Transcriptome analysis of pathogens – further datasets

A. Introduction

In this exercise we are going to look in a RNA-Seq dataset from *Clostridium difficile* from the paper:

Differential Stress Transcriptome Landscape of Historic and Recently Emerged Hypervirulent Strains of *Clostridium difficile* Strains Determined Using RNA-seq Scaria et al, PLoS ONE, 2013

The author took four different *C. difficile* strains and put them through two stresses; 1. Osmotic drop and 2. Nutrition change, with biological replicates.

B. Exercise

Aim of the exercise is to analyse an RNA-Seq dataset with more than two different conditions. As we have already performed mapping of reads (RNA-Seq and genomic) we pre-mapped the reads and generated the raw reads count for each transcript (different to RPKM).
Following steps will be performed:

- Quality the data through different correlation methods in R
- Apply the clustering method MBCluster.Seq
- Look at the expression profiles of the different cluster
- Access the function of each cluster through the product
- Perform GO enrichment of each cluster, to look for enrichment
- If time permits to pairwise differential expression with DESeq and compare the outcome to the results of the clustering

The questions will be:

- Do specific clusters represent specific functions
- Can the different function explain the results
- How confident would you be in the findings
- Would the GO enrichment be as powerful as for example pathway enrichment, as performed in the paper.

Important: Please do not quit the R terminal through the complete exercise!

1. Quality control

Create a directory and download the read Count files

```
$ cd
$ mkdir Module_RNA-Seq2
$ cd Module_RNA-Seq2
$ wget
ftp://ftp.sanger.ac.uk/pub/pathogens/tdo/London/ReadCounts.zip #
one line! You can also download through a browser
$ unzip ReadCounts.zip
```

Look into the file *Cdiff.Readcounts.txt*. Do a

```
$ head Cdiff.Readcounts.txt
```

The files *Cdiff.ReadCounts.txt* contains the reads count for each gene in the six different conditions. 630 and R20291 represent the different strains. WT is the wild type, OS osmotic shift and NS Nutrition shift. Of each condition we have two replicates.
The other two files contain the same information, but are in the format for the clustering program.

The read counts were obtained with bedtools (*bedtools multicov -q 5 -D -p -bams aln.1.bam aln.2.bam ... aln.n.bam -bed Cdiff.gtf > readCount.pre.txt*). Basically, the amount of reads on each transcript is counted. Reads with a mapping quality below 5 are exclude (Repetitive mapped reads). We do include PCR replicates with the -D parameter.
First things to do is to quality control of the data. We want to understand how similar the biological replicates are and how different the conditions are.
How could you do that?
How could you get some correlation values between the samples?

Here it becomes quite handy that you got some introduction in R. The idea is to first look at some scatterplot between the 12 different measurement and calculate the Pearson correlation.

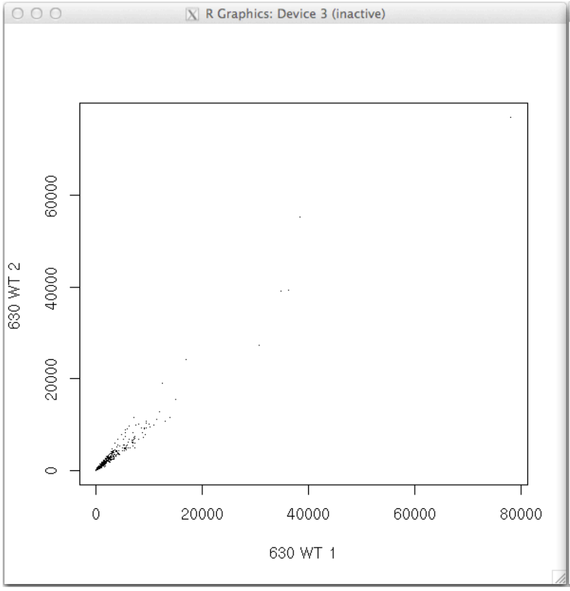
Open R and load the file Cdiff.Readcounts.txt

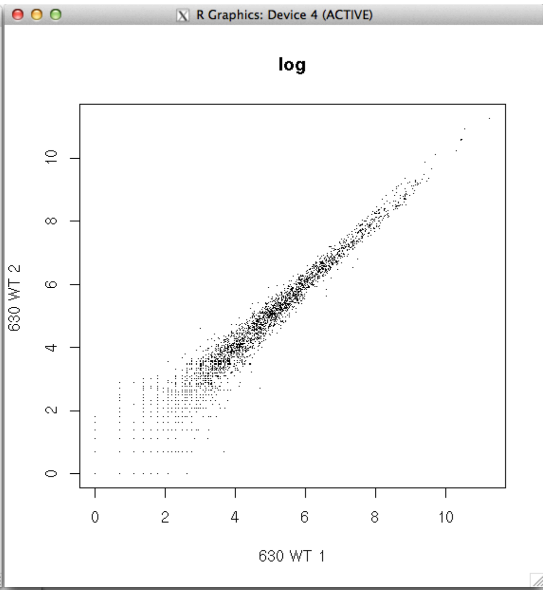
```
$ R
> countTable <- read.table( "Cdiff.Readcounts.txt", header=T, row.names=1)
> head(countTable)
```

The last command 'head' is very similar in R than in Linux. Next compare the reads counts between the two biological replicates of the WT of the strain 630.

```
> plot(countTable[,1],countTable[,2],pch=".",xlab="630 WT 1", ylab="630 WT
2")
> x11()
> plot(log(countTable[,1]),log(countTable[,2]),pch=".",main="log",xlab="630
WT 1", ylab="630 WT 2")
```

Which plot illustrates better the difference between the two replicates? (Plots are also on the next page).





