Package ‘ttScreening’

October 12, 2018

Type Package

Title Genome-Wide DNA Methylation Sites Screening by Use of Training and Testing Samples

Version 1.6

Date 2018-09-18

Author Meredith Ray, Xin Tong, Hongmei Zhang

Maintainer Meredith Ray <maray@memphis.edu>

Description A screening process utilizing training and testing samples to filter out uninformative DNA methylation sites. Surrogate variables (SVs) of DNA methylation are included in the filtering process to explain unknown factor effects.

License Artistic-2.0

Depends matrixStats,sva,limma

Imports graphics,stats,corpcor,simsalapar,MASS

biocViews

Suggests mvtnorm

NeedsCompilation no

Repository CRAN

Date/Publication 2018-10-11 22:00:02 UTC

R topics documented:

ttScreening-package *Genome-Wide DNA Methylation Sites Screening by Use of Training and Testing Samples*

Description

A screening process to filter out non-informative DNA methylation sites by applying (ordinary or robust) linear regressions to training data, and the results are further examined using testing samples. Surrogate variables are included to account for unknown factors.

Details

Package:	ttScreening
Type:	Package
Version:	1.6
Date:	2018-09-18
License:	Artistic-2.0

This package utilizes training and testing samples to filter out uninformative DNA methylation sites. Surrogate variables (SVs) of DNA methylation are included in the filtering process to explain unknown factor effects.

Author(s)

Meredith Ray, Xin Tong, Hongmei Zhang

Maintainer: Meredith Ray <maray@memphis.edu>

References

Ray MA, Tong X, Lockett GA, Zhang H, and Karmaus WJJ. (2016) “An Efficient Approach to Screening Epigenome-Wide Data”, BioMed Research International.

Leek JT and Storey JD. (2007) “Capturing heterogeneity in gene expression studies by ‘Surrogate Variable Analysis’.” PLoS Genetics, 3: e161.

See Also

sva

irwsva.build2 *Adjusted irwsva.build which builds surrogate variables from gene expression data*

Description

This function is directly modified from the original irwsva.build() in the SVA package. It was noticed that under certain circumstances a subscript out of bounds error would occur while running the SVA function. Therefore, this modified code has a single line altered that conditionally uses the generic singular decomposition, svd(), instead of fast singular decomposition, fast.svd().

Usage

```
irwsva.build2(dat, mod, mod0 = NULL, n.sv, B = 5)
```

Arguments

`dat`	A m CpG sites by n subjects matrix of methylation data.
`mod`	A n by k model matrix corresponding to the primary model fit (see model.matrix)
`mod0`	A n by k0 model matrix corresponding to the null model to be compared to mod.
`n.sv`	The number of surrogate variables to construct.
`B`	The number of iterations of the algorithm to perform.

Details

See http://www.bioconductor.org/packages/release/bioc/manuals/sva/man/sva.pdf

Value

`sv`	A n by n.sv matrix where each column is a distinct surrogate variable.
`pprob.gam`	A vector with the posterior probability estimates that each row is affected by dependence.
`pprob.b`	A vector with the posterior probabiliity estimates that each row is affected by the variables in mod, but not in mod0.
`n.sv`	The number of suggorate variables estimated.

Note

sva Vignette http://www.biostat.jhsph.edu/~jleek/sva/

Author(s)

Original irwsva.build: Jeffrey T. Leek <jleek@jhsph.edu>, John Storey jstorey@princeton.edu

References

Original sva: Leek JT and Storey JD. (2008) A general framework for multiple testing dependence.Proceedings of the National Academy of Sciences, 105: 18718-18723.

Leek JT and Storey JD. (2007) Capturing heterogeneity in gene expression studies by surrogate variable analysis. PLoS Genetics, 3: e161.

`num.sv2` *Adjusted num.sv which estimates the number of important surrogate variables from a gene expression data set.*

Description

This function is directly modified from the orginal num.sv() in the **sva** package. This function has the tolerance level in the fast.svd() function set back to its orginal default instead of 0.

