Microarray analysis (GEO24842)

Reference: Williams K, Christensen J, Pedersen M, Johansen JV, Cloos PAC, Rappsilber J et al. Tet1 and hydroxymethylcytosine in transcription and DNA methylation fidelity. Nature 2011, 473(7347):343-348.

Summary of the article: Methylation of cytosines by DNA methyl transferases is an important factor contributing to the downregulation of gene expression. This epigenetic modification mostly occurs in CpG islands, expect in the ones of promotors. Tet1 katalyzes the hydoroxylation of methylated cytosines, which results in demethylation. Previous studies show that methylated cytosines frequently occur in embryonic stem cells and that this modification is a crucial factor in gene regulation.

In this paper, mouse embryonic stem cells were used in order to unravel the function of Tet1 since these cells exhibit a high Tet1 expression level. By using ChIP-seq, they found out that Tet1 preferentially binds to transcription start sites and gene bodies of CpG-rich promotors. Microarray was performed on both normal and Tet1-knockdown embryonic stem cells and showed that Tet1 represses transcription. Next, they investigated the molecular pathway of this repression by using ChIP-seq and found out that Tet1 recrutes SIN3A, thereby inhibiting transcription.

Location of the original ipynb: /mnt/storage/r0368030/jupyternotebooks

Files: /mnt/storage/r0368030/Microarray

In [3]:

```
# Loading of the limma and affy library.
library(limma)
library(affy)
```

```
Attaching package: ‘limma’

The following object is masked from ‘package:BiocGenerics’:

    plotMA
```

First, the CEL-files were downloaded from the GEO website and placed (after unzipping) in the Microarray directory. In the article, they used two different antibodies against TET1 in triplicate. We only downloaded the data from one antibody in triplicate. To load these CEL-files, we used the ReadAffy function of the affy package.

In [4]:

```
celfiles = Sys.glob("/mnt/storage/r0368030/Microarray/*.gz")
celfiles
```

'/mnt/storage/r0368030/Microarray/GSM611209.CEL.gz'
'/mnt/storage/r0368030/Microarray/GSM611210.CEL.gz'
'/mnt/storage/r0368030/Microarray/GSM611211.CEL.gz'
'/mnt/storage/r0368030/Microarray/GSM611212.CEL.gz'
'/mnt/storage/r0368030/Microarray/GSM611213.CEL.gz'
'/mnt/storage/r0368030/Microarray/GSM611214.CEL.gz'
'/mnt/storage/r0368030/Microarray/GSM675884.CEL.gz'
'/mnt/storage/r0368030/Microarray/GSM675885.CEL.gz'
'/mnt/storage/r0368030/Microarray/GSM675886.CEL.gz'
'/mnt/storage/r0368030/Microarray/GSM675887.CEL.gz'
'/mnt/storage/r0368030/Microarray/GSM675888.CEL.gz'
'/mnt/storage/r0368030/Microarray/GSM675889.CEL.gz'

In [5]:

```
# Loading of the CEL-files.
rawcel <- ReadAffy(filenames=celfiles)
```

```
Warning message:
:

The affy package can process data from the Gene ST 1.x series of arrays,
but you should consider using either the oligo or xps packages, which are sp
ecifically
designed for these arrays.
```

In [6]:

```
sampleNames(rawcel)
```

'GSM611209.CEL.gz' 'GSM611210.CEL.gz' 'GSM611211.CEL.gz'
'GSM611212.CEL.gz' 'GSM611213.CEL.gz' 'GSM611214.CEL.gz'
'GSM675884.CEL.gz' 'GSM675885.CEL.gz' 'GSM675886.CEL.gz'
'GSM675887.CEL.gz' 'GSM675888.CEL.gz' 'GSM675889.CEL.gz'

Next, we changed the CEL-filenames into more meaningfull names based the information that was available on the GEO website by using a vector (knockdown). Then, the CEL-filenames and knockdown names are represented together in a table.

In [7]:

```
# The CEL-files are renamed.
knockdown = c("Scrambled_Tet1_1",
              "Scrambled_Tet1_2",
              "Scrambled_Tet1_3",
              "Tet1 knockdown_1",
              "Tet1 knockdown_2",
              "Tet1 knockdown_3",
              "Scrambled_SIN3A_1",
              "Scrambled_SIN3A_2",
              "Scrambled_SIN3A_3",
              "SIN3A knockdown_1",
              "SIN3A knockdown_2",
              "SIN3A knockdown_3")
```

