coexnet: An R package to build CO-EXpression NETworks from Microarray Data

Juan David Henao

2017-02-07

Contents

1 Abstract

2 Introduction

3 Workflow

3.1 get.info

All microarray raw data associated with the same study is stored in a CEL file, this file containe the GSM files, each one corresponding to each sample inside of the study, you can obtain each GSM file in a individual way, but, is rather obtain all the samples in the CEL file to avoid the work to join each GSM and additionally the way to analyse this kind of data is simultainously (in future normalization process for example). Therefore, all the micrroarray chips are documented in the GEO Datasets database. Each one of them are identify with the letters GPL follows of a unique number. The information in the GPL file is related with the information of every probeset in the microarray chip, including the gene, funtion, type and another information to take advantage of the analysis of the results.

This function will create in your actual pathfile a folder with the GSE(unique number) name where are stored the GSM download files and otherwise the file GPL(unique number).soft file with the information of the microaaray chip.

```
# Downloading the microarray raw data from GSE8216 study
# The accession number of the microarray chip related with this study is GPL2025

get.info(GSE = "GSE8216", GPL = "GPL2025",dir = ".")
```

```
# Show the actual pathfile with the folder with the GSE number and the .soft file

dir()
```

Take account

In some cases the information in the GPL file are partial, take this in mind in the future analysis and is recomended not store the files in a temporal folder due to in many cases you will need the raw data to re-process the expression values using different methods.

3.2 get.affy

The AffyBatch object is one of the most widely kind of data used to process and analyse microarray expression data. The AffyBatch object store information about the data of scan to each one of the samples, the information related with the phenotype, the raw expression values to each probe in the microarray chip, the kind of library to read the expression data and anther one.

You can use the AffyBatch object in many different packages mainly in the affy package, additionally you can modify the AffyBatch object if you consider it necessary.

This function search in your actual or designaded pathfile the folder with the GSE accession number and read the filelist.txt file with the name of the every GSM samples to recognize them and join in an only AffyBatch object.

```
# Reading some GSM samples from GSE4773 study, the folder with the
# GSM files are called GSE1234.

affy <- get.affy(GSE = "GSE1234",dir = system.file("extdata",package = "coexnet"))
```

Take account

In some cases the AffyBatch doesn't have all the information and a warining messege is shown when you view the variable with the AffyBatch object, but you can edit the AffyBatch to fill all the requiered information. If you try to process the AffyBatch in some of the packages that use this kind of object you will recieved an error messege.

```
# The variable affy doesn't have the CDF (Chip Definition File) information.
# You can include this information modifying the AffyBatch object.

affy@cdfName <- "HG-U133_Plus_2"
```

3.3 gene.symbol

In most cases, the idea of create a co-expression network is to visualize the relations among diferent genes, proteins, specific DNA or RNA fragments or another kind of molecular entity identify with a specific ID. For this reason, is very useful to have the information about the corresponding ID to each one of the probeset in the microarray. This kind of information will use when you need pass from a matrix of probeset-sample to one of gene(or another ID)-sample before of the construction of the co-expression network.

The .soft file, downloaded from GEO Datasets database using the GPL identifier have the information to create a table with the relationship between probeset and one molecular ID, in this table one ID can be related with two or more probeset, the process to create only one expression values to one ID from different probeset is called *summarization* (see bellow).

This function search in the actual or the designaded pathfile the .soft file and from this searchand create a data.frame where the first column have each one of the probeset names and in the second one have the corrisponding ID (gene symbol, protein name or symbol, etc).

```
# Create the table with the relationship between probesets and IDs.

gene_table <- gene.symbol(GPL = "GPL2025",d = system.file("extdata",package = "coexnet"))

head(gene_table)
```

Take account

In some cases, the .soft file dosen't have all the IDs to each one of the probeset inside of the microarray, you can ignore this probeset under the assumption of the another probeset can have the ID and correspond at the no identify probeset. On the other hand, one ID can be related with more than two names, this function create a ID with all the related names separed by "-", that is useful in future analysis when the biological information is related at one specific name among the several names to one ID.

```
# The before table have NA and empty information in the IDs.
# We can delete this unuseful information.

# Deletion of IDs with NA information

gene_na <- na.omit(gene_table)

# Deletion of empty IDs

final_table <- gene_na[gene_na$ID != "",]

head(final_table)
```

3.4 exprs.mat

Take account

3.5 cof.var

In some cases, the co-expression network is built from two or more microarrays studies, in this sense, is necessary to define wich of this studies represent the most source of background noise and probably affect in a negative way the future results. One way to determinate the most harmful studies is from variation analysis, the study with more variation among the normalized expression values can be the source of the future background noise and is necessary to consider the use of this studies in the construction of the co-expression network.

To define the variation among the normalized expression values can be determinated the *coefficient of variation* of each one of gene or ID in each one of the studies and generate the boxplot from the results. So, in a graphical way is possible to define the studies that will generate background noise watching the atipic data. On the other hand, is posible to define the number of atipic data and determinate the more variant studies using the boxplot and the number of atipic data defing a threshold value, for example the studies with more of 10% of atipic data wont be take account in the construction of the co-expression network.

This function take the normalized ID-sample matrix and claculate the median and the coefficient of variation to each one of the ID, this process is neccesary to do study-by-study. Additionally, this function allow to calculate the median and the coefficient of variation to cases and controls samples separately using a vector of 0s and 1s to identify the case and control samples. This vector is possible be defined in the description of

each sample in the GEO Datasets database and is necessary no identify the gene (or another ID as proteins) differentially expressed (see bellow).

```
# Simulated expression data

n <- 200
m <- 20

# The vector with treatment samples and control samples

t <- c(rep(0,10),rep(1,10))

# Calculating the expression values normalized

mat <- as.matrix(rexp(n, rate = 1))
norm <- t(apply(mat, 1, function(nm) rnorm(m, mean=nm, sd=1)))

# Calculating the coefficient of variation to case samples

case <- cof.var(data = norm,complete = FALSE,treatment = t,type = "case")
head(case)

# Creating the boxplot to coefficient of variation results

boxplot(case$cv)

# Extracting the number of atipic data

length(boxplot.stats(case$cv)$out)
```

Take account

The desition of delete a microarray study from the result of coefficient of variation result depend of the data and the criteria of the researcher to filter the studies (the selection of a threshold value), don't exist a gold rule to discard a study, is advisable calculate the coefficient of variation of all samples at time to compare and determinate the most variant.

```
# Calculating the coefficient of variation to whole matrix

complete <- cof.var(norm)
head(complete)

# Creating the boxplot to coefficient of variation results

boxplot(complete$cv)

# Extracting the number of atipic data

length(boxplot.stats(complete$cv)$out)
```