line(color="black", size = 0.25),

axis.line.y = element\_line(color="black", size = 0.25)) +

ggsave(filename = paste0(rownames(res\_tmp[y,]),"\_counts.png")))

}

remove(tmp); invisible(gc())

### GAGE ###

res\_tmp <- subset(res\_tmp, EntrezID != 'NA')

foldchanges <- res\_tmp$log2FoldChange

names(foldchanges) <- res\_tmp$EntrezID

foldchanges\_cnet <- subset(res\_tmp, log2FoldChange != 'NA')$log2FoldChange

names(foldchanges\_cnet) <- subset(res\_tmp, log2FoldChange != 'NA')$GeneSymbol

### KEGG ###

keggres <- gage(foldchanges, gsets=kegg, same.dir=F)

keggres <- subset(keggres$greater,keggres$greater[,4] != "NA" & keggres$greater[,4] < fdr)

enrichMap(keggres,kegg,fontsize=1)

par(new=T)

splot(slim = range(keggres[,3]),legend.width=0.25,legend.shrink=0.2,col=color\_scale("red","#E5C494"),horizontal=F,legend.lab="P-Value",legend.mar = 6)

par(new=F)

gc <- get\_cnet\_kegg(keggres,res\_tmp,kegg,lfc,fdr)

cnetplot(gc,foldChange=foldchanges\_cnet,fontsize=1)

par(new=T)

splot(slim = range(foldchanges\_cnet),legend.width=0.5,legend.shrink=0.25,col=smoothColors("royalblue2",50,"white",50,"red"),horizontal=T,legend.lab="log2FoldChange",legend.mar = 4.1)

par(new=F)

### GO ###

gores <- gage(foldchanges, gsets=go$go.sets, same.dir=T)

gos\_up <- subset(gores$greater,gores$greater[,4] != "NA" & gores$greater[,4] < 0.01)

gos\_down <- subset(gores$less,gores$less[,4] != "NA" & gores$less[,4] < 0.01)

gc <- get\_cnet\_go(gos\_up,res\_tmp,go,lfc,fdr)

cnetplot(gc,foldChange=foldchanges\_cnet)

par(new=T)

splot(slim = range(foldchanges\_cnet),legend.width=0.5,legend.shrink=0.25,col=smoothColors("royalblue2",50,"white",50,"red"),horizontal=T,legend.lab="log2FoldChange",legend.mar = 4.1)

par(new=F)

gc <- get\_cnet\_go(gos\_down,res\_tmp,go,lfc,fdr)

cnetplot(gc,foldChange=foldchanges\_cnet)

par(new=T)

splot(slim = range(foldchanges\_cnet),legend.width=0.5,legend.shrink=0.25,col=smoothColors("royalblue2",50,"white",50,"red"),horizontal=T,legend.lab="log2FoldChange",legend.mar = 4.1)

par(new=F)

### networks ###

res\_tmp <- subset(res\_tmp, GeneSymbol != "" & pvalue != "NA")

pvals <- cbind(res\_tmp$pvalue)

rownames(pvals) <- res\_tmp$GeneSymbol

pvals <- aggrPvals(pvals, order = ncol(pvals), plot = FALSE)

pvals <- pvals + .Machine$double.xmin

logFC <- res\_tmp$log2FoldChange

if(species == 'hsa') {

names(pvals) <- paste0(res\_tmp$GeneSymbol,'(',res\_tmp$EntrezID,')')

names(logFC) <- paste0(res\_tmp$GeneSymbol,'(',res\_tmp$EntrezID,')')