It seems that they look pretty good… Now get the correlation:

```
> cor(countTable[,1],countTable[,2])
[1] 0.9843846
cor(log(countTable[,1]+1),log(countTable[,2]+1))
[1] 0.9741434
```

Now please compare following conditions

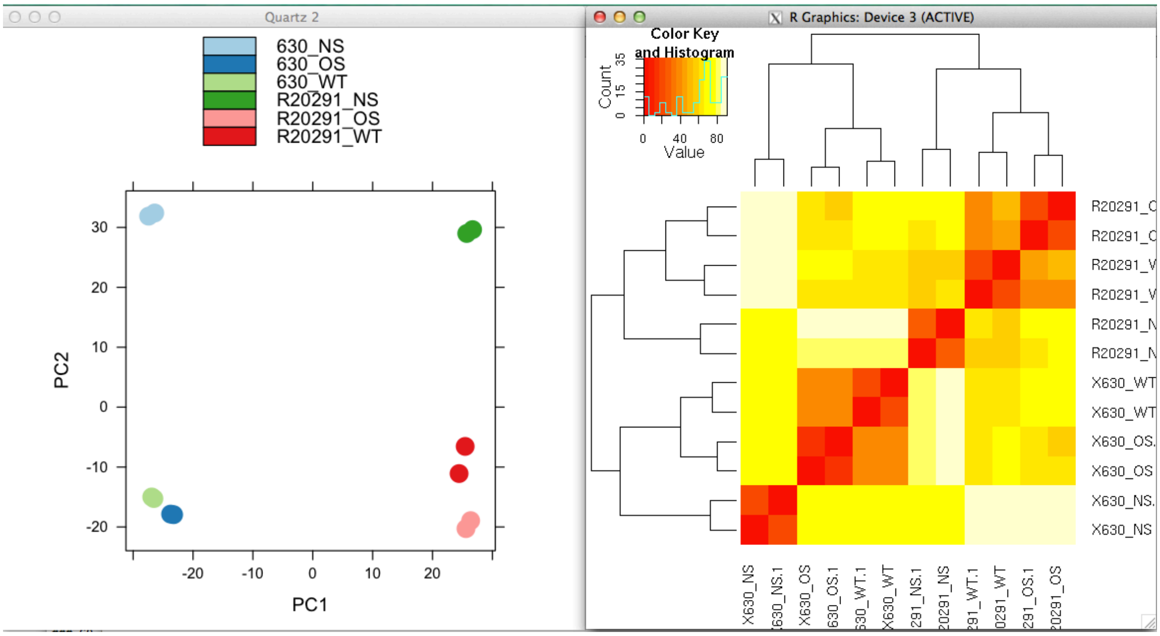
- 630 WT versus 630 OS
- 630 WT versus 630 NS
- 630 WT versus R20291 WT

by doing the scatterplot and the Pearson correlation (cor).

Discuss with your neighbour, why the correlation between the WT versus OS is higher than the correlation between WT NS.

There are other way to product a correlation. We are going to use two functions from DESeq package:

```
library(DESeq)
library("gplots")
conds=c("630_WT","630_WT","630_OS","630_OS","630_NS","630_NS",
"R20291_WT","R20291_WT","R20291_OS","R20291_OS","R20291_NS","R
20291_NS")
cds <- newCountDataSet( countTable, conds )
cds <- estimateSizeFactors( cds )
cds <- estimateDispersions( cds)
vsd <- varianceStabilizingTransformation( cds )
plotPCA( vsd )
x11()
dists = dist( t( exprs(vsd) ) )
heatmap(as.matrix( dists ), trace="none")
```



You should see two plots… and yes, a lot of typing of weird functions. If needed, we can discuss the meaning of the calls later in the group. But basically, the reads counts of the different conditions are normalized through a function in DESeq. Then, we plot difference between the 12 samples as PCA (Principal component Analysis) plot and as heatmap.

We hope that you agree that those plots are nicer than the scatterplots. They basically represent a pairwise comparison of all the samples at the same time.

Question to discuss with your neighbour:

- How good are the replicates?
- Why is there a split on the principal component 1 between the two strains?
- Which conditions seem to be more different?

This quality control analysis is VERY crucial in RNA-Seq experiments. If your biological replicates do not correlate well, or are even mixed with other conditions could means that you might not be able to analyse the data…

Interestingly (or as expected) there are differences between the conditions. The aim is now to find the genes that change expression, for example with the osmotic drop. On the next page, we are going to introduce the how to do differential expression with DESeq. This exercise is OPTIONAL, as the drawback is that the comparison can just be done pairwise. But we prefer to look into a method where all condition can be analysed at once, page 6.

