Package ‘YuGene’

May 17, 2018

Type Package

Title A Simple Approach to Scale Gene Expression Data Derived from Different Platforms for Integrated Analyses

Version 1.1.6

Date 2018-05-18

Author Kim-Anh Le Cao, Florian Rohart, Leo McHugh, Othmar Korn, Christine A. Wells

Maintainer Florian Rohart <f.rohart@uq.edu.au>

Description Simple method for comparison of gene expression generated across different experiments, and on different platforms; that does not require global renormalization, and is not restricted to comparison of identical probes. YuGene works on a range of microarray dataset distributions, such as between manufacturers. The resulting output allows direct comparisons of gene expression between experiments and experimental platforms.

License GPL (>= 2)

Depends mixOmics, R (>= 2.10)

Repository CRAN

Date/Publication 2018-05-17 03:34:00 UTC

NeedsCompilation no

R topics documented:

YuGene-package ................................................. 2
array .......................................................... 3
ascorbate ...................................................... 4
pca.default .................................................. 5
pca.YuGene ................................................... 6
YuGene ......................................................... 8

Index 10
YuGene-package

A Simple Approach to Scale Gene Expression Data Derived from Different Platforms for Integrated Analyses

Description

Simple method for comparison of gene expression generated across different experiments, and on different platforms; that does not require global renormalization, and is not restricted to comparison of identical probes. YuGene works on a range of microarray dataset distributions, such as between manufacturers. The resulting output allows direct comparisons of gene expression between experiments and experimental platforms.

Details

The DESCRIPTION file:

Package: YuGene
Type: Package
Title: A Simple Approach to Scale Gene Expression Data Derived from Different Platforms for Integrated Analyses
Version: 1.1.6
Date: 2018-05-18
Author: Kim-Anh Le Cao, Florian Rohart, Leo McHugh, Othmar Korn, Christine A. Wells
Maintainer: Florian Rohart <f.rohart@uq.edu.au>
Description: Simple method for comparison of gene expression generated across different experiments, and on different platforms; that does not require global renormalization, and is not restricted to comparison of identical probes. YuGene works on a range of microarray dataset distributions, such as between manufacturers. The resulting output allows direct comparisons of gene expression between experiments and experimental platforms.
License: GPL (>= 2)
Depends: mixOmics,R (>= 2.10)
Repository: CRAN
Date/Publication: 2012-11-29 05:47:58

Index of help topics:

YuGene-package A Simple Approach to Scale Gene Expression Data Derived from Different Platforms for Integrated Analyses
array Combination of multiple array experiments
ascorbate Ascorbate Experiment
pca.YuGene Principal component analysis for the 'YuGene' class.
pca.default Principal Components Analysis from the mixOmics package

This package provides a single function (YuGene). It takes a log transformed dataset (ie multiple
microarray samples in an experiment) and converts the values to a cumulative proportion. Values close to zero have the lowest expression, and values close to 1 have the highest expression. When many datasets have been YuGene transformed, relative expression levels (YuGene values) can be directly compared across experiments without re-normalization without significant loss of sensitivity when compared to quantile normalized data.

Author(s)

Kim-Anh Le Cao, Florian Rohart, Leo McHugh, Othmar Korn, Christine A. Wells
Maintainer: Florian Rohart <f.rohart@uq.edu.au>

References


---

array

Combination of multiple array experiments

Description

Combination of 5 experiments. The data has been YuGene transformed, mapped to Ensembl ID. 2000 genes have been randomly selected.

Usage

data(array)

Format

A list containing the following components:

data.all Matrix of 82 samples and 2000 gene expression.
experiment.all a factor containing the name of the experiments.
platform.all a factor containing the platform of each sample.
type.all a factor containing the type of each sample.