In [8]:

```
pd <- data.frame(knockdown = knockdown)
rownames(pd) <- sampleNames(rawcel)
pd
```

	knockdown
GSM611209.CEL.gz	Scrambled_Tet1_1
GSM611210.CEL.gz	Scrambled_Tet1_2
GSM611211.CEL.gz	Scrambled_Tet1_3
GSM611212.CEL.gz	Tet1 knockdown_1
GSM611213.CEL.gz	Tet1 knockdown_2
GSM611214.CEL.gz	Tet1 knockdown_3
GSM675884.CEL.gz	Scrambled_SIN3A_1
GSM675885.CEL.gz	Scrambled_SIN3A_2
GSM675886.CEL.gz	Scrambled_SIN3A_3
GSM675887.CEL.gz	SIN3A knockdown_1
GSM675888.CEL.gz	SIN3A knockdown_2
GSM675889.CEL.gz	SIN3A knockdown_3

In [9]:

```
library(Biobase)
# The created dataframe is now converted into an AnnotatedDataFrame.
phenoData(rawcel) <- AnnotatedDataFrame(data=pd)
```

We used SCAN.UPC to normalize our data as an alternative for frma. SCAN normalizes each sample individually by removing probe- and array-specific background noise using only data from within each array. The Universal exPression Codes (UPC) method estimates whether a given gene or transcript is active above background levels in a given sample. Next, a SCAN.eset file is generated and saved in het Microarray directory.

In [10]:

```
library(SCAN.UPC)
```

```
Loading required package: oligo
Loading required package: oligoClasses
Welcome to oligoClasses version 1.30.0

Attaching package: ‘oligoClasses’

The following object is masked from ‘package:affy’:

    list.celfiles

Loading required package: Biostrings
Loading required package: XVector
========================================================================
====
Welcome to oligo version 1.32.0
========================================================================
====

Attaching package: ‘oligo’

The following objects are masked from ‘package:affy’:

    intensity, MAplot, mm, mm<-, mmindex, pm, pm<-, pmindex,
    probeNames, rma

The following object is masked from ‘package:limma’:

    backgroundCorrect

Loading required package: GEOquery
Setting options('download.file.method.GEOquery'='curl')
Loading required package: affyio
Loading required package: foreach
Parallel computing support for 'oligo/crlmm': Disabled
     - Load 'ff'
     - Load and register a 'foreach' adaptor
        Example - Using 'multicore' for 2 cores:
             library(doMC)
             registerDoMC(2)
========================================================================
====
Loading required package: sva
Loading required package: mgcv
Loading required package: nlme

Attaching package: ‘nlme’

The following object is masked from ‘package:Biostrings’:

    collapse

The following object is masked from ‘package:IRanges’:

    collapse

This is mgcv 1.8-16. For overview type 'help("mgcv-package")'.
Loading required package: genefilter

Attaching package: ‘genefilter’
```

```
The following object is masked from ‘package:base’:

    anyNA
```

In [11]:

```
celFilePath = file.path("/mnt/storage/r0368030/Microarray/*.gz")
```

In [17]:

```
norm = SCAN(paste(celFilePath, sep="/"))
```

```
Warning message:
: executing %dopar% sequentially: no parallel backend registeredLoading re
quired package: pd.mogene.1.0.st.v1
Loading required package: RSQLite
Loading required package: DBI
Platform design info loaded.

Reading in : /mnt/storage/r0368030/Microarray/GSM611209.CEL.gz

Starting EM
Attempting to converge...iteration 1, c = 25.655448
Attempting to converge...iteration 2, c = 151.237319
Attempting to converge...iteration 3, c = 1.615866
Attempting to converge...iteration 4, c = 1.641265
Attempting to converge...iteration 5, c = 45.796311
Attempting to converge...iteration 6, c = 0.329374
Attempting to converge...iteration 7, c = 0.170804
Attempting to converge...iteration 8, c = 0.108798
Attempting to converge...iteration 9, c = 0.100099
Attempting to converge...iteration 10, c = 0.064869
```

In [18]:

```
save(norm, file="/mnt/storage/r0368030/Microarray/GSE24842.SCAN.eset")
```

In [19]:

```
norm
```

```
ExpressionSet (storageMode: lockedEnvironment)
assayData: 35556 features, 12 samples
  element names: exprs
protocolData: none
phenoData: none
featureData: none
experimentData: use 'experimentData(object)'
Annotation:
```

The samplenames are again not very meaningfull, so we converted them info more informative names and presented both in a table.