2. Pairwise differential expression with DESeq +++ OPTIONAL +++

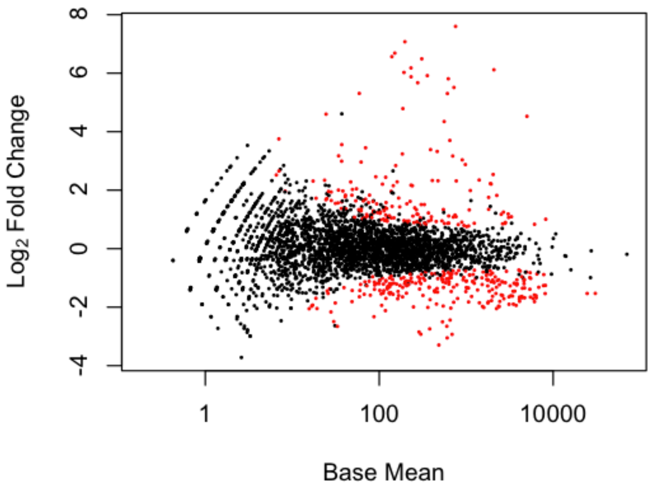
As we already done the correlation with DESeq, the data are already is a perfect format. First we need to do the binomial test, to see which genes are differentially expressed.

```
> res <- nbinomTest( cds, "630_WT", "630_OS" )
> head( res )
```

returns you the results for each gene. It is very similar to the results from cuffdiff from the exercise before, like the fold change and the likelihood that the gene is differentially expressed.

With the following command you can visualize the data (through an MA-plot). Due to time restriction, we show here the plots, so you don't need to type the commands…

```
plotDE <- function( res, sig ) plot( res$baseMean, res$log2FoldChange, log="x", pch=20, cex=.3, col = ifelse( res$padj <
sig, "red", "black" ), xlab="Base Mean", ylab=expression(Log[2]~Fold~Change) ) ;
plotDE(res, .1)
```



MA-plot red dots represent gene that are differentially expressed.

To obtain the most differentially up-regulated expressed genes do:

```
> resSig <- res[ res$padj < 0.1, ]
> resSig[order(-resSig$foldChange, -resSig$baseMean)[1:min(10, length(resSig$id))],]
```

And the down regulated one

```
> resSig[order(resSig$foldChange, -resSig$baseMean)[1:min(10, length(resSig$id))],]
```

Obviously, at with stage we cannot see the function. Leave R with ctrl-z and type for example

```
$ grep CD630_23400 Cdiff.product.txt
```

Following functions are down-regulated

"Succinate-semialdehyde dehydrogenase (NAD(P)+)" "Succinyl-CoA:coenzyme A transferase" and "putative membrane protein".

Is this something expected for osmotic shift? Go back to R with

```
$ fg
```

3. Cluster analysis with MBCluster.Seq

Though doing pairwise differential expression a feasible method to analyse the data, it is also possible to analyse them all together. For this we are going to classify (or cluster) the reads depending on their expression profile over the 6 condition. Here we use MBCluster.Seq which is a Bioconductor package. It implements a k-means method, and processes the reads counts though a negative binominal modelling…. any questions?

Due to time restriction, and because we want to be nice, you can copy the code from a file.