Source

The data were downloaded from www.stemformatics.org.
References


---

ascorbate  Ascorbate Experiment

Description

log2 transformed samples using Illumina HumanWG-6 chips, 3 of which were controls, and three of which were sampled after the addition of ascorbate to the medium. Details and data available by searching ‘ascorbate’ at www.stemformatics.org. This dataset is a random subset of 5000 genes for smaller package size and faster example times

Usage

data(ascorbate)

Format

A list containing the following components:

gene  data frame with 48803 rows and 6 columns. The expression levels of 48803 transcripts for the 6 subjects.

condition a vector of 6 elements indicating the condition of each subject (‘4ng.ml’ or ‘100ng.ml’)

Source

The data were downloaded from www.stemformatics.org datasetID 5006.
References


pca.default

Principal Components Analysis from the mixOmics package

Description

Performs a principal components analysis from the pca function of the mixOmics package.

Usage

```r
## Default S3 method:
pca(x, ncomp = 2, center = TRUE, scale = FALSE,
    max.iter = 500, tol = 1e-09, ...)  
```

Arguments

- `x`: a numeric matrix (or data frame) which provides the data for the principal components analysis. It can contain missing values.
- `ncomp`: integer, if data is complete ncomp decides the number of components and associated eigenvalues to display from the pcasvd algorithm and if the data has missing values, ncomp gives the number of components to keep to perform the reconstitution of the data using the NIPALS algorithm. If NULL, function sets ncomp = min(nrow(x), ncol(x))
- `center`: a logical value indicating whether the variables should be shifted to be zero centered. Alternately, a vector of length equal the number of columns of `x` can be supplied. The value is passed to `scale`.
- `scale`: a logical value indicating whether the variables should be scaled to have unit variance before the analysis takes place. The default is FALSE for consistency with `prcomp` function, but in general scaling is advisable. Alternatively, a vector of length equal the number of columns of `x` can be supplied. The value is passed to `scale`.
- `max.iter`: integer, the maximum number of iterations in the NIPALS algorithm.
- `tol`: a positive real, the tolerance used in the NIPALS algorithm.
- `...`: not used.

Details

see `pca`
pca.YuGene

Principal component analysis for the ‘YuGene’ class.

Description

Performs a principal components analysis thanks to the `pca` function of the `mixOmics` package. The data are centered by study before performing the analysis, if the argument `study` is given.

Usage

```r
## S3 method for class 'YuGene'
pca(X, study, ncomp = 2, center = TRUE, scale = FALSE,
    max.iter = 500, tol = 1e-09,...)
```

Arguments

- `x`: a numeric matrix (or data frame) which provides the data for the principal components analysis. It can contain missing values.
- `study`: Factor of the study effect.
- `ncomp`: integer, if data is complete `ncomp` decides the number of components and associated eigenvalues to display from the `pcasvd` algorithm and if the data has missing values, `ncomp` gives the number of components to keep to perform the reconstitution of the data using the NIPALS algorithm. If NULL, function sets `ncomp = min(nrow(X), ncol(X))`
- `center`: a logical value indicating whether the variables should be shifted to be zero centered. Alternately, a vector of length equal the number of columns of `X` can be supplied. The value is passed to `scale`.
- `scale`: a logical value indicating whether the variables should be scaled to have unit variance before the analysis takes place. The default is FALSE for consistency with `prcomp` function, but in general scaling is advisable. Alternatively, a vector of length equal the number of columns of `X` can be supplied. The value is passed to `scale`.
- `max.iter`: integer, the maximum number of iterations in the NIPALS algorithm.
- `tol`: a positive real, the tolerance used in the NIPALS algorithm.
- `...`: not used.

Details

If the argument `study` is given, the data are centered per study prior to performing the PCA with the `pca` function of the `mixOmics` package. Otherwise, the PCA is performed on the input data `X`. 
Value

Same outputs as the pca function from the mixOmics package.
pca returns a list with class "pca" and "prcomp" containing the following components:

- `ncomp`: the number of principal components used.
- `sdev`: the eigenvalues of the covariance/correlation matrix, though the calculation is actually done with the singular values of the data matrix or by using NIPALS.
- `rotation`: the matrix of variable loadings (i.e., a matrix whose columns contain the eigenvectors).
- `X`: if `retx` is true the value of the rotated data (the centred (and scaled if requested) data multiplied by the rotation matrix) is returned.
- `center`, `scale`: the centering and scaling used, or FALSE.