Usage

```
num.sv2(dat, mod, method = c("be", "leek"), vfilter = NULL,
B = 20, sv.sig = 0.1, seed = NULL)
```

Arguments

`dat` A m genes by n arrays matrix of expression data.

`mod` A n by k model matrix corresponding to the primary model fit (see model.matrix).

`method` The method to use for estimating surrogate variables, for now there is only one option (based ib Buja and Eyuboglu 1992).

`vfilter` The number of most variable genes to use when building SVs, must be between 100 and m.

`B` The number of null iterations to perform. Only used when method="be".

`sv.sig` The significance cutoff for eigengenes. Only used when method="be".

`seed` A numeric seed for reproducible results. Optional, only used when method="be".

Details

See http://www.bioconductor.org/packages/release/bioc/manuals/sva/man/sva.pdf

Value

`n.sv` The number of significant surrogate variables

Note

sva Vignette http://www.biostat.jhsph.edu/~jleek/sva/

Author(s)

Original num.sv: Jeffrey T. Leek <jleek@jhsph.edu>, John Storey jstorey@princeton.edu

References

Original sva: Leek JT and Storey JD. (2008) A general framework for multiple testing dependence.Proceedings of the National Academy of Sciences, 105: 18718-18723.

Leek JT and Storey JD. (2007) Capturing heterogeneity in gene expression studies by surrogate variable analysis. PLoS Genetics, 3: e161.

`sva2` *The adjusted sva code using irwsva.build2*

Description

This function is the modified SVA function in which it uses the irwsva.build2 function rather than the irwsva.build function to build the surrogate variables. Thus, only a single line has been altered from the original SVA() function.

Usage

```
sva2(dat, mod, mod0 = NULL, n.sv = NULL, method = c("irw", "two-step"),
vfilter = NULL, B = 5, numSVmethod = "be")
```

Arguments

`dat`	An m by n (m cpg sites by n subjects) matrix of methylation data.
`mod`	A n by k model matrix corresponding to the primary model fit (see model.matrix).
`mod0`	A n by k0 model matrix corresponding to the null model to be compared to mod.
`n.sv`	Optional. The number of surrogate variables to estimate, can be estimated using the num.sv function.
`method`	Choose between the iteratively re-weighted or two-step surrogate variable estimation algorithms.
`vfilter`	The number of most variable CpG sites to use when building SVs, must be between 100 and m.
`B`	The number of iterations of the algorithm to perform.
`numSVmethod`	The method for determining the number of surrogate variables to use.

Details

See http://www.bioconductor.org/packages/release/bioc/manuals/sva/man/sva.pdf

Value

sv A n by n.sv matrix where each column is a distinct surrogate variable.

pprob.gam A vector with the posterior probability estimates that each row is affected by dependence.

pprob.b A vector with the posterior probabiliity estimates that each row is affected by the variables in mod, but not in mod0.

n.sv The number of suggorate variables estimated.

Note

sva Vignette http://www.biostat.jhsph.edu/~jleek/sva/

Author(s)

Original sva: Jeffrey T. Leek <jleek@jhsph.edu>, John Storey jstorey@princeton.edu

References

Original sva: Leek JT and Storey JD. (2008) A general framework for multiple testing dependence. Proceedings of the National Academy of Sciences, 105: 18718-18723.

Leek JT and Storey JD. (2007) Capturing heterogeneity in gene expression studies by surrogate variable analysis. PLoS Genetics, 3: e161.

ttScreening *A screening process built upon training and testing samples*

Description

A screening process to filter out non-informative DNA methylation sites by applying (ordinary or robust) linear regressions to training data, and the results are further examined using testing samples. Surrogate variables are included to account for unknown factors.

Usage

```
ttScreening(y = y, formula, imp.var, data, B.values=FALSE,iterations = 100,
sva.method = c("two-step", "irw"), cv.cutoff = 50, n.sv = NULL,
train.alpha = 0.05, test.alpha = 0.1, FDR.alpha = 0.05, Bon.alpha = 0.05,
percent = (2/3),linear = c("robust", "ls"), vfilter = NULL, B = 5,
 numSVmethod = "be", rowname = NULL, maxit=20)
```

Arguments

`y` Data matrix of DNA methylation measures (m by n, m CpG sites and n subjects). Each column represents DNA methylation measures of all CpG sites for one subject.