The code is in the file code3.txt. Leave R with crtl-z

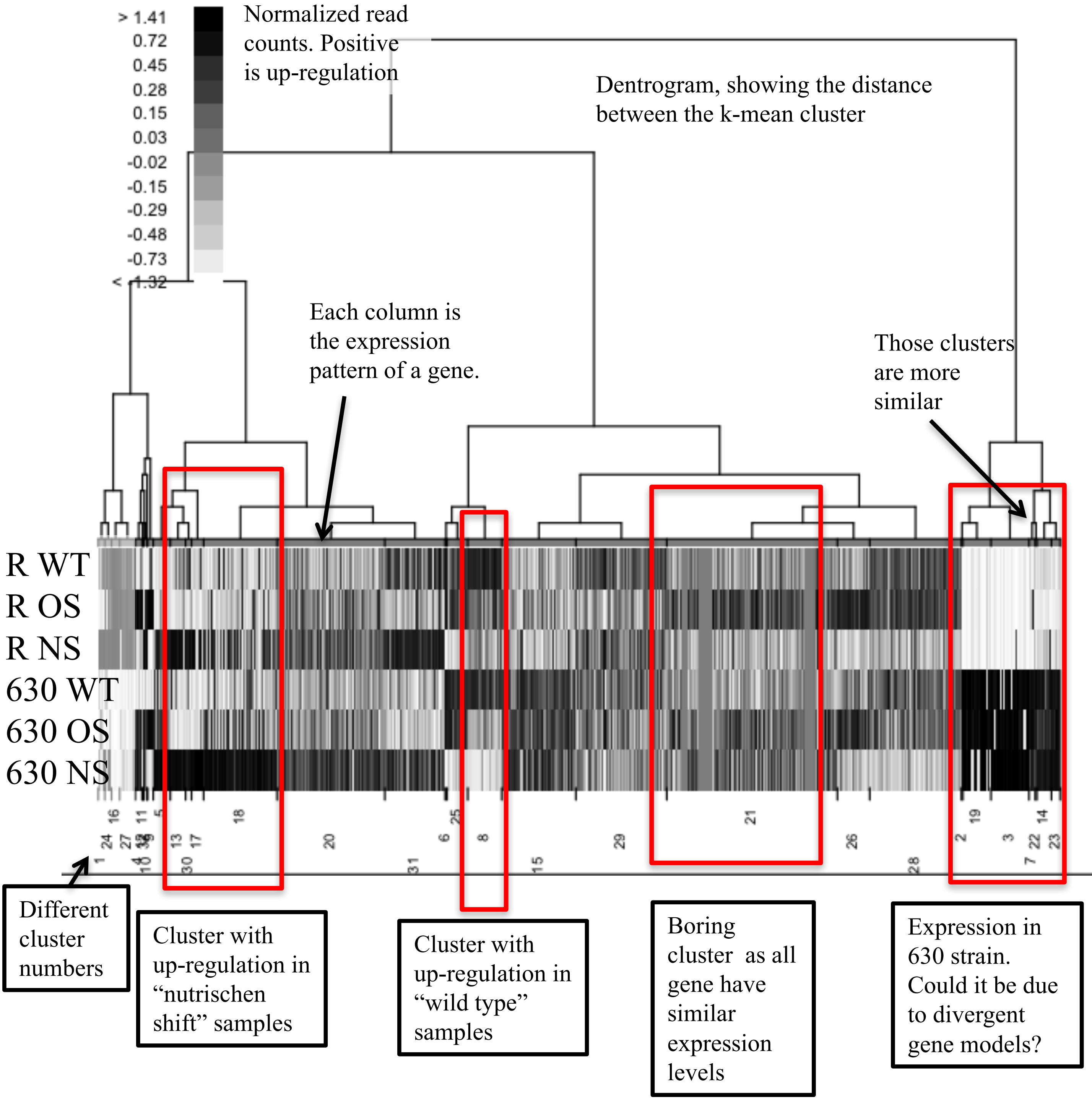
```
$ cat code3.txt
$ fg
```

And now copy and paste the code into R (you can selected the content of the file *code3.txt* with the mouse and paste with the middle mouse bottom (wheel)). In the wrap up we can discuss the meaning of different lines. The figure is explained on the next page. While the code is processed, you might want to have a quick look at it.

```
library(MBCluster.Seq)
countTable<-read.table("Cdiff.Readcounts.txt",header=T,sep="\t")
#a sample data set with RNA-seq expressions read counts
GeneID<-countTable[,1]; Count=countTable[,-1]
Normalizer=rep(1,ncol(Count))
# set the treaments
Treatment=c("630_WT","630_WT","630_OS","630_OS","630_NS","630_NS",
"R20291_WT","R20291_WT","R20291_OS","R20291_OS","R20291_NS","R2029
1_NS")
n=32 ## here define the amount of cluster
mydata=RNASeq.Data(Count,Normalize=NULL,Treatment,t(GeneID))
#standardized RNA-seq data
c0=KmeansPlus.RNASeq(mydata,nK=n)$centers
#choose n cluster centers to initialize the clustering
cls=Cluster.RNASeq(data=mydata,model="nbinom",centers=c0,method="E
M")$cluster
#use EM algorithm to cluster genes
tr=Hybrid.Tree(data=mydata,cluste=cls,model="nbinom")
#bulild a tree structure for the resulting 10 clusters
plotHybrid.Tree(merge=tr,cluster=cls,logFC=mydata$logFC,tree.title
=NULL)
# get the annotation
Product<-read.table("Cdiff.product.txt", sep="\t");
### combine the files and save it.
write(cls, file="Cluster.Cdiff.32cluster.txt",sep="\n")
```

The heatmap - output of MBCluster.Seq

Your output might be slightly different, as the initialization is random. Every column represents a gene and the rows are the 6 different conditions. The dentrogram reflexes the structure of the cluster.



Normalized read counts. Positive is up-regulation

Dentrogram, showing the distance between the k-mean cluster

Each column is the expression pattern of a gene.

Those clusters are more similar

Different cluster numbers

Cluster with up-regulation in “nutrischen shift” samples

Cluster with up-regulation in “wild type” samples

Boring cluster as all gene have similar expression levels

Expression in 630 strain. Could it be due to divergent gene models?

Looking at specific cluster

Now we want to see which genes have specific expression pattern. This can be easily done in R.

Have a look at the Product structure. It has the gene id as first column and then the Product. (still be in R)

```
> head(Product)
```

The vector *cls* contains the cluster-number for each gene.

```
> head(cls)
```

So to see the product of the genes from cluster 8 do

```
> Product[cls==8,1:2]
```