In [20]:

```
sampleNames(norm)
```

'GSM611209.CEL.gz' 'GSM611210.CEL.gz' 'GSM611211.CEL.gz'
'GSM611212.CEL.gz' 'GSM611213.CEL.gz' 'GSM611214.CEL.gz'
'GSM675884.CEL.gz' 'GSM675885.CEL.gz' 'GSM675886.CEL.gz'
'GSM675887.CEL.gz' 'GSM675888.CEL.gz' 'GSM675889.CEL.gz'

In [21]:

```
pData(phenoData(norm)) <- pd
```

In [22]:

```
eset.labels <- c("ShScrTet1_1", "ShScrTet1_2", "ShScrTet1_3", "ShTet1_1",
                 "ShTet1_2", "ShTet1_3", "ShScrSIN3A_1", "ShScrSIN3A_2",
                 "ShScrSIN3A_3", "ShSIN3A_1", "ShSIN3A_2", "ShSIN3A_3")

sampleNames(norm) <- eset.labels
```

In [23]:

```
pData(phenoData(norm))
```

	knockdown
ShScrTet1_1	Scrambled_Tet1_1
ShScrTet1_2	Scrambled_Tet1_2
ShScrTet1_3	Scrambled_Tet1_3
ShTet1_1	Tet1 knockdown_1
ShTet1_2	Tet1 knockdown_2
ShTet1_3	Tet1 knockdown_3
ShScrSIN3A_1	Scrambled_SIN3A_1
ShScrSIN3A_2	Scrambled_SIN3A_2
ShScrSIN3A_3	Scrambled_SIN3A_3
ShSIN3A_1	SIN3A knockdown_1
ShSIN3A_2	SIN3A knockdown_2
ShSIN3A_3	SIN3A knockdown_3

Next, we looked at the normalized expression data for every condition (knockdown and wildtype Tet1 and SIN3A). However, the probenames are not very informative. Therefore we tried to replace these probe names by gene names using two additional libraries (annotate and pd.mogene.1.0.st.v1). We were not able to annatate with the pd.mogene.1.0.st.v1 library so we tried two other ones (mogene10sttranscriptcluster.db and mogene10stprobeset.db), which unfornutately also did not work. We think that this is due to the absence of an annotation name after we ran the norm command (Annotation:). As an alternative we used GEO2R and the R-script that was provided by the GEO website.

In [112]:

```
probe.counts <- exprs(norm)
probe.counts[1:5,1:6]
probe.counts[1:5,7:12]
```

	ShScrTet1_1	ShScrTet1_2	ShScrTet1_3	ShTet1_1	ShTet1_2	ShTet1_3
10338001	1.99052919	2.34919202	2.1710533	2.28186671	2.30253887	2.351873
10338002	-0.77160100	-0.60615778	-0.6195020	-0.56644850	-0.64758546	-0.48886
10338003	1.15479050	1.41080888	1.2916037	1.41945043	1.37512757	1.468579
10338004	0.72249768	0.84836427	0.8928700	0.94998830	0.78968078	0.944737
10338005	-0.06424546	-0.06183135	-0.0452117	-0.04654384	-0.04949394	-0.02962

	ShScrSIN3A_1	ShScrSIN3A_2	ShScrSIN3A_3	ShSIN3A_1	ShSIN3A_2	ShS
10338001	1.83160701	1.86179113	1.90970202	1.81419685	1.83701744	1.8
10338002	-0.85069020	-0.85287054	-0.85434481	-1.11730279	-1.01492390	-0.9
10338003	1.02625584	1.02988382	1.04640635	0.91211949	0.97282403	1.05
10338004	0.65879553	0.69918912	0.69744519	0.59466264	0.66179958	0.7
10338005	-0.07926623	-0.07486841	-0.05903379	-0.09694955	-0.08487366	-0.0

In [29]:

```
library(annotate)
library("pd.mogene.1.0.st.v1")
```

In [30]:

```
probe.counts <- rownames(probe.counts)
gene.symbols <- getSYMBOL(probe.counts, "pd.mogene.1.0.st.v1")
```

```
Error in lookUp(x, data, "SYMBOL"): No keys provided
Traceback:

1. getSYMBOL(probe.counts, "pd.mogene.1.0.st.v1")
2. unlist(lookUp(x, data, "SYMBOL"))
3. lookUp(x, data, "SYMBOL")
4. stop("No keys provided")
```

In [126]:

```
library("mogene10sttranscriptcluster.db")
```

```
Error in library("mogene10sttranscriptcluster.db"): there is no package call
ed ‘mogene10sttranscriptcluster.db’
Traceback:

1. library("mogene10sttranscriptcluster.db")
2. stop(txt, domain = NA)
```

In [127]:

```
library("mogene10stprobeset.db")
```

```
Error in library("mogene10stprobeset.db"): there is no package called ‘mogen
e10stprobeset.db’
Traceback:

1. library("mogene10stprobeset.db")
2. stop(txt, domain = NA)
```

