Package: maCorrPlot (via r-universe)

November 25, 2026

Title Visualize artificial correlation in microarray data

Version 1.84.0

Author Alexander Ploner <Alexander.Ploner@ki.se>

Description Graphically displays correlation in microarray data that is due to insufficient normalization

Maintainer Alexander Ploner <Alexander.Ploner@ki.se>

Depends lattice

Imports graphics, grDevices, lattice, stats

License GPL (>= 2)

biocViews Microarray, Preprocessing, Visualization

URL http://www.pubmedcentral.gov/articlerender.fcgi?tool=pubmed&pubmedid=15799785

Repository Bioconductor 3.24

Date/Publication 2026-11-23 22:28:33 UTC

RemoteUrl https://github.com/bioc/maCorrPlot

RemoteRef RELEASE_3_24

RemoteSha 771ddd568ce44eedff5d7292f0c5d24fcb2faee5

Contents

CorrSample 2
CutCI 3
oligodata 5
plot.corr.sample 5

Index **9**

CorrSample *Sample correlations for random pairs of genes*

Description

CorrSample calculates the correlations, standard deviations and some auxiliary variables for random pairs of genes. A plot of the resulting object that shows that these correlations dependend systematically on the genes' variability, suggests a lack of normalization.

RandPairs is a helper function for generating random pairs from a list of genes.

Usage

```
CorrSample(x, np, seed, rp, ndx)

RandPairs(probes, number)
```

Arguments

x	a gene expression matrix, with samples as columns and genes as rows; missing values are accepted.
np, number	the number of random pairs
seed	an optional seed for the random sampling
rp	an optional matrix with two columns specifying the random pairs, see Details.
ndx	an optional logical matrix of the same dimension as x that allows to eliminate a subset of the expression values from the calculation of the correlations, standard deviations and auxiliary variables.
probes	a vector of genes from which to draw random pairs; can be integer, as a vector of row indices, or character, as a vector of row names.

Details

The sample of random pairs can be specified in a replicable manner either via np and seed, or by using the output from RandPairs for the parameter rp. In case we want to use the same set of random pairs (e.g. when comparing different expression measures on the same data set), the second option will be faster.

Value

An object of class corr.sample; this is just a data frame with an extra class tag to allow for a plotting method.

The data frame has np rows and nine columns:

Correlation	the correlation between the two genes across samples
StdDev	the geometric mean of the standard deviations of the two genes
sd1, sd2	the standard deviations of the genes

m1, m2	the means of the genes
ndx1, ndx2	the indices of the two genes; by default, these will be the corresponding row indices of x, but if rp is specified, they might be gene names.

Author(s)

Alexander Ploner <Alexander.Ploner@ki.se>

References

Ploner A, Miller LD, Hall P, Bergh J, Pawitan Y. Correlation test to assess low-level processing of high-density oligonucleotide microarray data. BMC Bioinformatics, 2005, 6(1):80 http://www.pubmedcentral.gov/articlerender.fcgi?tool=pubmed&pubmedid=15799785

See Also

plot.corr.sample

Examples

```
# Get small example data
data(oligodata)
dim(datA.rma)

# Compute the correlations for 500 random pairs,
# that is ca. 1/1000 of all possible pairs
# Larger numbers are reasonable for larger data sets
cs1 = CorrSample(datA.rma, 500, seed=210)
cs1[1:5,]

# Clear correlation for pairs of genes with low average variability
plot(cs1)

# A different way of specifying the same
set.seed(210)
rp = RandPairs(rownames(datA.rma), 500)
cs2 = CorrSample(datA.rma, rp=rp)
cs2[1:5,]
plot(cs2)
```

CutCI *Calculate confidence intervals for grouped values*

Description

CutCI groups values of one variable into intervals with the same number of observations each and computes confidence intervals for the mean of another variable in each interval.

CIrho computes the normal theory confidence interval for a vector of values.

Usage

```
CutCI(dat, number = 10, func = mean, alpha=0.95)

CIrho(rho, alpha = 0.95)
```

Arguments

dat	a numerical data frame or matrix with two columns, the first of which gets averaged, and the second of which defines the grouping
number	the number of equal-count intervals
func	summary function for computing the mean
rho	a vector of measurements
alpha	the desired confidence level

Details

The quantiles for the confidence interval are taken from the standard normal distribution, so a reasonable number of observations per interval would be good.