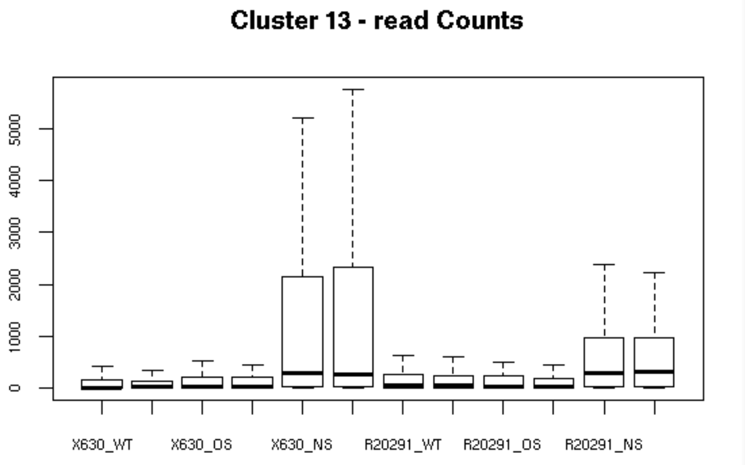
To visualize the counts for cluster 13

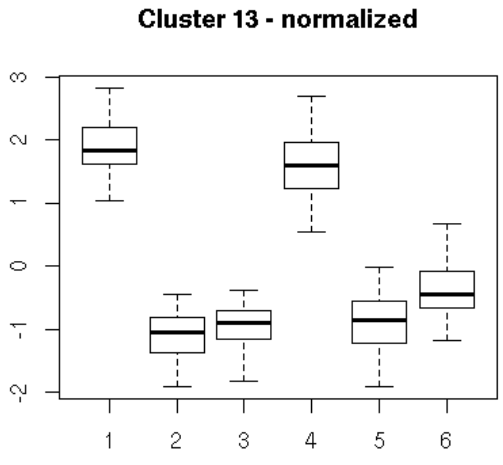
```
> boxplot(Count[cls==13,],outline = F)
> x11()
```

Now, show the normalized data for the same cluster

```
> boxplot(mydata$logFC[cls==13,] , outline = F)
```

Careful, the order in the boxplot is different!





Discuss following questions:

- Which clusters are interesting?
- Which cluster sshow differences between the two strains?
- Which cluster show differences in the Osmotic shift / Nutrition shift? How is the signal in the two strains?
- Do you prefer the cluster analysis or the pairwise differential expression?

4. Go enrichment

So now we have clusters that contain genes, that have the same expression patter, for example higher expression in the nutrition shift. The next step is to understand - through the analysis of the clusters - how the bacteria behaves due to the change of the environment. One obvious application is to do a GO-enrichment for interesting clusters.

As you are still in R, we can do the GO enrichment directly here. We prepared a file (*Cdiff.Goterm.txt*) with the GO terms for *C. difficile*. You can obtain the GO terms for most species from the EBI GOA website (http://www.ebi.ac.uk/GOA).
This time, the function to perform the GO enrichment is in a separate file that first needs to be loaded:

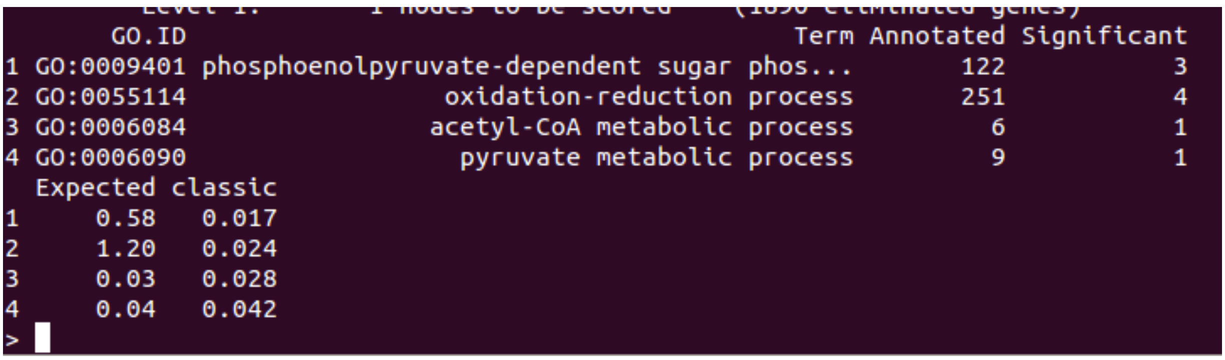
```
> source(“doGO.R”)
> library(“topGO”)
```

Now, you can call the function with for example:

```
> doGO(Product[cls==17,1],”BP”)
```

This will do the enrichment for genes in cluster 13 and look into the Biological process (BP). BTW, the other are cell component (CC) and molecular function (MF).

Here we are going to show the results of cluster 17, which is highly enriched for genes expressed in nutrition shift in both strains. Following results are given for BP:



```
       Level 1:        1 nodes to be scored    (1896 eliminated genes)
       GO.ID                                   Term Annotated Significant
1 GO:0009401 phosphoenolpyruvate-dependent sugar phos...       122           3
2 GO:0055114                 oxidation-reduction process       251           4
3 GO:0006084                acetyl-CoA metabolic process         6           1
4 GO:0006090                  pyruvate metabolic process         9           1
  Expected classic
1     0.58   0.017
2     1.20   0.024
3     0.03   0.028
4     0.04   0.042
> 
```

So, look at some GO enrichments for clusters you find interesting.

- For clusters that have similar expression pattern in the heatmap, do you get similar enriched GO terms?
- Could you confirm that cluster with many genes of similar expression represent the same function in the cell?

5. Integrated analysis?

At this stage, we should be able to get an idea of the function of the genes in each cluster. This was done by looking at the attributed function and also through the GO analysis. But next we would like to analysis those results in a more combined way. The idea is to integrate further information to our data with the hope to understand better the impact of the experiments.

First we are going to look at protein interaction between specific proteins. One example is the STING database (http://string-db.org/), where interaction between proteins is recorded.

