Package ‘selectKSigs’

January 10, 2024

Type Package
Title Selecting the number of mutational signatures using a perplexity-based measure and cross-validation
Depends R(>= 3.6)
Imports HiLDA, magrittr, gtools, methods, Rcpp
Suggests knitr, rmarkdown, testthat, BiocStyle, ggplot2, dplyr, tidyr
Version 1.14.0
Date 2021-10-18
Description A package to suggest the number of mutational signatures in a collection of somatic mutations using calculating the cross-validated perplexity score.
URL https://github.com/USCbiostats/selectKSigs
BugReports https://github.com/USCbiostats/HiLDA/selectKSigs
License GPL-3
biocViews Software, SomaticMutation, Sequencing, StatisticalMethod, Clustering
RoxygenNote 7.1.2
LinkingTo Rcpp
VignetteBuilder knitr
Encoding UTF-8
git_url https://git.bioconductor.org/packages/selectKSigs
git_branch RELEASE_3_18
git_last_commit d940521
git_last_commit_date 2023-10-24
Repository Bioconductor 3.18
Date/Publication 2024-01-09
Author Zhi Yang [aut, cre], Yuichi Shiraishi [ctb]
Maintainer Zhi Yang <zyang895@gmail.com>
calcPMSLikelihood

A function for calculating the log-likelihood from the data and parameters

Usage

calcPMSLikelihood(p, y)

Arguments

p  this variable includes the parameters for mutation signatures and membership parameters

y  this variable includes the information on the mutation features, the number of mutation signatures specified and so on

Value

a value
Calculate_Likelihood_test

*Output the maximum potential scale reduction statistic of all parameters estimated*

**Description**

Output the maximum potential scale reduction statistic of all parameters estimated.

**Usage**

Calculate_Likelihood_test(train, test, paramG)

**Arguments**

- **train**: a MutationFeatureData S4 class output of training data.
- **test**: a MutationFeatureData S4 class output of test data.
- **paramG**: an estimatedParameters S4 class with estimated parameters.

**Value**

the likelihood of the test data

---

convertFromTurbo_F

*Restore the converted parameter F for turboEM*

**Description**

Restore the converted parameter F for turboEM.

**Usage**

convertFromTurbo_F(turboF, fdim, signatureNum, isBackground)

**Arguments**

- **turboF**: F (converted for turboEM)
- **fdim**: a vector specifying the number of possible values for each mutation signature
- **signatureNum**: the number of mutation signatures
- **isBackground**: the logical value showing whether a background mutation features is included or not

**Value**

a vector
**convertFromTurbo_Q**  
*Restore the converted parameter Q for turboEM*

**Description**

Restore the converted parameter Q for turboEM

**Usage**

```r
cvtColor_F(turboQ, signatureNum, sampleNum)
```

**Arguments**

- `turboQ`: Q (converted for turboEM)
- `signatureNum`: the number of mutation signatures
- `sampleNum`: the number of cancer genomes

**Value**

a vector

---

**convertToTurbo_F**  
*Convert the parameter F so that turboEM can treat*

**Description**

Convert the parameter F so that turboEM can treat

**Usage**

```r
cvtColor_F(vF, fdim, signatureNum, isBackground)
```

**Arguments**

- `vF`: F (converted to a vector)
- `fdim`: a vector specifying the number of possible values for each mutation signature
- `signatureNum`: the number of mutation signatures
- `isBackground`: the logical value showing whether a background mutation feature is included or not

**Value**

a vector
**convertToTurbo_Q**

*Convert the parameter Q so that turboEM can treat*

**Description**

Convert the parameter Q so that turboEM can treat

**Usage**

`convertToTurbo_Q(vQ, signatureNum, sampleNum)`

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vQ</td>
<td>Q (converted to a vector)</td>
</tr>
<tr>
<td>signatureNum</td>
<td>the number of mutation signatures</td>
</tr>
<tr>
<td>sampleNum</td>
<td>the number of cancer genomes</td>
</tr>
</tbody>
</table>

**Value**

a vector

---

**cv_PMSignature**

*Output the maximum potential scale reduction statistic of all parameters estimated*

**Description**

Output the maximum potential scale reduction statistic of all parameters estimated

**Usage**

`cv_PMSignature(inputG, Kfold = 3, nRep = 3, Klimit = 8)`

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>inputG</td>
<td>a MutationFeatureData S4 class.</td>
</tr>
<tr>
<td>Kfold</td>
<td>an integer number of the number of cross-validation folds.</td>
</tr>
<tr>
<td>nRep</td>
<td>an integer number of replications.</td>
</tr>
<tr>
<td>Klimit</td>
<td>an integer of the maximum value of number of signatures.</td>
</tr>
</tbody>
</table>

**Value**

a matrix of measures
Examples

```r
load(system.file("extdata/sample.rdata", package = "selectKSigs"))
results <- cv_PMSignature(G, Kfold = 3)
```

### getBG

Get the status of using the background signature

#### Description

Get the status of using the background signature

#### Usage

```r
getBG(object)
```

#### Arguments

- `object`: the EstimatedParameters class (the result of pmgetSignature)

