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

Juan David Henao

2017-02-14

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

The raw expression data in a microarray experiment mut be processed to transform the original dat in the correct way to be analyzed and obtain confidence results. The first step is the normalizaion of the data, that consist in a background correction of the raw data and the follow transformation of the result in a normalize way. The second one is the pass of probeset to gene or any other ID to represent the molecular entity to analyze. Additionally, exist the posibility to make a second kind of background correction based on the batch of sampes scaned in a separete way due to the large number of samples and the limitation in the size of the microarray chip used, this correction is knowledge as *Bactch Effect Correction*.

Exist different methods to normalize raw expression data from microrray experiments, each one of this methodologies consider a way to generate a background correction, the process of normalization and the pass from probes to probesets. The differencies of this methods consist in the underliying mathematics assumptions used and the range of the noralized reults, in som cases the expression data with a range more wide than anothers. In the same way, the process to transform the probeset-samples matrix in a gene(or another ID)-sample matrix consider diferent methodologies to do that including the obtantion of the average of the expression values of each one of the probeset corresponding to the same gene or protein, the selection of the maximum or minimum value, among others.

This function have the possibility to choose among two different methods to normalize the raw expression values, including the process of background correction and the pass from probes to probeset. The first one is *rma* (Robust Multi-Array Average), this method do a background correction and normalization in a separate calculations (Irizarry,*et al.* 2003). The second one is *vsn* (Variance Stabilizing Normalization), this method, contrary to *rma*, generate the background correction and the normalization in the same equation (Huber, *et al.* 2002). On the other hand, the function have the option to do a *Batch Effect Correction* wheter you know

if the samples were scaned in a separate way using the *Scan Date* data inside the AffyBatch object (one of the input in the function).

Additionally, this method consider two ways to calculate the values in the process to pass from probesets to gene/ID. The first one is selecting a representative probeset to each one of the gene, protein or another kind of ID. To do that, is calculated the average of each one of the probeset associated with the same gene/ID, and the probeset with the most high value in the average is selected. The second one is obtain the median of the each one of the samples to the probesets associated with the same gene/ID, obtained a only one expression value for sample as the transformation of the normalized data.

```
# Loading AffyBatch object

data("affy")

# Loading table with probeset and gene/ID information

data("info")

# Calculating the expression matrix witj rma

rma <- expr.mat(affy = affy,genes = info,NormalizeMethod = "rma",
SummaryMethod = "median",BatchCorrect = FALSE)
head(rma)
```

Take account

Consider that *rma* is a method whose results have less range of amplitude than *vsn*, take account this to select the method to normalize the raw expresion values. In some cases. The method *vsn* take account every probes in the normalize process, that could takes time. In some cases is a good idea prove the normalization using and no using the *Batch Effect Correction*.

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
```

```r
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.

```r
# 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)
```

3.6 dif.exprs

3.7 find.threshold

When you have the final expression matrix, is necessary analyze that in a way to obtain the co-expression network. Exist two methods widely used to do this process, both of them are related with the definition of correlation value between all the genes/IDs creating a square matrix. In one hand you can calculate the *Pearson Correlation Coefficient*, this method calculate the correlation between every genes/IDs from the

expression values of each one, as result, the square matrix will have values between zero and one, due to in the future construction of co-expression network is necessary to use the absolute value of the results. On the other hand, the *Mutual Information* is calculated in a same way in the aforementioned method, but, in this case, is necessary an additional transformation of the results to obtain a square matrix with values between zero and one.

The obtaintion of a square matrix with a range of values from zero to one is necessary to the future transformation of this correlation matrix to the adjacency matrix (see bellow). Additionally, is necessary this range of values due to a threshold value must be defined to establish the final relations between the genes/IDs, to do this, a value among zero to one is difened and the values of correlation less than threshold value will indicate the no existance of a real correlation among them, obtianed finally the relations among the genes/IDs as co-expression network.