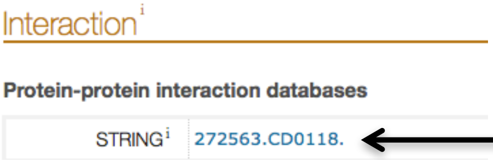
Second we are going to look at metabolic pathways in KEGG. The aim would not be to do a pathway enrichment per se, but more to give you an idea how to look at enzymatic maps.

Important: The aim of this section is not to give you a complete overview of the tools, but rather to show you how the results from function studies could be further analyzed!

5.1 Protein Interaction

Go the interpro website (http://www.uniprot.org/) and search for the gene id CD630_01180. Open the entry.
This gene is part of the cluster 17, highly expressed in the nutrition shift. This proteins probably encodes a “ferredoxin/flavodoxinoxidoreductase, gamma subunit”.

You can skim through the result page, but the goal is to find the STRING entry and to open that page:



Interaction[i]

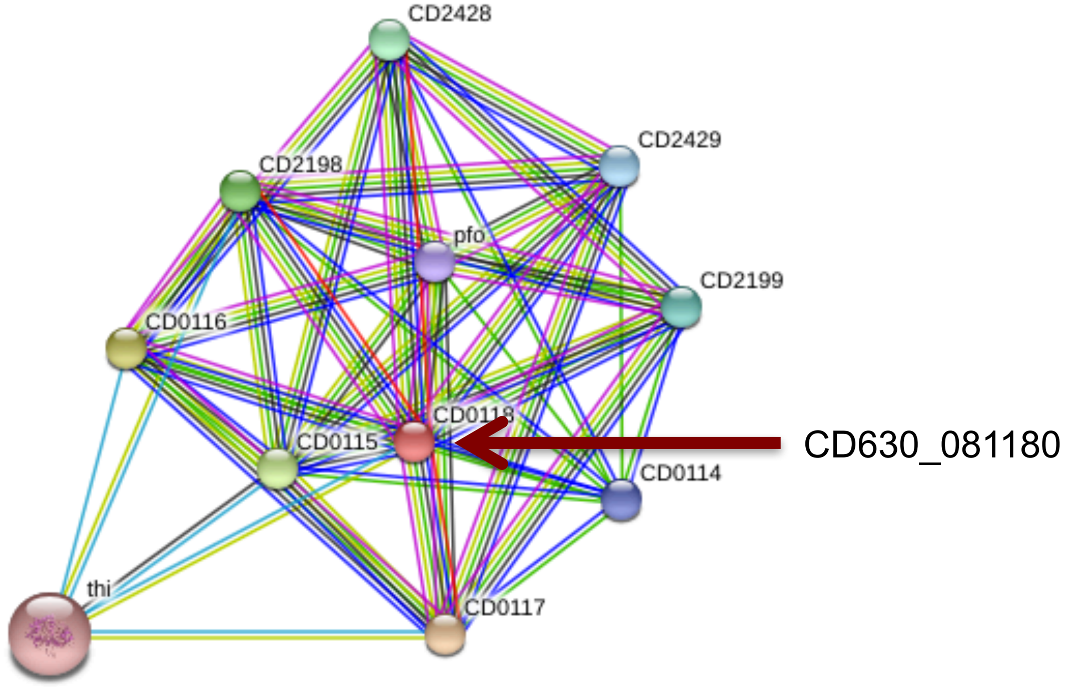
Protein-protein interaction databases

STRING[i] 272563.CD0118.

click here

A graph should pop up where the nodes are proteins in *C. difficile* and the edge represent different interactions. The proteins are named slightly different CD630_01180 transforms to CD_1180. Several other proteins with a similar ID (some operon?) are connected. Could they be other subunits of the ferredoxin?
Would you expect that the connected proteins are expressed similarly? So are they in the same cluster?

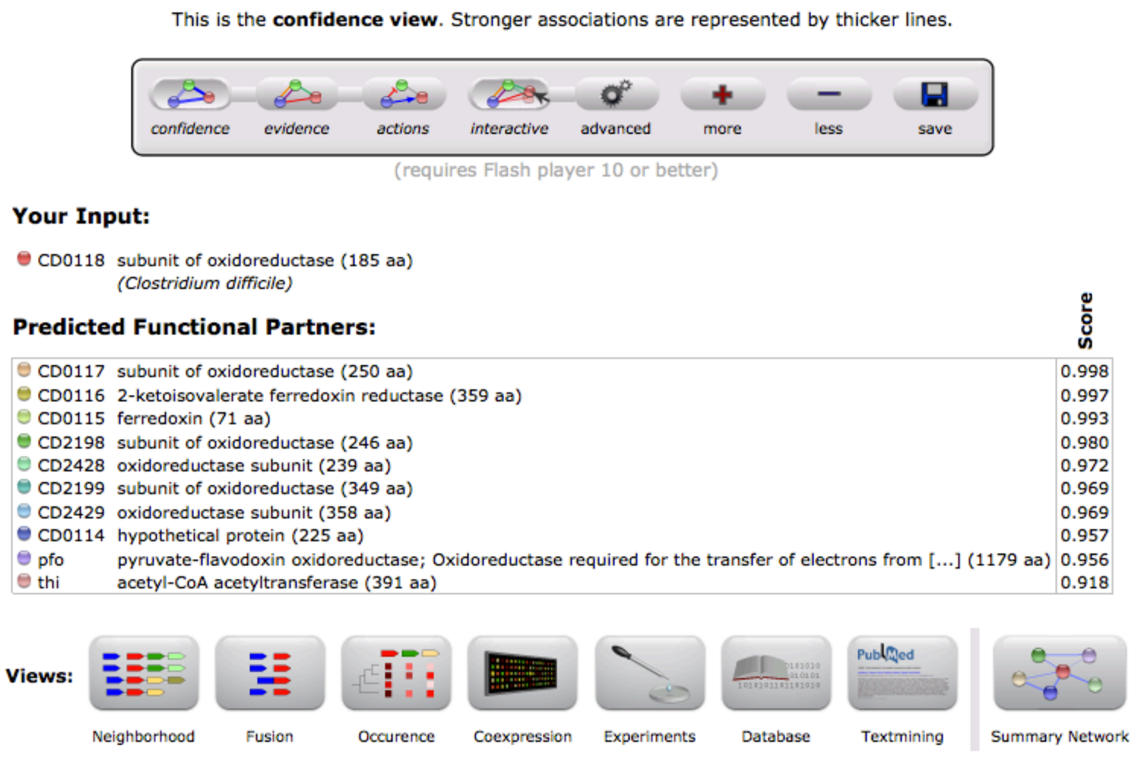
Explore the options of the tools. Maybe go back to the cluster analysis and look for the other genes. Some are in neighboured clusters.



