Package 'ffpe'

November 12, 2026

Type Package

Title Quality assessment and control for FFPE microarray expression data

Version 1.56.0

Author Levi Waldron

Maintainer Levi Waldron <lwaldron.research@gmail.com>

Description Identify low-quality data using metrics developed for expression data derived from Formalin-Fixed, Paraffin-Embedded (FFPE) data. Also a function for making Concordance at the Top plots (CAT-plots).

Collate sampleQC.R sortedIqrPlot.R CATplot.R

Depends R (>= 2.10.0), TTR, methods

Imports Biobase, BiocGenerics, affy, lumi, methylumi, sfsmisc

Suggests genefilter, ffpeExampleData

License GPL (>2)

LazyLoad yes

biocViews Microarray, GeneExpression, QualityControl

git_url https://git.bioconductor.org/packages/ffpe

git_branch RELEASE_3_23

git_last_commit e4f9fc4

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Contents

ffpe-package 2
CATplot 3
sampleQC 4
sortedIqrPlot 7

Index **9**

`ffpe-package` *Quality assessment and control for FFPE microarray expression data*

Description

Identify low-quality data using metrics developed for expression data derived from Formalin-Fixed, Paraffin-Embedded (FFPE) data. Also a function for making Concordance at the Top plots (CAT-plots).

Details

Package:	ffpe
Type:	Package
Version:	1.0.0
Date:	2011-11-17
License:	GPL (>=2)
LazyLoad:	yes
biocViews:	Microarray, GeneExpression, QualityControl, Bioinformatics

Quality control of FFPE expression data for Illumina and Affymetrix microarrays. The function sampleQC identifies low-quality expression data, using IQR or any other surrogate quality measure for expression data. sortedIqrPlot provides a simplified, sorted boxplot of raw expression intensities as a quality summary for the experiment, suitable for large sample sizes and multiple batches.

Author(s)

Levi Waldron

Maintainer: Levi Waldron <lwaldron@hsph.harvard.edu>

References

under review

Examples

```
library(ffpeExampleData)
data(lumibatch.GSE17565)

QC <- sampleQC(lumibatch.GSE17565,xaxis="index",cor.to="pseudochip",QCmeasure="IQR")

##sort samples
QCvsRNA <- data.frame(inputRNA.ng=lumibatch.GSE17565$inputRNA.ng,rejectQC=QC$rejectQC)
QCvsRNA <- QCvsRNA[order(QCvsRNA$rejectQC,-QCvsRNA$inputRNA.ng),]

##QC rejects samples with lowest input RNA concentration\n
par(mgp=c(4,2,0))
```

```
dotchart(log10(QCvsRNA$inputRNA.ng),
         QCvsRNA$rejectQC,
         xlab="log10(RNA conc. in ng)",
         ylab="rejected?",
         col=ifelse(QCvsRNA$rejectQC,"red","black"))
```

CATplot *Make a Concordance at the Top plot.*

Description

For the i top-ranked members of each list, concordance is defined as length(intersect(vec1[1:i],vec2[1:i]))/i. This concordance is plotted as a function of i.

Usage

```
CATplot(vec1, vec2, maxrank = min(length(vec1),length(vec2)), make.plot = TRUE, ...)
```

Arguments

vec1, vec2	Two numeric vectors, for computing concordance. If these are numeric vectors with names, the numeric values will be used for sorting and the names will be used for calculating concordance. Otherwise, they are assumed to be already-ranked vectors, and the values themselves will be used for calculating concordance.
maxrank	Optionally specify the maximum size of top-ranked items that you want to plot.
make.plot	If TRUE, the plot will be made. Set to FALSE if you just want the concordance calculations.
...	Optional arguments passed onto plot()

Value

Returns a dataframe with two columns:

i	length of top lists
concordance	fraction in common that the two provided lists have in the top i items

Author(s)

Levi Waldron <lwaldron@hsph.harvard.edu>

References

The CAT-plot was suggested by Irizarry et al.:

Irizarry, R.A. et al. Multiple-laboratory comparison of microarray platforms. Nat Meth 2, 345-350 (2005).

The CAT-boxplot for multiple splits of a single dataset was suggested by Waldron et al. (under review).