This function calculate a threshold value using a novel method based in two *Biological Systems* approachs. First, each possible threshold value, from 0.01 to 0.99 with an increase of 0.01. Each one of this possible threshold values is then analyzed in the first approche aforementioned mentioned. To each value, is calculated the *Clustering Coefficient* to the network creating using the actual threshold value under test. After, is calculated an artificial *Clustering Coefficient* to simulate a random network created using the same threshold value. Then, is calculated the difference among the two *Clustering Coefficient* values, and the result that do acomplishment of the criteria of Elo, *et al* (2008), will be take account to the next analysis. Finally, the remain threshold values will be analyze using the *Degree Distribution* under normal distribution using *Kolgomorov-Smirnov* test, expecting the resulting p-value rejecting the distribution, that as result of the assumption that the biological networks dosen't have normal distribution when the *Degree Distribution* is analyzed. Finally, the minimum threshold value that comply this two criteria will be selected as the final threshold value to the construction of the co-expression network.

```
# Loading data

pathfile <- system.file("extdata","expression_example.txt",package = "coexnet")
data <- read.table(pathfile,stringsAsFactors = FALSE)

# Find threshold value

cor_pearson <- find.threshold(difexp = data,method = "correlation")
cor_pearson
```

Take account

3.8 create.net

The last step in the construction of a co-expression matrix is the construction of the network. Once you had defined a threshold value, the last step is transform the expression value in a adjacency matrix using the correlation or the mutual information method to obtain the values of relationship between the diferent genes or proteins (or another kind of ID).

The process to pass from expression matrix to network consist of two steps. The first one is the build of a correlation matrix, to do that is necessary to apply some of the two method to calculate the relation among the genes/ID (Pearson correlation or mutual information, see above) (López-Kleine, *et al.* 2013). The second one is the pass from expression matrix to adjacency matrix, to create this kind of matrix is necessary the threshold value, every correlation value inside the matrix less than the threshold will be replace by zero, while all remainded values will be replaced by one. Aditionally, the product of related one gene/ID with itself will be always zero.

Finally, from the adjacency matrix, is created a list where one gene/ID and another one conected to the first are related separated by a space. It means, if *gene A* and the *gene B*, in the adjacency matrix have a value of one, then, in the final adjacency matrix they appear as:

gene A – gene B

and so on to the every genes/IDs connected in the final co-expression matrix.

This function takes the expression matrix and create the correlation matrix using or *Pearson Correlation Coefficient* or *Mutual Informtion*. After that, create the adjacency matrix taking account the threshold vlue given by the user and finally create the network from the adjacency list as a igraph object to be analyze using the igraph R package or another one that recognize this kind of object.

```
# Loading data

pathfile <- system.file("extdata","expression_example.txt",package = "coexnet")
data <- read.table(pathfile,stringsAsFactors = FALSE)

# Building the network

cor_pearson <- create.net(difexp = data,threshold = 0.7,method = "correlation")
head(cor_pearson)
```

Take account

In the process of construction of adjacency matrix, some gene/ID could not be related with another one and in the process to pass from matrix to adjacency list, this gene/ID will be deleted. Additionally, the network in the igraph object can be export as adjacency list using the igraph package to viualize the network in another program as *Cytoscape* or *Gephi*.

4 References

- Elo, L. L., Järvenpää, H., Orešič, M., Lahesmaa, R., & Aittokallio, T. (2007). Systematic construction of gene coexpression networks with applications to human T helper cell differentiation process. Bioinformatics, 23(16), 2096-2103.
- Huber, W., Von Heydebreck, A., Sultmann, H., Poustka, A., & Vingron, M. (2002). Variance stabilization applied to microarray data calibration and to the quantification of differential expression. Bioinformatics, 18(suppl 1), S96-S104.
- Irizarry, R. A., Hobbs, B., Collin, F., Beazer Barclay, Y. D., Antonellis, K. J., Scherf, U., & Speed, T. P. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. Biostatistics, 4(2), 249-264.
- López-Kleine, L., Leal, L., & López, C. (2013). Biostatistical approaches for the reconstruction of gene co-expression networks based on transcriptomic data. Briefings in functional genomics, elt003.