If you click on a note the information on the protein is given.

If you click on the edge, the evidence for the interaction is given. You can see, that the connection are base on neighbours in the genomes, gene fusion (in orthologous), co-expression, experimental data etc.

Try some of the options below…



This is the **confidence view**. Stronger associations are represented by thicker lines.

(requires Flash player 10 or better)

Your Input:

● CD0118 subunit of oxidoreductase (185 aa)
(Clostridium difficile)

Predicted Functional Partners:

		Score
● CD0117	subunit of oxidoreductase (250 aa)	0.998
● CD0116	2-ketoisovalerate ferredoxin reductase (359 aa)	0.997
● CD0115	ferredoxin (71 aa)	0.993
● CD2198	subunit of oxidoreductase (246 aa)	0.980
● CD2428	oxidoreductase subunit (239 aa)	0.972
● CD2199	subunit of oxidoreductase (349 aa)	0.969
● CD2429	oxidoreductase subunit (358 aa)	0.969
● CD0114	hypothetical protein (225 aa)	0.957
● pfo	pyruvate-flavodoxin oxidoreductase; Oxidoreductase required for the transfer of electrons from [...] (1179 aa)	0.956
● thi	acetyl-CoA acetyltransferase (391 aa)	0.918

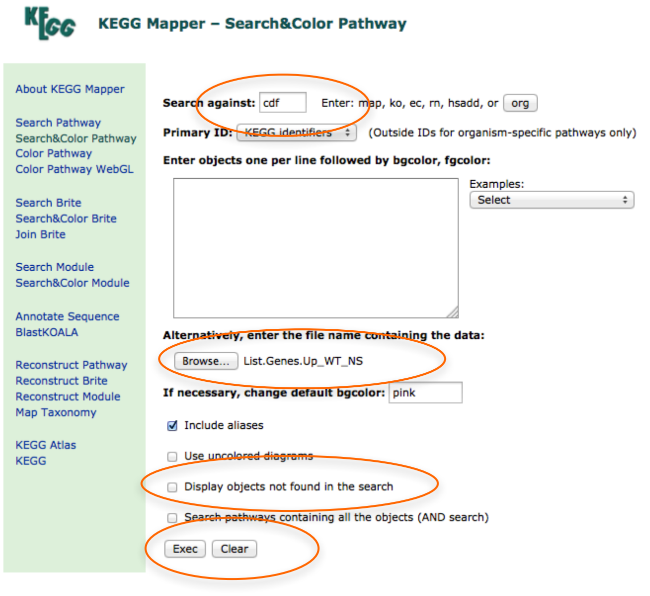
5.2 Metabolic Pathways

Another possibility of visualizing the data is to look at pathways, i.e. signal pathways or metabolic pathways. Let’s look at the latter in KEGG (http://www.kegg.jp/)!

Again, there are many tools, some might be better… important here is to know that tool to visualize pathways exist.

First, you have to generate a list of gene id’s that you want to visualize in KEGG. You can save them in R, as a list of gene id’s. Alternatively, you can use the list of genes from cluster 17 (name: *List.Genes.Cluster17.txt*) or the genes up-regulated in DESeq 630 WT vs NS (filename: *List.Genes.Up_WT_NS.txt*).

Open the webpage http://www.kegg.jp/kegg/tool/map_pathway2.html in a browser. (or search for kegg paint pathway).