In [52]:

```
library(Biobase)
library(GEOquery)
```

In [53]:

```
# Loading series and platform data from the GEO website.

gset <- getGEO("GSE24842", GSEMatrix =TRUE, AnnotGPL=TRUE)
if (length(gset) > 1) idx <- grep("GPL11078", attr(gset, "names")) else idx <- 1
gset <- gset[[idx]]
```

```
ftp://ftp.ncbi.nlm.nih.gov/geo/series/GSE24nnn/GSE24842/matrix/ (ftp://ftp.n
cbi.nlm.nih.gov/geo/series/GSE24nnn/GSE24842/matrix/)
Found 1 file(s)
GSE24842_series_matrix.txt.gz
File stored at:
/tmp/RtmpI3CQHf/GPL11078.annot.gz
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
Warning message:
In readLines(con, 1): seek on a gzfile connection returned an internal error
```

```
Warning message:
In read.table(con, sep = sep, header = header, nrows = sampleRows, : seek on
 a gzfile connection returned an internal errorWarning message:
In read.table(con, sep = sep, header = header, nrows = sampleRows, : seek on
 a gzfile connection returned an internal errorWarning message:
In read.table(file = file, header = header, sep = sep, quote = quote, : seek
 on a gzfile connection returned an internal errorWarning message:
In read.table(file = file, header = header, sep = sep, quote = quote, : not
 all columns named in 'colClasses' exist
```

In [54]:

```
# Make proper column names to match toptable
fvarLabels(gset) <- make.names(fvarLabels(gset))
```

In [55]:

```
# group names for all samples
gsms <- "000111XXXXXXXXX"
sml <- c()
for (i in 1:nchar(gsms)) { sml[i] <- substr(gsms,i,i) }
```

In [56]:

```
# eliminate samples marked as "X"
sel <- which(sml != "X")
sml <- sml[sel]
gset <- gset[ ,sel]
```

In [57]:

```
# log2 transform
ex <- exprs(gset)
qx <- as.numeric(quantile(ex, c(0., 0.25, 0.5, 0.75, 0.99, 1.0), na.rm=T))
LogC <- (qx[5] > 100) ||
          (qx[6]-qx[1] > 50 && qx[2] > 0) ||
          (qx[2] > 0 && qx[2] < 1 && qx[4] > 1 && qx[4] < 2)
if (LogC) { ex[which(ex <= 0)] <- NaN
  exprs(gset) <- log2(ex) }
```

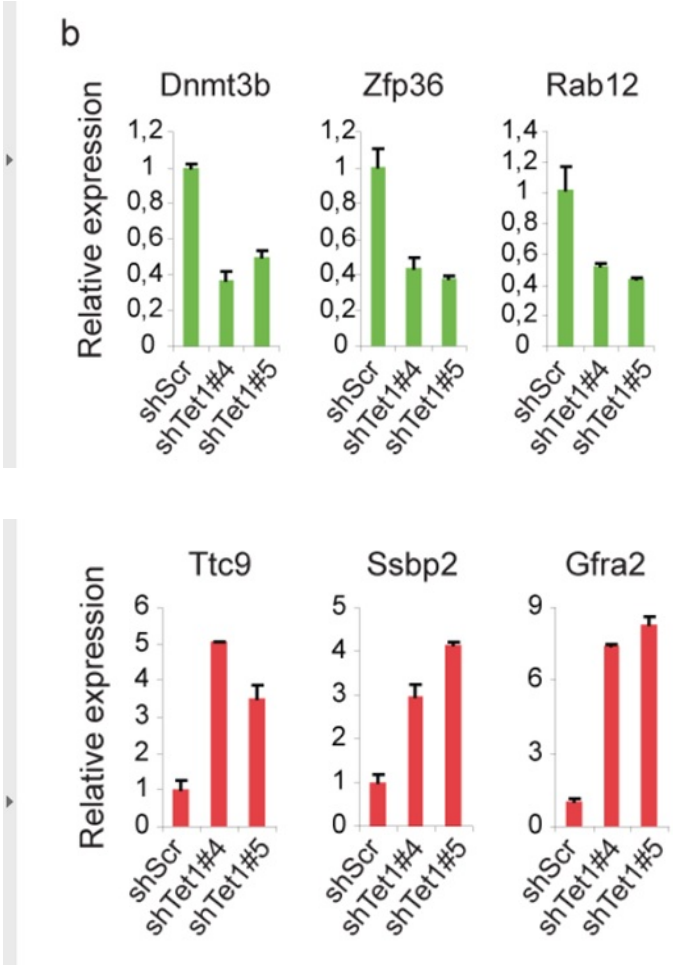
In [58]:

```
# set up the data and proceed with analysis
sml <- paste("G", sml, sep="")    # set group names
fl <- as.factor(sml)
gset$description <- fl
design <- model.matrix(~ description + 0, gset)
colnames(design) <- levels(fl)
fit <- lmFit(gset, design)
cont.matrix <- makeContrasts(G1-G0, levels=design)
fit2 <- contrasts.fit(fit, cont.matrix)
fit2 <- eBayes(fit2, 0.01)
tT <- topTable(fit2, adjust="fdr", sort.by="B", number=250)

tT <- subset(tT, select=c("ID","adj.P.Val","P.Value","t","B","logFC","Gene.symbol","Gene.ti
write.table(tT, file=stdout(), row.names=F, sep="\t")
```