`formula` An object of class `formula` (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under "Details".

`imp.var` A vector indicating the location of the term(s) in the formula option on which the selection of CpG sites are made. Interactions are considered a single term. For example, suppose the right-hand side of the equation is: x + z + x:z. If the decision of selecting a CpG site is based on one single term, e.g., the significance of interaction effect, then imp.var is set as the location of that term, e.g., imp.var=3 (the third term). If the decision is desired to base on all the three terms, then imp.var=c(1,2,3).

`data` Data frame created from model.frame. Also is the data frame containing the variables defined in formula.

`B.values` Logical, TRUE if indicating the methylation is measured as beta values, FALSE if methylation is measured as M-values. The default is FALSE.

`iterations` Number of loops for the training/testing (TT) procedure. The default is 100.

`sva.method` Option of the two surrogate variable estimation algorithms, the iteratively reweighted, `"irw"`, or two-step, `"two-step"`. The default is `"two-step"`.

`cv.cutoff` The minimum frequency required for a DNA methylation site to be treated as an informative site. After "iterations" iterations, the frequency of each DNA methylation being selected out of `"iterations"` iterations is recorded. The higher the frequency, the more likely the site is informative. The default is 50.

`n.sv` Number of surrogate variables. If NULL, the number of surrogate variables will be determined based on the data. The default is NULL.

`train.alpha` Significance level for training samples. The default is 0.05.

`test.alpha` Significance level for testing samples. The default is 0.05.

`FDR.alpha` False discovery rate. The default is 0.05. This is to fit the need of selecting variables based on FDR.

`Bon.alpha` Overall significance level by use of the Bonferroni method for mulitple testing correction. The default is 0.05. This is to fit the need of selecting variables based on the Bonferroni multiple testing correction.

`percent` Proportion of the full sample to be used for training. The default is 2/3.

`linear` Choice of linear regression methods, `"robust"` (robust regression) or `"ls"` (ordinary least squares). The default is `"ls"`.

`vfilter` The number of most variable CpG sites to use when building SVs, must be between 100 and the number of genes; Must be NULL or numeric (> 0), The default is NULL.

`B` Number of iterations in generating surrogate variables. The default is 5.

`numSVmethod` The method for determining the number of surrogate variables to use. The default is `"be"`, the other method is `"leek"`.

`rowname` Optional, NULL or `"TRUE"`. The default is NULL. If rownames are not already present within the data, the order in which the DNA methylation sites are listed will become the rowname. Surrogate variable estimates are formed based on the algorithms in Leek and Storey (2007).

`maxit` Optional, controls the number of iterations for linear regression estimation methods. The default is 20.

Details

See *lm* or *glm* for details.

Value

`sub.remove` Denotes which subjects (based on order) were removed due to incomplete or missing data within the prediction variables defined in the formula arguement.

`train.cpg` Number of DNA methylation sites selected after the training step of each loop.

`test.cpg` Number of DNA methylation sites selected after the testing step of each loop.

`selection` Indicator matrix for the TT method after the testing step. The number of rows is the number of methylation sites, and the number of columns is the number of iterations. An entry of 1 indiates the selection of a site, and 0 otherwise.

`pvalue.matrix` Matrix of p-values of the selected DNA methylation sites after the testing step. The number of rows is the number of methylation sites and the number of columns is the number of iterations. For methylation sites not selected, NA is listed.

`TT.cpg` Final list of the DNA methylation sites by their original rownames selected from the TT method.

`FDR.cpg` Final list of the DNA methylation sites by their original rownames selected from the FDR method.

`Bon.cpg` Final list of the DNA methylation sites by their original rownames selected from the Bonferroni method.

`SV.output` Data frame containing the estimated surrogate variables.