KEGG Mapper – Search&Color Pathway

Search against: cdf Enter: map, ko, ec, rn, hsadd, or org

Primary ID: KEGG identifiers (Outside IDs for organism-specific pathways only)

Enter objects one per line followed by bgcolor, fgcolor:

Alternatively, enter the file name containing the data: List.Genes.Up_WT_NS

If necessary, change default bgcolor: pink

☑ Include aliases

☐ Use uncolored diagrams

☐ Display objects not found in the search

☐ Search pathways containing all the objects (AND search)

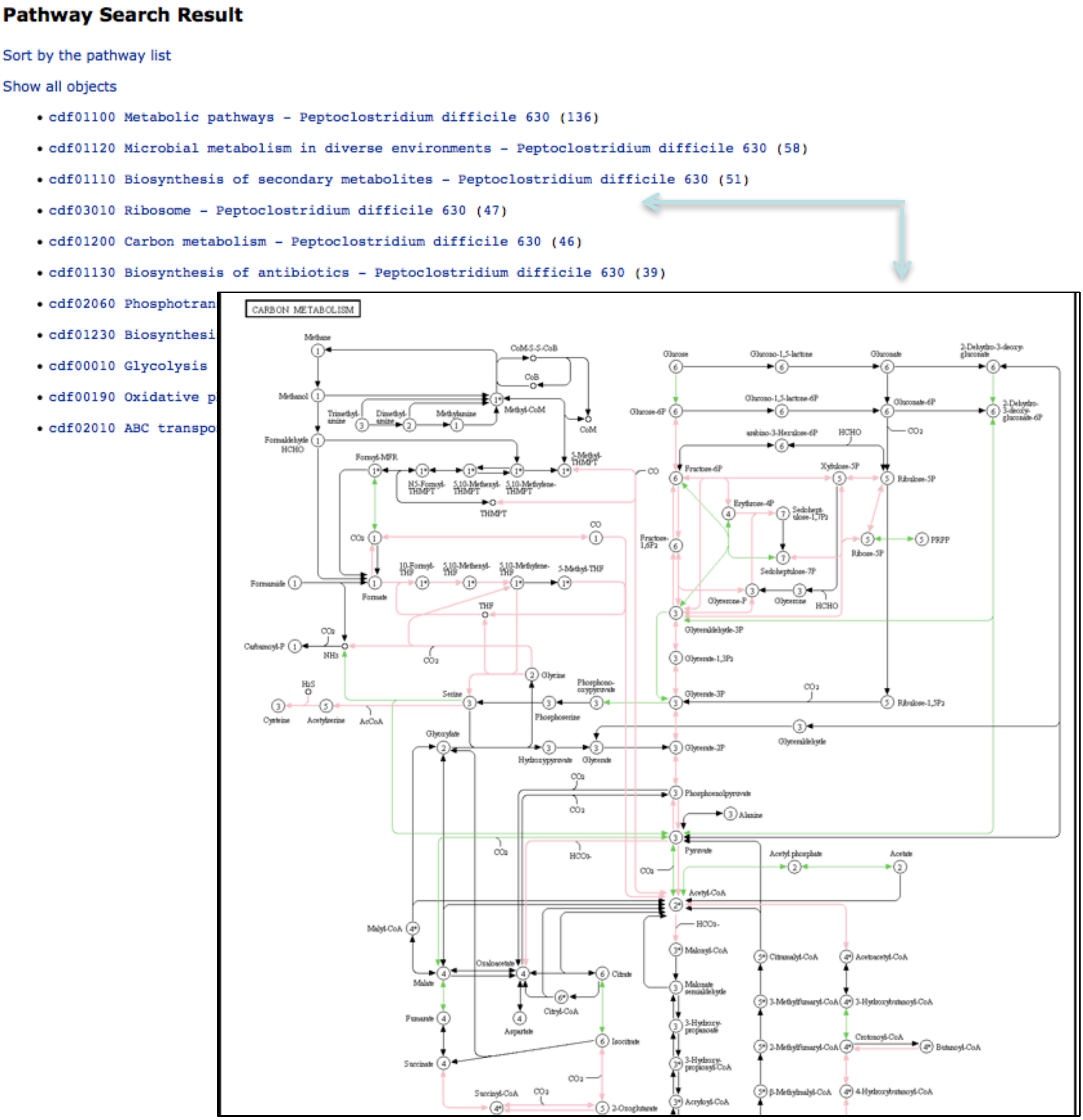
1. Select the *C. difficile* as species ‘cdf’

2. Select the file you want to upload

3. Unselect this box

4. Press exec

The painting tool will check in all the pathways - that are generated for *C. difficile* - how many genes are listed in your list. It will paint those genes red. Enzymes that are green are annotated in *C. difficile*, but are not in the list, therefore not up-regulated.



Pathway Search Result

Sort by the pathway list

Show all objects

- cdf01100 Metabolic pathways - Peptoclostridium difficile 630 (136)
- cdf01120 Microbial metabolism in diverse environments - Peptoclostridium difficile 630 (58)
- cdf01110 Biosynthesis of secondary metabolites - Peptoclostridium difficile 630 (51)
- cdf03010 Ribosome - Peptoclostridium difficile 630 (47)
- cdf01200 Carbon metabolism - Peptoclostridium difficile 630 (46)
- cdf01130 Biosynthesis of antibiotics - Peptoclostridium difficile 630 (39)
- cdf02060 Phosphotran
- cdf01230 Biosynthesi
- cdf00010 Glycolysis
- cdf00190 Oxidative p
- cdf02010 ABC transpo

Basically, some of the pathways have more genes from your list (enriched?). So the nutrition shift has more effect on those pathways…

Luckily, this is also a finding in the original paper.