Package: ExpressionNormalizationWorkflow (via r-universe)

October 8, 2026

Type Package

Title Gene Expression Normalization Workflow

Version 1.39.0

Published 2014-04-30

Description An extensive, customized expression normalization workflow incorporating Supervised Normalization of Microarryas(SNM), Surrogate Variable Analysis(SVA) and Principal Variance Component Analysis to identify batch effects and remove them from the expression data to enhance the ability to detect the underlying biological signals.

Imports Biobase (>= 2.24.0), limma (>= 3.20.9), lme4 (>= 1.1.7), matrixStats (>= 0.10.3), pvca (>= 1.4.0), snm (>= 1.12.0), sva (>= 3.10.0), vsn (>= 3.32.0)

VignetteBuilder knitr

biocViews ImmunoOncologyWorkflow, Workflow, GeneExpressionWorkflow

Suggests knitr, BiocStyle

License GPL (>=3)

Encoding UTF-8

LazyData true

BugReports https://github.com/

RoxygenNote 5.0.1.9000

Config/pak/sysreqs cmake make libpng-dev libxml2-dev libssl-dev zlib1g-dev

Repository https://bioc.r-universe.dev

Date/Publication 2026-04-28 12:40:28 UTC

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

RemoteRef HEAD

RemoteSha 1a71bf288645a27c6e47a188c76f28eb48892120

Contents

ExpressionNormalizationWorkflow-package 2
conTocat 3
expSetobj 3
pvcAnaly 4
snmAnaly 5
surVarAnaly 7

Index **8**

`ExpressionNormalizationWorkflow-package`
Gene Expression Normalization Workflow

Description

An extensive, customized expression normalization workflow incorporating Supervised Normalization of Microarryas(SNM), Surrogate Variable Analysis(SVA) and Principal Variance Component Analysis to identify batch effects and remove them from the expression data to enhance the ability to detect the underlying biological signals.

Details

Package:	ExpressionNormalizationWorkflow
Type:	Package
Version:	1.0
Date:	2014-04-30
License:	GPL (>=3)

Author(s)

Karthikeyan Murugesan<karthikeyanm60 at yahoo.com>

References

~~ Literature or other references for background information ~~

See Also

~~ Optional links to other man pages, e.g. ~~ ~~ <pkg> ~~

Examples

```
#~~ simple examples of the most important functions ~~
```

conTocat — *Convert continuous values to categorical/discrete values*

Description

This function calculates the quantiles for the list of input values and assigns a quantile group to every data point

Usage

```
conTocat(covariates, var_names)
```

Arguments

covariates	samples x covariates matrix containing the continous covariates which need to be discretized
var_names	a list of the continuous variables that need to be discretized

Note

the function returns a modified samples x covariates matrix with the categorized variables appended to the end as another column

Author(s)

Karthikeyan Murugesan<karthikeyanm60 at yahoo.com> #@examples conTocat(covariates, var_names)

expSetobj — *Create a compact ExpressionSet Datatype Object*

Description

An expression set object can store the expression values and the meta data associated like the covariate information, annotations and experiment protocol data

Usage

```
expSetobj(expression, covariates, cov_desc = NULL)
```

Arguments

expression	the gene expression value martix with features (probes) as rows and samples as columns
covariates	the phenotype (covariate) matrix with samples as rows and the covariates as columns
cov_desc	a list of descriptors for each of the covariates

Value

exp_datObj ExpressionSet Datatype Object

Note

The return values of the function can be accessed as variable$parameter

Author(s)

Karthikeyan Murugesan <karthikeyanm60 at yahoo.com>

References

Gentleman RC, Carey VJ, Bates DM and others (2004). "Bioconductor: Open software development for computational biology and bioinformatics." Genome Biology, 5, pp. R80. #@examples expSetobj(expression, covariates)

---

pvcAnaly *Principal variance Component Analysis*

Description

PVCA estimates the variance in the expression dataset due to each of the given covariates and attributes the remaining fraction to residual. It efficiently combines principal component analysis

Usage

```
pvcAnaly(exp_datObj, pct_threshold, batch_factors)
```

Arguments

exp_datObj	an expression set class object
pct_threshold	PVCA Threshold Value (a value between 0 and 1) is the percentile value of the minimum amount of the variabilities that the selected principal components need to explain
batch_factors	a list of covariates whose effect size on the expression data is to be determined

Value

dat A numerical vector that contains the percentile of sources of batch effect for each covariate term

label A character vector containing the name for each covariate term

Note

The return values of the function can be accessed as <variable>$<parameter>

Author(s)

Karthikeyan Murugesan <karthikeyanm60 at yahoo.com>

References

Bushel P (2013). pvca: Principal Variance Component Analysis (PVCA).R package version 1.6.0.