Value

CutCI returns invisibly a list of length three:

x	the midpoints of the grouping intervals
y	the means within each interval, as computed by func
yci	a matrix with two columns, giving the lower and upper end of the confidence interval respectively

CIrho returns a vector of length two, containing the lower and upper end of the confidence interval.

See Also

co.intervals

Examples

```
x = rnorm(100, mean=2)
CIrho(x)

y = 2 + 3*x + rnorm(100)
cc = CutCI(cbind(x,y), number=5)
print(cc)

# Show it
plot(cc$x, cc$y)
arrows(cc$x, cc$yci[,1], cc$x, cc$yci[,2], length=0)
```

oligodata *Example data for package maCorrSample*

Description

Example expression data to demonstrate the functionality of the package: two data sources A and B with 30 patients and 1000 genes each, for each of which we have RMA expression values, (logarithmized) MAS5 expression values, and MAS5 absent/present calls.

Correspondingly, we have six data matrices whose name are constructed as `dat[A|B].[rma|mas5|amp]`.

Usage

```
data(oligodata)
```

Format

All matrices have genes as rows and samples as columns.

Source

These are small anonymized excerpts from a real breast cancer data set.

See Also

`CorrSample`, `plot.corr.sample`

Examples

```
data(oligodata)
str(datA.rma)
str(datB.rma)
str(datA.mas5)
str(datB.mas5)
str(datA.amp)
str(datB.amp)
```

plot.corr.sample *Plot correlation of random pairs of genes*

Description

`plot.corr.sample` provides the main functionality of package `maCorrPlot`: it plots the correlation of random pairs of genes against their variability. Systematic deviations of the plot from a constant zero indicate lack of normalization of the underlying expression matrix.

Formally, `plot.corr.sample` is the plotting method for objects of class `corr.sample` generated by `CorrSample`.

`panel.corr.sample` is the panel function that does the actual plotting work.

Usage

```
plot.corr.sample(x, ..., cond, groups, grid = TRUE, refline = TRUE, xlog = TRUE,
                 scatter = FALSE, curve = FALSE, ci = TRUE, nint = 10,
     alpha=0.95, length = 0.1, xlab="Standard Deviation")

panel.corr.sample(x, y, grid = TRUE, refline = TRUE, xlog = TRUE,
                  scatter = FALSE, curve = FALSE, ci = TRUE, nint = 10,
      alpha=0.95, length = 0.1, col.line, col.symbol, ...)
```

Arguments

x, y	for `plot.corr.sample`, x is an object of class `corr.sample`, generated by function `CorrSample` that contains the pre-computed correlations and standard deviations for the random pairs of genes; for `panel.corr.sample`, x and y are the x- and y-components (or standard deviation and correlation) of the pairs of genes to be plotted in a specific panel.
...	either more objects of class `corr.sample` or plotting arguments passed to the underlying `xyplot`.
cond	either a vector or a list of vectors describing multiple objects of class `corr.sample`; ignored if only one such object (x) is specified. See Details and Examples.
groups	a vector or a list of vectors giving group membership for the random pairs of genes in the `corr.sample` objects to be plotted, resulting in multiple overlayed plots for each object. See Details and Examples.
grid	logical value indicating whether to draw a reference grid
refline	logical value indicaitng whether to draw a horizontal reference line a zero.
xlog	logical value indicating whether to use log-scale on the horizontal axis.
scatter	logical value indicaitng whether the plot the individual pairwise correlations.
curve	logical value indicating whether to fit a simple model for lack of fit to the correlations.
ci	logical value indicating whether to add confidence intervals.
nint	number of intervals into which to divide the horizontal axis for calculating average correlations.
alpha	the level of confidence to be plotted.
length	the length of the horizontal ticks indicating the ends of the confidence intervals (in inches).
xlab	the label for the horizontal axis.
col.line, col.symbol	graphical parameters that control the color of the correlation lines and the scatter plotting symbols

Details

The underlying plotting engine is `xyplot`, using `panel.corr.sample` as panel function, which also interprets most of the graphical parameters. Note that two kinds of arguments can be specified via

...: First, an unlimited number of extra corr.sample objects, in case we want to display different expression measures for the same expression matrix, or compare different expression matrices, or both; this is somewhat similar to the behaviour of boxplot.default. Second, everything that does not inherit from corr.sample is passed on to xyplot, so in theory, the full range of lattice control options is available, as long as they do not conflixt with named arguments to plot.corr.sample, like xlog or xlab.