Author(s)

KA Le Cao, Translational Research Institute, The University of Queensland Diamantina Institute, Australia
Florian Rohart, Australian Institute for Bioengineering and Nanotechnology, The University of Queensland, St Lucia, Australia
Leo McHugh, Queensland Facility for Advanced Bioinformatics
Othmar Korn, Australian Institute for Bioengineering and Nanotechnology, The University of Queensland, St Lucia, Australia
Christine A. Wells, Australian Institute for Bioengineering and Nanotechnology, The University of Queensland, St Lucia, Australia

References


Examples

```r
# load data
data(array)

YuGene.data = t(YuGene(t(array$data.all)))  # transpose the data to get the samples in columns

# PCA on YuGene data, centered by study
res.pca.yugene.center = pca(YuGene.data, ncomp = 3, scale = TRUE, center = TRUE, study = array$experiment.all)
expl.var = round(res.pca.yugene.center$sdev/sum(res.pca.yugene.center$sdev),4)*100

# plot of the results, one color per cell-type, one shape per study
plot(res.pca.yugene.center$x[,1],res.pca.yugene.center$x[,2],
     pch = as.numeric(array$experiment.all),
     col = as.numeric(array$type.all)+1, lwd = 2,
     cex = 1.5, cex.lab = 1.5,xlab=paste("PC1:"),ylab=paste("PC2:"))
```
YuGene


Description

YuGene is a simple method for comparison of gene expression generated across different experiments, and on different platforms; that does not require global renormalization, and is not restricted to comparison of identical probes. YuGene works on a range of microarray dataset distributions, such as between manufacturers. The resulting output allows direct comparisons of gene expression between experiments and experimental platforms.

Usage

YuGene(data.prop, progressBar = TRUE)

Arguments

data.prop a matrix or data.frame of log intensity values, with samples in columns and expression levels in rows. Can be probe or transcript level. Can be raw or previously (i.e. quantile) normalized data.

progressBar set to FALSE to suppress progress bar

Value

returns an object of class ‘YuGene’: a matrix of the same dimensions with each sample transformed to the cumulative proportion (YuGene) metric.

Note

Support for missing values not yet implemented. Will implement if requested.
YuGene

Author(s)

KA Le Cao, Translational Research Institute, The University of Queensland Diamantina Institute, Australia
Florian Rohart, Australian Institute for Bioengineering and Nanotechnology, The University of Queensland, St Lucia, Australia
Leo McHugh, Queensland Facility for Advanced Bioinformatics
Othmar Korn, Australian Institute for Bioengineering and Nanotechnology, The University of Queensland, St Lucia, Australia
Christine A. Wells, Australian Institute for Bioengineering and Nanotechnology, The University of Queensland, St Lucia, Australia

References


See Also

pca

Examples

data(ascorbate) # gene expression data available in YuGene package
# apply the transform to the data
YuGene.transformed <- YuGene(ascorbate$gene)

# show distributions before and after YuGene
opar <- par() # make a copy of current settings
par(mfrow=c(1,2))
plot(density(ascorbate$gene[,1]),main='Expression values', xlab='log2 expr.');
plot(density(YuGene.transformed[,1]),main='YuGene values',xlab='YuGene value');
par(opar) # restore original settings

# unadjusted pvals from the quantile normalized data
quant.pvals <- apply(ascorbate$gene,1,function(row){return(t.test(row[1:3],row[4:6])$p.value)})
YuGene.pvals <- apply(YuGene.transformed,1,function(row){return(t.test(row[1:3],row[4:6])$p.value)})
plot(quant.pvals,YuGene.pvals,pch='.',main='comparison of pvals before and after YuGene Transform')
text(0.8,0.2,paste("Pearson cor: ",round(cor(quant.pvals,YuGene.pvals,method='pearson'),digits=3)))
Index

*Topic datasets
  array, 3
  ascorbate, 4
*Topic manip
  YuGene, 8
*Topic package
  YuGene, 8

array, 3
ascorbate, 4

pca, 5, 6, 9
pca (pca.YuGene), 6
pca.default, 5
pca.YuGene, 6

scale, 5, 6

YuGene, 8
YuGene-package, 2