As a control, the table above was also generated using the GEO2R tab on the GEO website. We first identified the groups we wanted to compared with each other (Tet1 wildtype vs Tet1 knockdown). Next, the GEO website gave us a table with probe names, gene names, log Fold Change, p-values, which we then saved as an Excel file. The logFC was ranked from low to high values and we selected the genes that changed the most and compared this with the results of the article (see printsceen below).

14	Tet1 knockdown compared to wild type Tet1					
15	ID	adj.P.Val	P.Value	t	logFC	Gene.symbol
16	52463_at	0.0000907	7.08e-09	-31.68	-1.778	Tet1
17	15109_at	0.0006755	1.50e-06	14.67	1.81	Hal
18	232345_at	0.0007586	2.24e-06	13.85	1.974	A2m
19	14586_at	0.0004567	4.95e-07	17.23	2.111	Gfra2
20	333564_at	0.0004703	6.21e-07	-16.68	-2.177	Fndc3c1
21	12798_at	0.0000907	1.28e-08	-29.1	-2.223	Cnn2
22	21825_at	0.0001186	2.24e-08	-26.88	-2.245	Thbs1
23	13436_at	0.0004323	4.18e-07	-17.66	-0.995	Dnmt3b
24	69480_at	0.0020806	1.87e-05	10.11	1.14	Ttc9
25	66970_at	0.0010986	5.06e-06	12.28	0.823	Ssbp2



As a control: Tet1 seems to be downregulated in the Tet1 knockdown cells compared to the wildtype Tet1 cells, which is correct. The three genes that are most upregulated in the Tet knockdown cells are (according to our results): Hal, A2m and Gfra2. The latter was also found to be upregulated in the article (red graph). According to the article, Ttc9 and Ssbp2 were also upregulated in Tet1 knockdown cells. However, we only found Ttc9 to be (slightly) upregulated and Ssbp2 was downregulated. Rab12 and Zfp36 were not found in our results.

In [65]:

```
#   Boxplot for selected GEO samples
library(Biobase)
library(GEOquery)
```

In [66]:

```
# load series and platform data from GEO

gset <- getGEO("GSE24842", GSEMatrix =TRUE, getGPL=FALSE)
if (length(gset) > 1) idx <- grep("GPL11078", attr(gset, "names")) else idx <- 1
gset <- gset[[idx]]
```

```
ftp://ftp.ncbi.nlm.nih.gov/geo/series/GSE24nnn/GSE24842/matrix/ (ftp://ftp.n
cbi.nlm.nih.gov/geo/series/GSE24nnn/GSE24842/matrix/)
Found 1 file(s)
GSE24842_series_matrix.txt.gz
Using locally cached version: /tmp/RtmpI3CQHf/GSE24842_series_matrix.txt.gz
```

In [68]:

```
# group names for all samples in a series
gsms <- "000111XXXXXXXXX"
sml <- c()
for (i in 1:nchar(gsms)) { sml[i] <- substr(gsms,i,i) }
sml <- paste("G", sml, sep="")
```

In [69]:

```
# eliminate samples marked as "X"
sel <- which(sml != "X")
sml <- sml[sel]
gset <- gset[ ,sel]
```

In [70]:

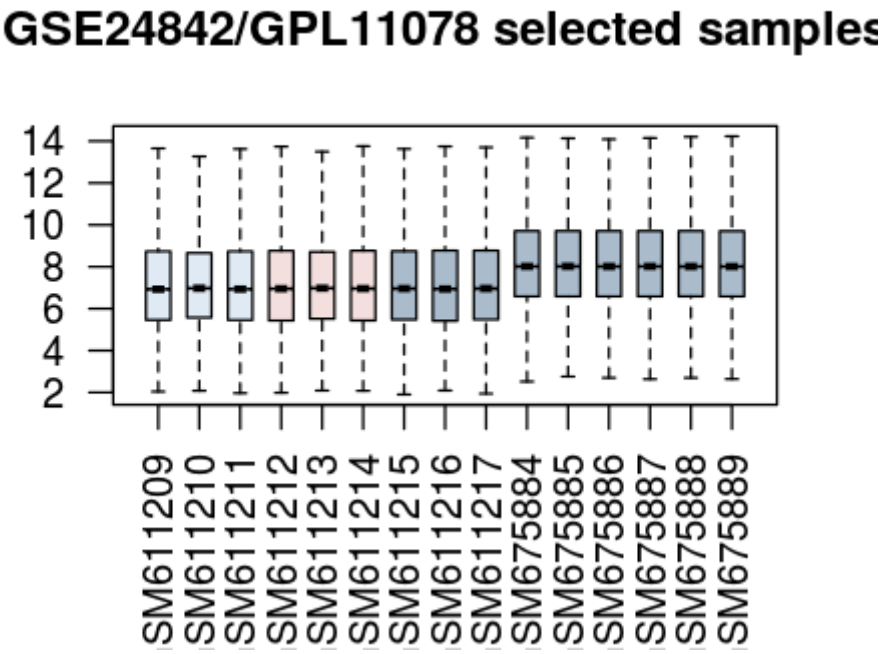
```
# order samples by group
ex <- exprs(gset)[ , order(sml)]
sml <- sml[order(sml)]
fl <- as.factor(sml)
labels <- c("ShScr","TET1_KD")
```

In [71]:

```
# set parameters and draw the plot
palette(c("#dfeaf4","#f4dfdf", "#AABBCC"))
dev.new(width=4+dim(gset)[[2]]/5, height=6)
par(mar=c(2+round(max(nchar(sampleNames(gset)))/2),4,2,1))
title <- paste ("GSE24842", '/', annotation(gset), " selected samples", sep ='')
boxplot(ex, boxwex=0.6, notch=T, main=title, outline=FALSE, las=2, col=fl)
legend("topleft", labels, fill=palette(), bty="n")
```

In [80]:

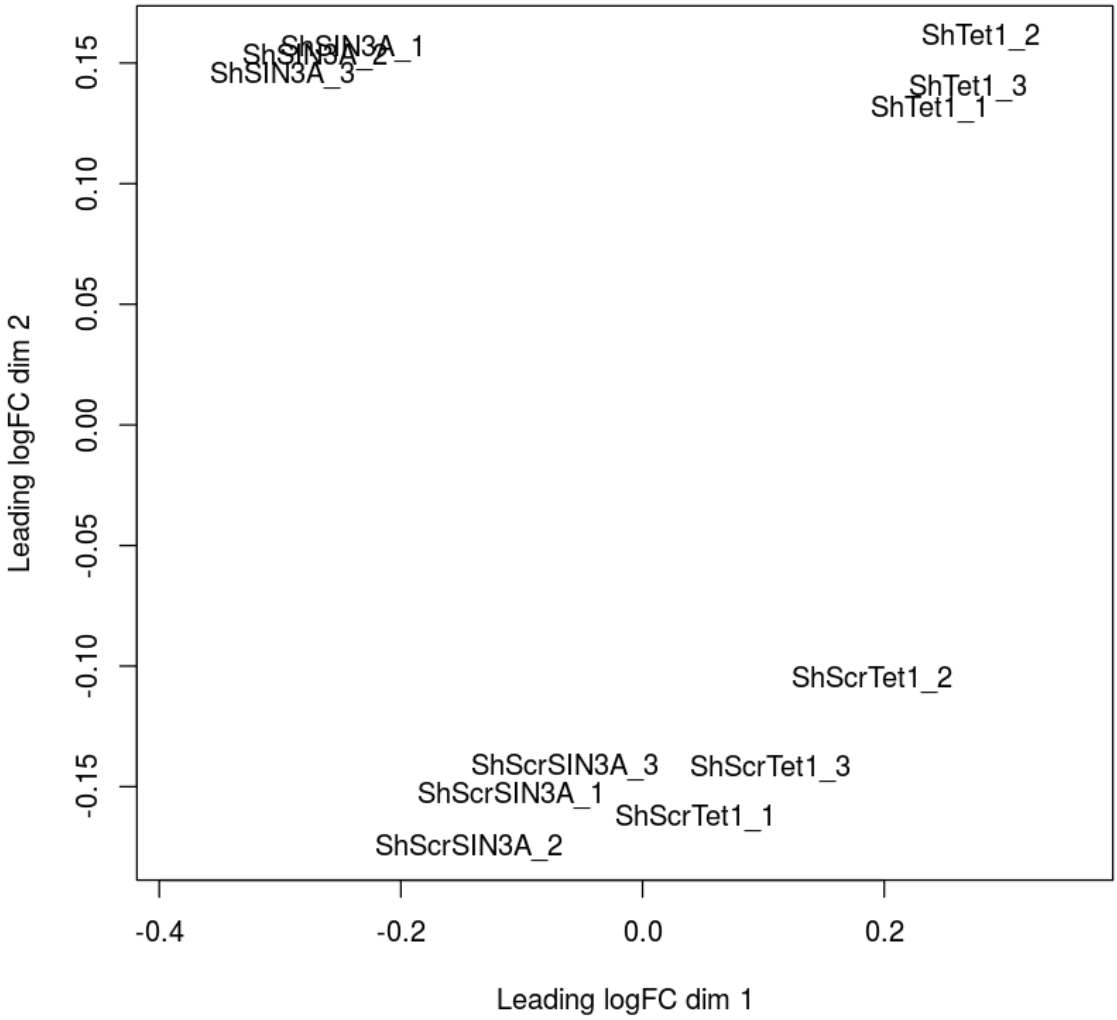
```
boxplot(ex, boxwex=0.6, notch=T, main=title, outline=FALSE, las=2, col=fl)
```