#### Value

- the status of using the background signature

### getCounts

Get the count data in a matrix

#### Description

Get the count data in a matrix

#### Usage

```r
counts <- getCounts(object)
```

#### Arguments

- `object`: the MutationFeatureData class

#### Value

- the count data in a matrix
getExposures

Get a matrix of mutational exposures of signatures

Description
Get a matrix of mutational exposures of signatures

Usage
getExposures(object)

Arguments
object the EstimatedParameters class (the result of pmgetSignature)

Value
a matrix of mutational exposures of signatures

getFeatures

Get a vector of possible features

Description
Get a vector of possible features

Usage
getFeatures(object)

Arguments
object the EstimatedParameters class (the result of pmgetSignature)

Value
a vector of possible features
getFeatureVec

Get a matrix of feature vector list

Description
Get a matrix of feature vector list

Usage
getFeatureVec(object)

Arguments
object the MutationFeatureData class

Value
a matrix of feature vector list

getK

Get the number of signatures

Description
Get the number of signatures

Usage
getK(object)

Arguments
object the EstimatedParameters class (the result of pmgetSignature)

Value
the number of signatures in pmgetSignature in HiLDA
Description
Get the values of loglikelihood

Usage
getLL(object)

Arguments
object the EstimatedParameters class (the result of pmgetSignature)

Value
likelihood values estimated by pmgetSignature in HilDA

Description
Calculate the value of the log-likelihood for given parameters

Usage
getLogLikelihoodC(
vPatternList,
vSparseCount,
vF,
vQ,
fdim,
signatureNum,
sampleNum,
patternNum,
samplePatternNum,
isBackground,
vF0
)
getSamplelist

Arguments

- **vPatternList**: The list of possible mutation features (converted to a vector)
- **vSparseCount**: The table showing (mutation feature, sample, the number of mutation) (converted to a vector)
- **vF**: F (converted to a vector)
- **vQ**: Q (converted to a vector)
- **fdim**: a vector specifying the number of possible values for each mutation signature
- **signatureNum**: the number of mutation signatures
- **sampleNum**: the number of cancer genomes
- **patternNum**: the number of possible combinations of all the mutation features
- **samplePatternNum**: the number of possible combination of samples and mutation patterns
- **isBackground**: the logical value showing whether a background mutation features is included or not
- **vF0**: a background mutation features

Value

- a value

---

getSamplelist  
*Get the sample list*

Description

Get the sample list

Usage

getSamplelist(object)

Arguments

- **object**: the EstimatedParameters class (the result of pmgetSignature)

Value

the sample list of named elements.
**getSamplelistG**

*Get the sample list*

**Description**

Get the sample list

**Usage**

```r
getSamplelistG(object)
```

**Arguments**

- `object` the MutationFeatureData class

**Value**

the sample list of named elements.

---

**getSignatures**

*Get an array of signature feature distributions*

**Description**

Get an array of signature feature distributions

**Usage**

```r
getSignatures(object)
```

**Arguments**

- `object` the EstimatedParameters class (the result of pmgetSignature)

**Value**

an array of signature feature distributions
getTranscription  

*Get the status of specifying the transcription bias*

**Description**

Get the status of specifying the transcription bias

**Usage**

```r
getTranscription(object)
```

**Arguments**

- `object`: the MutationFeatureData class

**Value**

the status of specifying the transcription bias

---

select_kth_fold  

*Output the training data or test data*

**Description**

Output the training data or test data

**Usage**

```r
select_kth_fold(inputG, k, f_s, folds, include)
```

**Arguments**

- `inputG`: a MutationFeatureData S4 class output by the pmsignature.
- `k`: an integer number of the number of cross-validation folds.
- `f_s`: a primary key of combining the feature pattern and sample ID.
- `folds`: the assignment to each fold.
- `include`: a boolean indicator of whether to include kth fold or not.

**Value**

a MutationFeatureData S4 class of either include or exclude kth fold.
**splitG**

*Output the maximum potential scale reduction statistic of all parameters estimated*

**Description**

Output the maximum potential scale reduction statistic of all parameters estimated

**Usage**

```r
splitG(inputG, Kfold = 3)
```

**Arguments**

- `inputG`: a MutationFeatureData S4 class output by the pmsignature.
- `Kfold`: an integer number of the number of cross-validation folds.

**Value**

A matrix made of perplexity from the results of cross-validation.

**Examples**

```r
load(system.file("extdata/sample.rdata", package = "selectKSigs"))
G_split <- splitG(G, Kfold = 3)
```
Index

calcPMSLikelihood, 2
Calculate_Likelihood_test, 3
convertFromTurbo_F, 3
convertFromTurbo_Q, 4
convertToTurbo_F, 4
convertToTurbo_Q, 5
cv_PMSignature, 5

getBG, 6
getCounts, 6
getExposures, 7
getFeatures, 7
getFeatureVec, 8
getK, 8
getLL, 9
getLogLikelihoodC, 9
getSamplelist, 10
getSamplelistG, 11
getSignatures, 11
getTranscription, 12

select_kth_fold, 12
splitG, 13