Examples

```
library(ffpeExampleData)
data(lumibatch.GSE17565)

##preprocessing, individually for rep1 and rep2
lumibatch.rep1 <- lumibatch.GSE17565[,lumibatch.GSE17565$replicate==1]
lumbiatch.rep1 <- lumiT(lumibatch.rep1,"log2")
lumbiatch.rep1 <- lumiN(lumibatch.rep1,"quantile")
probe.var <- apply(exprs(lumibatch.rep1),1,var)
lumibatch.rep1 <- lumibatch.rep1[probe.var > median(probe.var),]

lumibatch.rep2 <- lumibatch.GSE17565[,lumibatch.GSE17565$replicate==2]
lumibatch.rep2 <- lumiT(lumibatch.rep2,"log2")
lumibatch.rep2 <- lumiN(lumibatch.rep2,"quantile")
lumibatch.rep2 <- lumibatch.rep2[featureNames(lumibatch.rep1),]

##row t-tests for differential expression
library(genefilter)
ttests.rep1 <- rowttests(exprs(lumibatch.rep1),fac=factor(lumibatch.rep1$cell.type))
ttests.rep2 <- rowttests(exprs(lumibatch.rep2),fac=factor(lumibatch.rep2$cell.type))

pvals.rep1 <- ttests.rep1$p.value;names(pvals.rep1) <- rownames(ttests.rep1)
pvals.rep2 <- ttests.rep2$p.value;names(pvals.rep2) <- rownames(ttests.rep2)

## Very high concordance between top differentially expressed gene lists
## identified by different replicates
x <- CATplot(pvals.rep1,pvals.rep2,maxrank=1000,xlab="Size of top-ranked gene lists",ylab="Concordance")
legend("topleft",lty=1:2,legend=c("Actual concordance","Concordance expected by chance"), bty="n")
```

---

sampleQC *Sample quality control for FFPE expression data*

Description

Expression data from FFPE tissues may contain a much larger range of quality than data from fresh-frozen tissues. This function sorts samples by some specified measure of array quality, Interquartile Range (IQR) by default, and plots the correlation of each sample's expression values against a "typical" sample for the study, as a function of the quality measure. A "typical" sample can either be the median pseudochip (default), or a specified number of samples with quality measure most

similar to that sample. An attempt is made to automatically select a threshold for rejection of low-quality samples, at the point of largest negative inflection of a Loess smoothing curve. for this plot.

Usage

```
sampleQC(data.obj,logtransform = TRUE, goby = 3, xaxis = "notindex", QCmeasure = "IQR", cor.to = "pseudo
       ## S4 method for signature 'LumiBatch'
sampleQC(data.obj,logtransform = TRUE, goby = 3, xaxis = "notindex", QCmeasure = "IQR", cor.to = "pseudo
       ## S4 method for signature 'AffyBatch'
sampleQC(data.obj,logtransform = TRUE, goby = 3, xaxis = "notindex", QCmeasure = "IQR", cor.to = "pseudo
       ## S4 method for signature 'matrix'
sampleQC(data.obj,logtransform = TRUE, goby = 3, xaxis = "notindex", QCmeasure = "IQR", cor.to = "pseudo
```

Arguments

`data.obj`	A data object of class LumiBatch, AffyBatch, or matrix. If matrix, the columns should contain samples and the rows probes. If using QCmeasure="IQR", it is critical that the data not be normalized. QCmeasure="ndetectedprobes" currently works only for LumiBatch objects.
`logtransform`	If TRUE, data will be log2-transformed before calculating IQR and correlation.
`goby`	This number of samples above and below each sample will be used to for calculating correlation. Used only if cor.to="similar".
`xaxis`	If "index", the QC measure will be converted to ranks. This can be useful for very discontinuous values of the QC measure, which interfere with generation of a smoothing line. If "notindex", the QC measure is used as-is.
`QCmeasure`	Automated options include "IQR" and "ndetectedprobes". These are the Interquartile Range and number of probes called present, respectively. QCmeasure can also be a numeric vector of length equal to the number of samples, to manually specify some other quality metric.
`cor.to`	"similar" to calculate correlation of each chip to neighbors within a sliding window of size 2*goby+1, or "pseudochip" to calculate correlation to a study-wide pseudochip. The former can be more sensitive, or more appropriate with a large number of failed chips, but does not work well with small sample size (<20). The default is "pseudochip".
`pseudochip.samples`	An integer vector, specifying the column numbers of samples to use in calculation of the median pseudochip. Default is to use all samples.
`detectionTh`	Nnominal detection p-value to consider a probe as detected or not (0.01 by default). Used only if QCmeasure="ndetectedprobes" and class(data.obj)=="LumiBatch"
`manualcutoff`	Optional manual specification of a cutoff for good and bad samples. If xaxis="index", manualcutoff specifies the number of samples that will be rejected. If xaxis="notindex", it is the value of the QC measure plotted on the x-axis, below which samples will be rejected.

mincor	Optional specification of a minimum correlation to the sliding window samples or median pseudochip, below which all samples will be rejected for QC. This is drawn as a horizontal line on the output plot.
maxcor	Optional specification of an upper limit of correlation to sliding window samples or median pseudochip, above which samples will not be considered in determining the QC cutoff. This can be useful if some structure in the high-quality end of the plot has the maximum downward inflection, causing most samples to be incorrectly rejected. In such case, specifying maxcor can force the otherwise automatically-determined threshold into a more reasonable region. Only values above at least 0.25, and probably below 0.8, make sense.
below.smoothed.threshold	Samples falling more than below.smoothed.threshold times the IQR of the residuals will be rejected for low QC. Large negative residuals from the Loess best-fit line may indicate outlier samples even if that sample has a high IQR or other quality measure.
lowess.f	Degree of smoothing of the Loess best-fit line (see ?loess)
labelnote	An optional label for the plot, used only if QCmeasure is a numeric vector.
pch	Plotting character to be used for points (see ?par).
lw	Line width for Loess curve.
linecol	Line color for Loess curve.
make.legend	If TRUE, an automatic legend will be added to the plot.
main.title	If specified, this over-rides the automatically-generated title.
...	Other arguments passed on to plot().