As a quality control we made a MDS plot. In this plot, replicates of the same sample cluster close together, whereas the distance between different samples is larger. For example, there is little variation between the ShSIN3A triplicate samples and there is a larger distance between the ShSIN3A and ShTet1 triplicates, which is good.

In [43]:

```
plotMDS(norm)
```

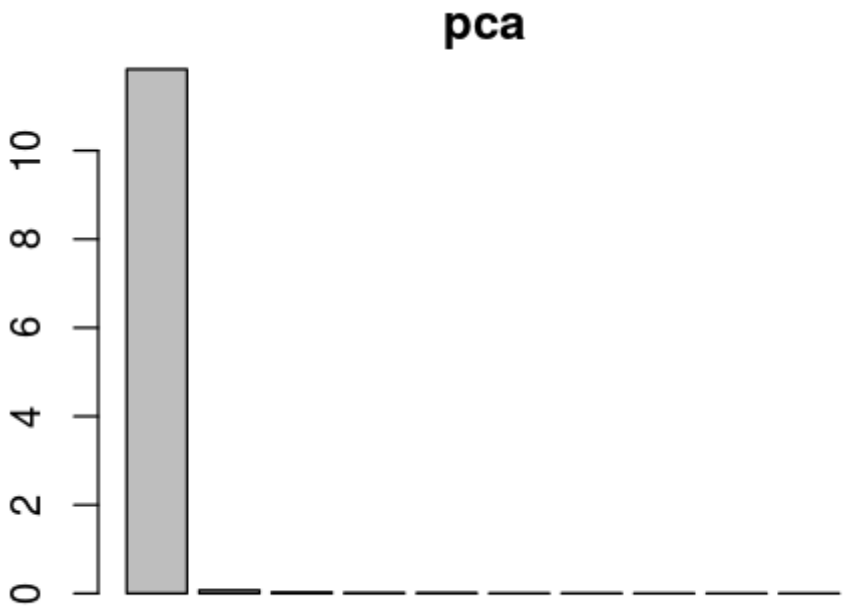


In [134]:

```
# PCA was used as an alternative for MDS.
pca =prcomp(norm, scale=T)
```

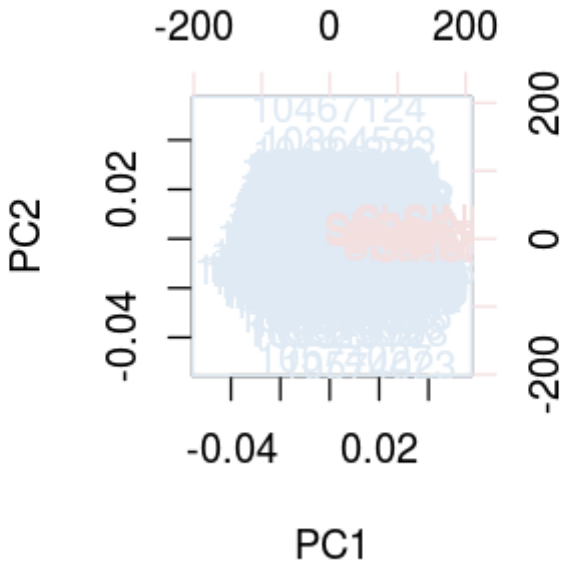
In [135]:

```
par(mar = rep(2, 4))
plot(pca)
```



In [140]:

```
pca$rotation=-pca$rotation
pca$x=-pca$x
biplot (pca)
```



Differential expression analysis

For the differential expression analysis we need the limma library which we already loaden at the beginning. First, we renamed the samples in the knockdown vector because it did not work when the samples of the same triplicate had a different name. Next, we created two conditions (scrambled (wild type) and knockdown

Tet1). Afterwards we did the same for SIN3A.