Two mechanisms for comparisons within the same plot are available: First, as mentioned above, multiple corr.sample objects can be shown in the same graph, each within its own panel. If no cond is specified, these panels are just numbered in the order in which the objects appear in the arguments. Alternatively, one or two factors can be associated with each factor: in the first case, cond is just a vector with as many entries as corr.sample objects in the argument list; these entries are used to label the panels of the corresponding corr.sample objects. In the second case, cond is a list with two such vectors, and the objects are cross-classified according to both categories, and the panels are arranged in a row-column pattern reflecting this cross-classification, see Examples.

The other mechanism for graphical comparisons within the same plot is via groups, which draws different correlation curves for different sub-groups of pairs of genes; the standard example is to classify pairs of genes according to their common or average score in regard to a quality control measure like the MAS5 presence calls, see Examples. These sub-groups are specified via groups; if there is only one corr.sample object in the function call (x), groups is just a vector with as many entries as there are random paris of genes in x. If several objects of class corr.sample have been specified in the function call, groups is a list of as many vectors as objects, where each vector has as many entries as the corresponding object has pairs of genes.

Value

A plot created by xyplot.

Warning

cond is translated into conditioning variables for xyplot, which will not hesitate to average correlations across different corr.sample objects. It's hard to see when this would be a good idea, therefore plot.corr.sample will generate a warning.

Author(s)

Alexander Ploner <Alexander.Ploner@ki.se>

References

Ploner A, Miller LD, Hall P, Bergh J, Pawitan Y. Correlation test to assess low-level processing of high-density oligonucleotide microarray data. BMC Bioinformatics, 2005, 6(1):80 http://www.pubmedcentral.gov/articlerender.fcgi?tool=pubmed&pubmedid=15799785

See Also

CorrSample, xyplot

Examples

```
# Get small example data
data(oligodata)
dim(datA.rma)
dim(datB.rma)

# Compute the correlations for 500 random pairs,
# Larger numbers are reasonable for larger data sets
cs1.rma = CorrSample(datA.rma, 500, seed=210)
plot(cs1.rma)

# Change the plot
plot(cs1.rma, scatter=TRUE, curve=TRUE, alpha=0.99)

# Compare with MAS5 values for the same data set
cs1.mas5 = CorrSample(datA.mas5, 500, seed=210)
plot(cs1.rma, cs1.mas5, cond=c("RMA","MAS5"))

# We group pairs of gene by their average number of MAS5 present calls
pcntA = rowSums(datA.amp[cs1.mas5$ndx1, ]=="P") +
        rowSums(datA.amp[cs1.mas5$ndx2, ]=="P")
hist(pcntA)
pgrpA = cut(pcntA, c(0, 20, 40, 60), include.lowest=TRUE)
table(pgrpA)

# Plot the RMA values according to their MAS5 status
# The artificial correlation is due to gene pairs with few present calls
plot(cs1.rma, groups=pgrpA, nint=5, auto.key=TRUE, ylim=c(-0.3, 0.5))

# Combine grouping and multiple conditions
plot(cs1.rma, cs1.mas5, cond=c("RMA","MAS5"), groups=list(pgrpA, pgrpA),
     nint=5, auto.key=TRUE, ylim=c(-0.3, 0.5))

# Compare with second data set
# Specify more than one condition
cs2.rma  = CorrSample(datB.rma, 500, seed=391)
cs2.mas5 = CorrSample(datB.mas5, 500, seed=391)
plot(cs1.rma, cs1.mas5, cs2.rma, cs2.mas5,
     cond=list(c("RMA","MAS5","RMA","MAS5"), c("A","A","B","B")))
```

Index

∗ **datagen**
    CorrSample, 2
∗ **datasets**
    oligodata, 5
∗ **hplot**
    plot.corr.sample, 5
∗ **utilities**
    CutCI, 3

CIrho *(*CutCI*)*, 3
co.intervals, *4*
CorrSample, 2, *5*, *7*
CutCI, 3

datA.amp *(*oligodata*)*, 5
datA.mas5 *(*oligodata*)*, 5
datA.rma *(*oligodata*)*, 5
datB.amp *(*oligodata*)*, 5
datB.mas5 *(*oligodata*)*, 5
datB.rma *(*oligodata*)*, 5

oligodata, 5

panel.corr.sample *(*plot.corr.sample*)*, 5
plot.corr.sample, *3*, 5, *5*

RandPairs *(*CorrSample*)*, 2

xyplot, *7*