Details

These methods aid in the identification of low-quality samples from FFPE expression data, when technical replication is not available.

Value

If only one method is specified (one value each for xaxis, QCmeasure, and cor.to, the output is a dataframe with the following columns:

i	index or QC measure of each sample
spearman	spearman correlation to sliding window samples or to median pseudochip
movingaverage	moving average smoothing of spearman correlation
interpolate.i	evenly spaced QC measure (index or actual QC measure) used for plotting Loess curve
smoothed	values of the Loess curve
ddy	second derivative of the Loess curve
rejectQC	was this sample rejected in the QC process? logical TRUE or FALSE

If more than one method is specified, the output is a list, where each element contains a dataframe of the above description.

Author(s)

Levi Waldron <lwaldron@hsph.harvard.edu>

References

Under review.

Examples

```
library(ffpeExampleData)
data(lumibatch.GSE17565)

QC <- sampleQC(lumibatch.GSE17565,xaxis="index",cor.to="pseudochip",QCmeasure="IQR")

##sort samples
QCvsRNA <- data.frame(inputRNA.ng=lumibatch.GSE17565$inputRNA.ng,rejectQC=QC$rejectQC)
QCvsRNA <- QCvsRNA[order(QCvsRNA$rejectQC,-QCvsRNA$inputRNA.ng),]

##QC rejects samples with lowest input RNA concentration\n
par(mgp=c(4,2,0))
dotchart(log10(QCvsRNA$inputRNA.ng),
         QCvsRNA$rejectQC,
         xlab="log10(RNA conc. in ng)",
         ylab="rejected?",
         col=ifelse(QCvsRNA$rejectQC,"red","black"))
```

---

sortedIqrPlot *Minimal box plot of samples, which get ordered optionally by batch, then by IQR.*

Description

This plots only the 25th to 75th percentile of expression intensities (Interquartile Range), sorted from smallest to largest IQR. This modification is more readable than a regular boxplot for large sample sizes. An optional batch variable may be specified, so that the sorting is done within each batch.

Usage

```
sortedIqrPlot(data.obj, batchvar = rep(1, ncol(data.obj)), dolog2=FALSE, ...)
      ## S4 method for signature 'LumiBatch'
sortedIqrPlot(data.obj, batchvar = rep(1, ncol(data.obj)), dolog2=FALSE, ...)
      ## S4 method for signature 'AffyBatch'
sortedIqrPlot(data.obj, batchvar = rep(1, ncol(data.obj)), dolog2=FALSE, ...)
      ## S4 method for signature 'matrix'
sortedIqrPlot(data.obj, batchvar = rep(1, ncol(data.obj)), dolog2=FALSE, ...)
```

Arguments

data.obj	An object of class LumiBatch, AffyBatch, AffyBatch, or matrix. This should contain raw, unnormalized, expression intensities.
batchvar	Optional integer batch variable. If specified, samples will be sorted within batches only. Default is to assume a single batch for all data.
dolog2	If TRUE, data will be log2-transformed before plotting. Default is FALSE.
...	Optional arguments passed on to the plot() function.

Details

Function will be generally called for the side-effect of producing a plot of sorted IQRs, but if the output is redirected it also produces the IQRs.

Value

If the output is redirected, e.g.:

output <- sortedIqrPlot()

the function will return the IQR of each sample.

Author(s)

Levi Waldron <lwaldron@hsph.harvard.edu>

References

Under review.

See Also

sampleQC

Examples

```
library(ffpeExampleData)
data(lumibatch.GSE17565)
sortedIqrPlot(lumibatch.GSE17565,main="GSE17565")
```

Index

∗ **hplot**
CATplot, 3
sampleQC, 4
sortedIqrPlot, 7
∗ **package**
ffpe-package, 2

CATplot, 3

ffpe *(*ffpe-package*)*, 2
ffpe-package, 2

sampleQC, 4
sampleQC,AffyBatch-method *(*sampleQC*)*, 4
sampleQC,LumiBatch-method *(*sampleQC*)*, 4
sampleQC,matrix-method *(*sampleQC*)*, 4
sampleQC-method *(*sampleQC*)*, 4
sortedIqrPlot, 7
sortedIqrPlot,AffyBatch-method
*(*sortedIqrPlot*)*, 7
sortedIqrPlot,LumiBatch-method
*(*sortedIqrPlot*)*, 7
sortedIqrPlot,matrix-method
*(*sortedIqrPlot*)*, 7