Microarray_Analysis_Part_1

Hasani Perera

26/11/2021

```
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install(c("GEOquery","oligo","pd.hugene.1.0.st.v1",
                       "hugene10sttranscriptcluster.db"))
install.packages("ggplot2")
```

install packages

Query Raw Data load packages - GEO Accession No: GSE50397

```
library(knitr)
library(ggplot2)
library(GEOquery)
```

extract .cell files to local machine - GEO Series records (GSExxxxx)

```
gse <- getGEO("GSE50397",GSEMatrix=FALSE)
```

view information
- names of all the GSM objects contained in the GSE

```
print(names(GSMList(gse)))
```

first GSM object on the list

```
class(GSMList(gse)[[1]])
head(Meta(GSMList(gse)[[1]]))
```

names of the GPLs represented

```
names(GPLList(gse))
```

access raw data (downloaded file paths)

```
file_paths = getGEOSuppFiles("GSE50397")
head(file_paths)
```

choose tar file

```
tarfile <- file.choose()
```

extract tar archives

```
untar(tarfile, exdir="RawData_Files")
```

list of all gz files in the directory

```
cel.files <- list.files("RawData_Files/", pattern = "[gz]")
length(cel.files)
```

list/vector/dataframe —> vector/matrix extract gz archives - gunzip

```
sapply(paste("RawData_Files", cel.files, sep="/"), gunzip)
```

Data Preprocessing load packages

```
library(oligo)
```

list of all cel files in the directory

```
celpath <- "~/Documents/Learning/Bioinformatics/MicroArray Analysis/Nordic Islet Analysis/Raw_Data"
celfiles_list <- list.files(celpath,pattern = ".CEL", full.names=TRUE)
length(celfiles_list)
head(celfiles_list)
```

import CEL files containing raw probe-level data into an R AffyBatch object

```
cell_files <- read.celfiles(celfiles_list)
head(cell_files)
```

load packages

```
library(pd.hugene.1.0.st.v1)
??pd.hugene.1.0.st.v1
```

```
getClass("GeneFeatureSet")
```

max expression

```
max(exprs(cell_files[1:1102489,1:89]))
```

replace sampleNames

```
filename <- sampleNames(cell_files)
pData(cell_files)$filename <- filename
pData(cell_files)$filename
sampleNames <- sub("-islet.CEL$","",filename)
sampleNames <- sub("_HTL[[:digit:]]*","",sampleNames)
sampleNames(cell_files) <- sampleNames
sampleNames(cell_files)
```

information on variable values/ meta-data

```
pData(cell_files)
```

boxplot before RMA normalization

```
boxplot(cell_files, target="probeset")
mtext(text="log2 Intensity", side=2, line=3, las=0)
```

histogram before RMA normalization

```
hist(cell_files,target="probeset")
```

```
mtext(text="Samples", side=1, line=3, las=1)
```

perform RMA normalization (Robust Multi-Array Average) - converts an AffyBatch object into an ExpressionSet object

```
normData <- rma(cell_files)
nrow(normData)
```

boxplot after RMA normalization

```
boxplot(normData, target="probeset")
mtext(text="log2 Intensity", side=2, line=3, las=0)
mtext(text="Samples", side=1, line=3, las=1)
```

histogram after RMA normalization

```
hist(normData,target="probeset")
```

save the expression data (output - normalized and log2 transformed)

```
exprs(normData)[1:3,1:5]
write.exprs(normData,file="RMA_Normalised_Original.txt")
```

load packages

```
library(Biobase)
library(hugene10sttranscriptcluster.db)

??Biobase
??hugene10sttranscriptcluster.db
```

gene annotation - get a list of retrievable data

```
keytypes(hugene10sttranscriptcluster.db)
```

retrieve data for selected objects (ENTREZID and SYMBOL) as a data frame

```
anno<- select(hugene10sttranscriptcluster.db,keys(hugene10sttranscriptcluster.db),
              c("ENTREZID", "SYMBOL"))
head(anno)
tail(anno)
```

optional - to keep one match per gene

```
anno <- anno[!duplicated(anno[,1]),]
tail(anno)
```

set row names to ProbeID (for convenience)

```
anno = anno[,-1]
row.names(anno) = keys(hugene10sttranscriptcluster.db)
tail(anno)
```

retrieve gene expression matrix from normData as a dataframe

```
expr <- data.frame(exprs(normData))
head(expr)
```

merge gene expression and annotation according to row names (probe IDs)

```
expr_anno <- merge(x=anno,y=expr,by.y=0, by.x=2,all=TRUE)
head(expr_anno)
```

save the annotated gene expression matrix to local file

```
write.table(expr_anno, file = "RMA_Norm_Expr.txt",sep = "\t",
            row.names = FALSE, col.names = TRUE, quote =FALSE)
write.csv(expr_anno, file = "RMA_Norm_Expr_Anno.csv")
```