} else {

names(pvals) <- res\_tmp$GeneSymbol

names(logFC) <- res\_tmp$GeneSymbol

}

suppressWarnings(fb <- fitBumModel(pvals, plot = FALSE))

subnet <- subNetwork(names(pvals), interactome)

cc <- connectedComp(subnet)

for(x in names(cc)){if((sapply(cc,length) >= 10)[x] == F) cc[[x]] <- NULL}

for(y in names(cc)) {

network <- subGraph(cc[[y]], subnet)

suppressWarnings(network <- rmSelfLoops(network))

fdr\_ppi <- 0.1

scores <- scoreNodes(network, fb, fdr = fdr\_ppi)

while(length(subset(scores,scores > 0)) > 200) {

fdr\_ppi <- fdr\_ppi\*0.1

scores <- scoreNodes(network, fb, fdr = fdr\_ppi)

}

module <- runFastHeinz(network, scores)

plotModule(module, scores = scores, diff.expr = logFC)

par(new=T)

splot(slim = range(logFC),legend.width=0.5,legend.shrink=0.25,col=smoothColors("royalblue2",50,"white",50,"red"),horizontal=T,legend.lab='log2FoldChange',legend.mar = 4.1)

par(new=F)

}

### 3d plot ###

plot3dModule(module, diff.or.scores = logFC, red=c("positive"))

save3dModule("test.pdf")

dev.off()

### new intact ###

suppressPackageStartupMessages(library(RpsiXML))

intact <- psimi25XML2Graph(list.files('./',pattern='xml'), INTACT.PSIMI25, type="interaction",verbose=FALSE)

mat <- as(intact,"matrix")

mat <- mat[!(rownames(mat) == "NA"),]

mat <- mat[,!(colnames(mat) == "NA")]

write.table(mat,file="mouse\_raw",sep="\t",quote=F,col.names=T,row.names=T)

### donwload mart (Uniprot/Swissprot GeneName)

### ruby intact2mart mart\_export.txt mouse\_raw

### read intact ###

interactome <- as.matrix(read.table("mouse",sep="\t",quote="",row.names=1,header=T))

colnames(interactome) <- rownames(interactome)

interactome <- as(interactome,"graphNEL")

### msigdb ###

suppressPackageStartupMessages(labrary(GSEABase))

h <- geneIds(getGmt("~/scripts/htmlreport/msigdb/h.all.v5.1.entrez.gmt"))

c2 <- geneIds(getGmt("~/scripts/htmlreport/msigdb/c2.all.v5.1.entrez.gmt"))

c3 <- geneIds(getGmt("~/scripts/htmlreport/msigdb/c3.tft.v5.2.entrez.gmt"))

c7 <- geneIds(getGmt("~/scripts/htmlreport/msigdb/c7.all.v5.1.entrez.gmt"))

### stuff ###

### genen names of graphNEL ###

unlist(strsplit(nodes(module),"\\("))[seq(1,2\*length(nodes(module)),2)]

### biotype plots ###

library(NOISeq)

biotypes <- read.table("~/skripte/genes.csv",sep="\t",quote="",row.names=1)

mydata <- readData(data,factors=col,biotype=biotypes)

mybiodetection <- dat(mydata, k = 0, type = "biodetection", factor = NULL)

### one for each ###

i=1

for (x in rownames(col)){

pdf(paste0("biotypes\_",x,".pdf"), title=paste0("biotypes\_",x), pointsize=0.5)

par(oma=c(5,1,1,1))

explo.plot(mybiodetection, samples=c(i))

dev.off()

i=i+1

}

### one for all ###

pdf("biotypes.pdf", title="biotypes", pointsize=0.5)

#par(mfrow=c(2,2),oma=c(5,1,1,1),ps=5)

par(oma=c(5,1,1,1),ps=5)

for (i in 1:nrow(col)){explo.plot(mybiodetection, samples=c(i))}

dev.off()

counts <- as.matrix(read.table("counts\_lcl\_norm\_hiv.tsv"))

colnames(counts) <- gsub("X", "", colnames(counts))

heatmap.2(counts, trace = 'none', col = hmcol, dendrogram = 'col', Rowv = F, margins=c(15,10))

counts <- as.matrix(read.table("counts\_lcl\_norm\_hiv.tsv"))

counts[which(counts < 1)] <- 1

counts <- log(counts,10)

colnames(counts) <- gsub("X", "", colnames(counts))