See Also

pvcaBatchAssess #@examples pvcAnaly(exp_datObj, pct_threshold, batch_factors)

---

snmAnaly *Supervized Normalization of Microarrays*

Description

This function implements the Supervised Normalization of Microarrays on an expression matrix. The SNM technique tries to enhance our ability to detect the underlying biological signals by removing the effects of the adjustment variables which are usaully the causes of unwanted biases in the expression data #' It requires a set of biological covariates of interest and at least one probe-specific or intensity-dependent adjustment variable. Biological, Adjustment and the Intensity dependent adjustment variables are user defined

Usage

```
snmAnaly(exprs, cov, bv, av, iv, adj = TRUE)
```

Arguments

exprs	the gene expression value martix with features (probes) as rows and samples as columns
cov	the phenotype (covariate) matrix with samples as rows and the covariates as columns
bv	biological variable
av	adjustment variables whose effect on the expression has to be removed
iv	intenstity variables
adj	TRUE or FALSE value, default value is TRUE, If set to FALSE, then only the intensity dependent effects have been removed from the normalized data, implying the effects from the adjustment variables are still present. If TRUE, then the adjustment variables effects and the intensity dependent effects are both removed from the expression data

Value

norm.dat The matrix of normalized data

pvalues A vector of p-values testing the association of the biological variables with each probe. These p-values are obtained from an ANOVA comparing models where the full model contains both the probe-specific biological and adjustment variables versus a reduced model that just contains the probe-specific adjustment variables. The data used for this comparison has the intensity-dependent variables removed. These returned p-values are calculated after the final iteration of the algorithm.

pi0 The estimated proportion of true null probes pi_0, calculated after the final iteration of the algorithm

iter.pi0s A vector of length equal to num.iter containing the estimated pi_0 values at each iteration of the snm algorithm. These values should converge and any non-convergence suggests a problem with the data, the assumed model, or both

nulls A vector indexing the probes utilized in estimating the intensity-dependent effects on the final iteration.

bio.var The processed version of the same input variable

adj.var The processed version of the same input variable

int.var The processed version of the same input variable

df0 Degrees of freedom of the adjustment variable

df1 Degrees of freedom of the full model matrix, which includes the biological variables and the adjustment variables.

Note

The return values of the function can be accessed as <variable>$<parameter>

Author(s)

Karthikeyan Murugesan <karthikeyanm60 at yahoo.com>

References

Mecham BH, Nelson PS, Storey JD (2010) Supervised normalization of microarrays.Bioinformatics, 26: 1308-1315, Brig Mecham and John D. Storey<jstorey at princeton.edu>

See Also

snm #@examples snmAnaly(expression_matrix, covariate_matrix)

`surVarAnaly` *Estimate surrogate covariates unmodeled or hidden in the data*

Description

Surrogate variables are covariates constructed directly from high-dimensional data (like gene expression/RNA sequencing/methylation/brain imaging data) that can be used in subsequent analyses to adjust for unknown, unmodeled, or latent sources of noise.

Usage

```
surVarAnaly(expSetobj, blgy_var, n.sv = NULL)
```

Arguments

`expSetobj`	Expression set object
`blgy_var`	biological variable of interest
`n.sv`	number of surrogate variables to be estimated (if no argument provided then the algorithm estimates n.sv for the user)

Value

sv A n by n.sv matrix where each column is a distinct surrogate variable (the main quantity of interest)

pprob.gam posterior probability that each gene is associated with one or more latent variables

pprob.b posterior probability that each gene is associated with the variables of interest b

n.sv The number of surrogate variables estimated, Enter if n.sv known or just dont enter the parameter, the tool will identify the number of surrogate variables on its own

expSetobject Modified expression set object with the surrogate variables added as additional columns to covariate matrix

Note

The return values of the function can be accessed as <variable>$<parameter>

Author(s)

Karthikeyan Murugesan <karthikeyanm60 at yahoo.com>

References

Leek JT, Johnson WE, Parker HS, Fertig EJ, Jaffe AE and Storey JD. sva: Surrogate Variable Analysis. R package version 3.12.0.

See Also

`sva` #@examples surVarAnaly(expSetobj, blgy_var, adj_var)

Index

∗ **Adjustment,**
snmAnaly, 5
∗ **Analysis,**
pvcAnaly, 4
surVarAnaly, 7
∗ **Batch**
snmAnaly, 5
∗ **Component**
pvcAnaly, 4
∗ **Effect**
snmAnaly, 5
∗ **ExpressionSet**
expSetobj, 3
∗ **Microarray,**
snmAnaly, 5
∗ **Normalization,**
snmAnaly, 5
∗ **PCA**
pvcAnaly, 4
∗ **Principal**
pvcAnaly, 4
∗ **SNM,**
snmAnaly, 5
∗ **SVA**
surVarAnaly, 7
∗ **Supervised,**
snmAnaly, 5
∗ **Surrogate**
surVarAnaly, 7
∗ **VCA,**
pvcAnaly, 4
∗ **Variable**
surVarAnaly, 7
∗ **Variance**
pvcAnaly, 4
∗ **categorize,**
conTocat, 3
∗ **discretize**
conTocat, 3
∗ **package**
ExpressionNormalizationWorkflow-package, 2
∗ **quantile,**
conTocat, 3
<pkg>, *2*

conTocat, 3

ExpressionNormalizationWorkflow *(*ExpressionNormalizationWorkflow-package*)*, 2
ExpressionNormalizationWorkflow-package, 2
expSetobj, 3

pvcaBatchAssess, *5*
pvcAnaly, 4

snm, *6*
snmAnaly, 5
surVarAnaly, 7
sva, *7*