`TT.output` Data frame containing the list of DNA methylation sites selected from the TT method and the respective coefficients and pvalues for the variables and SVs.

`FDR.output` Data frame containing the list of DNA methylation sites selected from the FDR method and the respective coefficients and pvalues for the variables and SVs.

`Bon.output` Data frame containing the list of DNA methylation sites selected from the Bonferroni method and the respective coefficients and pvalues for the variables and SVs.

References

Meredith Ray, Xin Tong, Hongmei Zhang, and Wilfred Karmaus. (2014) "DNA methylation sites screening with surrogate variables", unpublished manuscript.

Leek JT and Storey JD. (2007) "Capturing heterogeneity in gene expression studies by 'Surrogate Variable Analysis'." PLoS Genetics, 3: e161.

Examples

```
## Not run:
library(mvtnorm)
nsub=600
imp=100
num=2000

set.seed(1)
x1= rnorm(nsub,1,1)
size1<-rmultinom(1,nsub,c(0.15,0.25,0.25,0.35))
x2= matrix(sample(c(rep(0,size1[1,]),
rep(1,size1[2,]),
rep(2,size1[3,]),
rep(3,size1[4,])),replace=F),byrow=250,ncol=1)

sur1<-rnorm(nsub,0,5)
sur2<-rnorm(nsub,3,1)
sur3<-rnorm(nsub,0,1)
sur4<-rnorm(nsub,2,4)
sur5<-rnorm(nsub,0,3)

sigma1<-matrix(0,nrow=num,ncol=num)
diag(sigma1)<-1.5

beta0<-0.5
beta1<-0.3
beta2<-0.3
beta3<-0.3

sbeta1<-rnorm(1,0.5,0.01)
sbeta2<-rnorm(1,0.5,0.01)
sbeta3<-rnorm(1,0.5,0.01)
sbeta4<-rnorm(1,0.5,0.01)
sbeta5<-rnorm(1,0.5,0.01)

#beta matrix#
beta<-as.matrix(cbind(beta0,beta1,beta2,beta3,sbeta1,sbeta2,sbeta3,sbeta4,sbeta5))
beta.no2<-as.matrix(cbind(beta0,beta1,beta3,sbeta1,sbeta2,sbeta3,sbeta4,sbeta5))
beta.sur<-as.matrix(cbind(sbeta1,sbeta2,sbeta3,sbeta4,sbeta5))
#design matrix#
X<-as.matrix(cbind(rep(1,length(x1)),x1,x2,x1*x2,sur1,sur2,sur3,sur4,sur5))
X.no2<-as.matrix(cbind(rep(1,length(x1)),x1,x1*x2,sur1,sur2,sur3,sur4,sur5))
X.sur<-as.matrix(cbind(sur1,sur2,sur3,sur4,sur5))
#mu matrix#
imp1.mu<-matrix(rep(X%*%t(beta),9),nrow=nsub,ncol=(imp*0.9))
imp2.mu<-matrix(rep(X.no2%*%t(beta.no2),1),nrow=nsub,ncol=(imp*0.1))
noimp.mu<-matrix(rep(X.sur%*%t(beta.sur),num-imp),nrow=nsub,ncol=num-imp)
mu.matrix=cbind(imp1.mu, imp2.mu, noimp.mu)
error<-rmvnorm(nsub,mean=rep(0,num),sigma=sigma1,method = "chol")
y<-t(mu.matrix+error)
```

```
runs<-ttScreening(y=y,formula=~x1+x2+x1:x2,imp.var=3,data=data.frame(x1,x2),sva.method="two-step",
B.values=FALSE,iterations=100,cv.cutoff=50,n.sv=NULL,train.alpha=0.05,
test.alpha=0.05,FDR.alpha=0.05,Bon.alpha=0.05,percent=(2/3),linear= "ls",
vfilter = NULL, B = 5, numSVmethod = "be",rowname=NULL,maxit=20)

runs$TT.output
runs$FDR.output
runs$Bon.output

## End(Not run)
```

Index