heatmap.2(counts, trace = 'none', col = hmcol, dendrogram = 'col', Rowv = F, margins=c(15,10))

counts <- as.matrix(read.table("counts\_lcl\_norm\_ebv.tsv"))

colnames(counts) <- gsub("X", "", colnames(counts))

heatmap.2(counts, trace = 'none', col = hmcol, dendrogram = 'col', Rowv = F, margins=c(15,10), cexRow = 0.5, cexCol = 0.5)

counts <- as.matrix(read.table("counts\_lcl\_norm\_ebv.tsv"))

counts[which(counts < 1)] <- 1

counts <- log(counts,10)

colnames(counts) <- gsub("X", "", colnames(counts))

heatmap.2(counts, trace = 'none', col = hmcol, dendrogram = 'col', Rowv = F, margins=c(15,10), cexRow = 0.5, cexCol = 0.5)

### excel ###

wb <- createWorkbook()

addWorksheet(wb = wb, sheetName = "Upregulated")

addWorksheet(wb = wb, sheetName = "Downregulated")

addWorksheet(wb = wb, sheetName = "Unfiltered")

writeData(wb = wb, sheet = 1, x = subset(res\_tmp,log2FoldChange >= lfc & FDR < fdr))

setColWidths(wb, sheet = 1, cols = 1:12, widths = c(24,43.15,13.65,15.75,20.4,25.25,20.9,26.4,16.45,26.4,9.5,9.5))

addStyle(wb, sheet = 1, createStyle(border='Bottom',textDecoration='bold',halign = 'center'), rows = 1, cols = 1:12)

addStyle(wb, sheet = 1, createStyle(halign = 'center'), rows = 2:(nrow(subset(res\_tmp,log2FoldChange >= lfc & FDR < fdr)) + 1), cols = c(1,3:10), gridExpand = TRUE)

addStyle(wb, sheet = 1, createStyle(numFmt = 'SCIENTIFIC', halign = 'center'), rows = 2:(nrow(res\_tmp) + 1), cols = c(11:12), gridExpand = TRUE)

writeData(wb = wb, sheet = 2, x = subset(res\_tmp,log2FoldChange <= -lfc & FDR < fdr))

setColWidths(wb, sheet = 2, cols = 1:12, widths = c(24,43.15,13.65,15.75,20.4,25.25,20.9,26.4,16.45,26.4,9.5,9.5))

addStyle(wb, sheet = 2, createStyle(border='Bottom',textDecoration='bold',halign = 'center'), rows = 1, cols = 1:12)

addStyle(wb, sheet = 2, createStyle(halign = 'center'), rows = 2:(nrow(subset(res\_tmp,log2FoldChange <= -lfc & FDR < fdr)) + 1), cols = c(1,3:10), gridExpand = TRUE)

addStyle(wb, sheet = 2, createStyle(numFmt = 'SCIENTIFIC', halign = 'center'), rows = 2:(nrow(res\_tmp) + 1), cols = c(11:12), gridExpand = TRUE)

writeData(wb = wb, sheet = 3, x = res\_tmp)

setColWidths(wb, sheet = 3, cols = 1:12, widths = c(24,43.15,13.65,15.75,20.4,25.25,20.9,26.4,16.45,26.4,9.5,9.5))

addStyle(wb, sheet = 3, createStyle(border='Bottom',textDecoration='bold',halign = 'center'), rows = 1, cols = 1:12)

addStyle(wb, sheet = 3, createStyle(halign = 'center'), rows = 2:(nrow(res\_tmp) + 1), cols = c(1,3:10), gridExpand = TRUE)

addStyle(wb, sheet = 3, createStyle(numFmt = 'SCIENTIFIC', halign = 'center'), rows = 2:(nrow(res\_tmp) + 1), cols = c(11:12), gridExpand = TRUE)

saveWorkbook(wb, "4\_13\_lcl\_infected\_vs\_4\_13\_mock.xlsx", overwrite = TRUE)