

Package ‘mutSignatures’

January 24, 2017

Type Package

Title Decipher Mutational Signatures from Somatic Mutational Catalogs

Version 1.2

Date 2017-01-22

Author Damiano Fantini, Joshua J Meeks

Maintainer Damiano Fantini <damiano.fantini@gmail.com>

Description Cancer cells accumulate DNA mutations as result of DNA damage and DNA repair processes. This computational framework is aimed at deciphering DNA mutational signatures operating in cancer. The input is a numeric matrix of DNA mutation counts detected in a panel of cancer samples. The framework performs Non-negative Matrix Factorization to extract the most likely signatures explaining the observed set of DNA mutations. The framework relies on parallelization and is optimized for use on multi-core systems. This framework is an R-based implementation of the original MATLAB WTSI framework by Alexandrov LB et al (2013) <DOI:10.1016/j.celrep.2012.12.008>.

Depends R(>= 3.1.0), doParallel, ggplot2, foreach

Imports cluster, pracma, proxy

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2017-01-24 08:12:27

R topics documented:

mutSignatures-package	2
addWeak	3
bootstrapCancerGenomes	4
citation	5
decipherMutationalProcesses	5
deconvoluteMutCounts	6
do.nmf	7
evaluateStability	8
extractSignatures	9

filterOutIterations	10
getTestRunArgs	11
leadZeros	12
plotSignatureExposures	13
removeWeak	14
setMutClusterParams	15
setMutCountObject	16
silhouetteMLB	17
Index	18

mutSignatures-package *Computational Framework for Deciphering Cancer Mutational Signatures*

Description

Accumulation of DNA mutations is a hallmark of cancer. Cancer cells accumulate DNA mutations as result of DNA damage and DNA repair processes. This computational framework allows to characterize mutational signatures from cancer-derived somatic mutational catalogs

Details

Please, cite the following article when using this software: Fantini D et al, Mol Biol Cell. 2017 Jan 1;28(1):192-200. More info at: <http://www.biotechworld.it/bioinf/2017/01/22/mutsignatures/>
 Software developed in the Meeks Lab, Dept. of Urology, Northwestern University, Chicago, USA
 This is a R implementation of the original MATLAB framework described here: Cell Rep. 2013 Jan 31;3(1):246-59.

Author(s)

Damiano Fantini, Joshua J Meeks Maintainer: Damiano Fantini <damiano.fantini@gmail.com>

Examples

```

citation()
## Not run:
tmp <- getTestRunArgs()
result <- extractSignatures(mutCountMatrix = tmp$mutCount.obj$mutCounts, params = tmp$params)
result
plotSignatureExposures(results)

## End(Not run)

```

`addWeak`*Add Weak Mutation Types*

Description

Restore Mutation Types that were initially excluded because a low number of total counts

Usage

```
addWeak(mutationTypesToAddSet, processes_I, processesStd_I,  
Wall_I, genomeErrors_I, genomesReconstructed_I)
```

Arguments

<code>mutationTypesToAddSet</code>	Set of mutations to restore
<code>processes_I</code>	Set of Mutational Processes
<code>processesStd_I</code>	Set of standard deviations of all Mutational Processes
<code>Wall_I</code>	Set of all W matrices previously extracted
<code>genomeErrors_I</code>	Set of all residuals
<code>genomesReconstructed_I</code>	Fitted Values according to the most likely Model

Value

Output is the final result of the deconvolution process

Note

Affiliation: Meeks Lab, Department of Urology, Feinberg School of Medicine, Northwestern University, Chicago, IL 60611 USA Software: Original MATLAB framework was described here: Cell Rep. 2013 Jan 31;3(1):246-59

Author(s)

Damiano Fantini

References

More info at: <http://www.biotechworld.it/bioinf/2017/01/22/mutsignatures/>

Examples

```
# there are no examples for this function  
message("No examples")
```

bootstrapCancerGenomes

Bootstrap a Mutation Count Matrix

Description

Rearrange a Mutation count Matrix using the multivariate normal distribution. The function returns a bootstrapped Mutation Count matrix whose dimensions are identical to the input matrix

Usage

```
bootstrapCancerGenomes(genomes)
```

Arguments

genomes a numeric matrix of Mutation Counts. Rows correspond to Mutation Types, columns to different samples

Value

a numeric matrix of Mutation Counts. Rows correspond to Mutation Types, columns to different samples

Note

Affiliation: Meeks Lab, Department of Urology, Feinberg School of Medicine, Northwestern University, Chicago, IL 60611 USA Software: Original MATLAB framework was described here: Cell Rep. 2013 Jan 31;3(1):246-59

Author(s)

Damiano Fantini

References

More info at: <http://www.biotechworld.it/bioinf/2017/01/22/mutsignatures/>

Examples

```
tmp <- getTestRunArgs()
extractSignatures(mutCountMatrix = tmp$mutCount.obj$mutCounts[1:10,1:3], params = tmp$params)
```

citation

Print Citation Information

Description

Print to screen the information about the article to cite when using the software

Usage

`citation()`

Note

Affiliation: Meeks Lab, Department of Urology, Feinberg School of Medicine, Northwestern University, Chicago, IL 60611 USA Software: Original MATLAB framework was described here: Cell Rep. 2013 Jan 31;3(1):246-59

Author(s)