In [88]:

```
# Renaming the samples.
knockdown = c("Scrambled_Tet1",
              "Scrambled_Tet1",
              "Scrambled_Tet1",
              "Tet1 knockdown",
              "Tet1 knockdown",
              "Tet1 knockdown",
              "Scrambled_SIN3A",
              "Scrambled_SIN3A",
              "Scrambled_SIN3A",
              "SIN3A knockdown",
              "SIN3A knockdown",
              "SIN3A knockdown")
```

In [113]:

```
knockdown <- factor(c('Scrambled_Tet1', 'Scrambled_Tet1', 'Scrambled_Tet1', 'Tet1 knockdowr
                levels=c('Scrambled_Tet1', 'Tet1 knockdown'))
knockdown
```

Scrambled_Tet1 Scrambled_Tet1 Scrambled_Tet1 Tet1 knockdown
Tet1 knockdown Tet1 knockdown

In [90]:

```
design <- model.matrix(~knockdown)
design
```

	(Intercept)	knockdownTet1 knockdown
1	1	0
2	1	0
3	1	0
4	1	1
5	1	1
6	1	1

In [114]:

```
knockdown2 <- factor(c('Scrambled_SIN3A', 'Scrambled_SIN3A', 'Scrambled_SIN3A', 'SIN3A knoc
                levels=c('Scrambled_SIN3A', 'SIN3A knockdown'))
knockdown2
```

Scrambled_SIN3A Scrambled_SIN3A Scrambled_SIN3A SIN3A knockdown
SIN3A knockdown SIN3A knockdown

In [119]:

```
design2 <- model.matrix(~knockdown2)
design2
```

	(Intercept)	knockdown2SIN3A knockdown
1	1	0
2	1	0
3	1	0
4	1	1
5	1	1
6	1	1

In [143]:

```
?lmFit
```

In [144]:

```
lmFit(norm, design=NULL, ndups=1, spacing=1, method="ls")
```

In [146]:

```
fit <- lmFit(design)
```

In [141]:

```
fit <- eBayes(lmFit(design))
```

In [147]:

```
fit <- eBayes(fit)
```

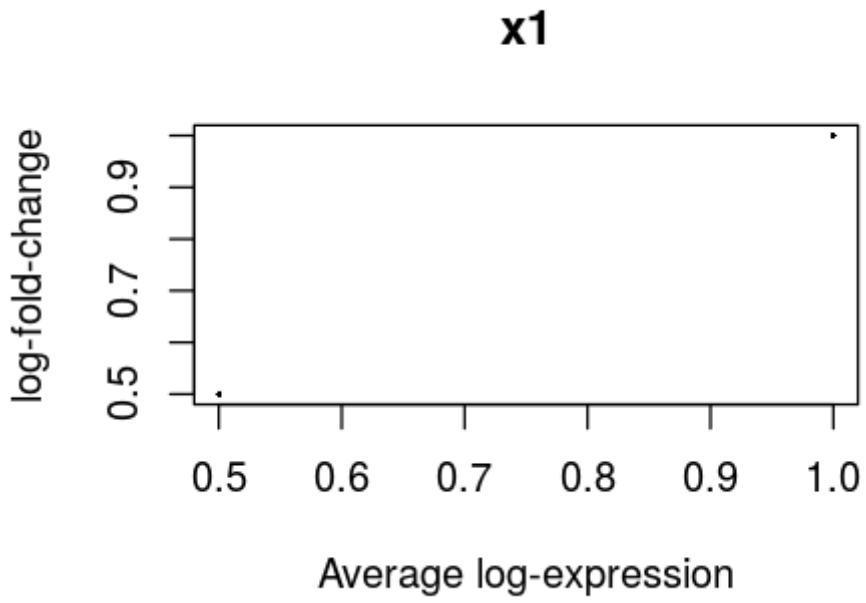
In [150]:

```
topTable(fit)
```

	logFC	AveExpr	t	P.Value	adj.P.Val	B
4	1.0	1.0	479.193216	0.000199308	0.0003986159	-0.4128942
5	1.0	1.0	479.193216	0.000199308	0.0003986159	-0.4128942
6	1.0	1.0	479.193216	0.000199308	0.0003986159	-0.4128942
1	0.5	0.5	1.150032	0.416811275	0.4168112755	-6.9571920
2	0.5	0.5	1.150032	0.416811275	0.4168112755	-6.9571920
3	0.5	0.5	1.150032	0.416811275	0.4168112755	-6.9571920

In [153]:

```
limma::plotMA(fit, 1)
abline(0,0,col='red')
```



Unfortunately, we were not able to make a MA plot and to correct for multiple testing.