Damiano Fantini

References

More info at: <http://www.biotechworld.it/bioinf/2017/01/22/mutsignatures/>

Examples

`citation()`

`decipherMutationalProcesses`

Decipher Mutational Processes Contributing to a Collection of Genomic Mutations

Description

Cancer cells accumulate DNA mutations as result of DNA damage and DNA repair processes. This computational framework allows to decipher mutational processes from cancer-derived somatic mutational catalogs

Usage

`decipherMutationalProcesses(input, params)`

Arguments

input	object (list) including a Mutation Count matrix. This is the output of a setMutCountObject() call
params	object (list) including the parameters required for running the analysis. This is the output of a setMutClusterParams() call

Value

list including all results of the analysis. The extracted signatures (processes) are included in the "processes" element of the list. The relative contribution of each signature in each sample is summarized in the "exposures" element

Note

Affiliation: Meeks Lab, Department of Urology, Feinberg School of Medicine, Northwestern University, Chicago, IL 60611 USA Software: Original MATLAB framework was described here: Cell Rep. 2013 Jan 31;3(1):246-59

Author(s)

Damiano Fantini

References

More info at: <http://www.biotechworld.it/bioinf/2017/01/22/mutsignatures/>

Examples

```
## Not run:  
tmp <- getTestRunArgs(2)  
decipherMutationalProcesses(input = tmp$mutCount.obj, params = tmp$params)  
  
## End(Not run)
```

deconvoluteMutCounts *Deconvolute Mutation Counts*

Description

Characterize mutational signatures from cancer-derived somatic mutational catalogs

Usage

```
deconvoluteMutCounts(input.mutCounts, params)
```

Arguments

input.mutCounts numeric matrix of Mutation Type Counts
params object (list) including all parameters required for running the analysis

Value

list including all the results from the deconvolution analysis. This function is called within the decipherMutationalProcesses() function after parameters and input data have been validated.

Note

Affiliation: Meeks Lab, Department of Urology, Feinberg School of Medicine, Northwestern University, Chicago, IL 60611 USA Software: Original MATLAB framework was described here: Cell Rep. 2013 Jan 31;3(1):246-59

Author(s)

Damiano Fantini

References

More info at: <http://www.biotechworld.it/bioinf/2017/01/22/mutsignatures/>

Examples

```
## Not run:  
tmp <- getTestRunArgs(2)  
tmp  
deconvoluteMutCounts(input.mutCounts = tmp$mutCount.obj$mutCounts[1:5,1:6], params = tmp$params)  
  
## End(Not run)
```

do.nmf

Perform Non-negative Matrix Factorization

Description

Perform Non-negative Matrix Factorization

Usage

```
do.nmf(v, r, params, verbose = TRUE)
```

Arguments

v	numeric matrix of Mutation Type Counts
r	numeric, number of signatures to extract
params	object (list) including all parameters for running the analysis
verbose	logic, TRUE for printing information about the progress of the analysis

Value

list including two elements.

w	extracted signatures
h	contribution of each signature in all the samples of the input mut count matrix

Note

Affiliation: Meeks Lab, Department of Urology, Feinberg School of Medicine, Northwestern University, Chicago, IL 60611 USA Software: Original MATLAB framework was described here: Cell Rep. 2013 Jan 31;3(1):246-59

Author(s)

Damiano Fantini

References

More info at: <http://www.biotechworld.it/bioinf/2017/01/22/mutsignatures/>

Examples

```
tmp <- getTestRunArgs(3)
do.nmf(v = tmp$v, r = tmp$r, params = tmp$params)
```

evaluateStability *Evaluate Results Stability*

Description

Perform a final Stability check comparing the results from all iterations of the analysis

Usage

```
evaluateStability(wall, hall, params)
```

Arguments

wall	numeric matrix including the w results from all the iterations of the analysis
hall	numeric matrix including the h results from all the iterations of the analysis
params	object (list) including all the parameters required for running the analysis

Details

The function evaluates the results from all iterations by performing a silhouette check. A silhouette plot will also be plotted.

Value

list including all results from the stability checks. This includes the most likely signatures (centroids) and exposures. All information for plotting the silhouette plot will also be returned

Note

Affiliation: Meeks Lab, Department of Urology, Feinberg School of Medicine, Northwestern University, Chicago, IL 60611 USA Software: Original MATLAB framework was described here: Cell Rep. 2013 Jan 31;3(1):246-59

Author(s)

Damiano Fantini

References

More info at: <http://www.biotechworld.it/bioinf/2017/01/22/mutsignatures/>

Examples

```
tmp <- getTestRunArgs(5)
evaluateStability(wall = tmp$W, hall = tmp$H, params = tmp$params)
```

extractSignatures	<i>Extract Signatures from Genomic Mutational Catalogs</i>
-------------------	------------------------------------------------------------

Description

Extract mutational signatures after the input Data and the input parameters have been checked and validated

Usage

```
extractSignatures(mutCountMatrix, params, verbose = TRUE)
```

Arguments

mutCountMatrix	numeric matrix of mutation counts
params	object (list) including all parameters for performing the analysis
verbose	logic, if TRUE the progress of the analysis will be printed to screen

Value

list including the following elements

wall	extracted signatures
hall	extracted effects
mutCounts.reconstructed	fitted values
mutCounts.errors	residuals

Note

Affiliation: Meeks Lab, Department of Urology, Feinberg School of Medicine, Northwestern University, Chicago, IL 60611 USA Software: Original MATLAB framework was described here: Cell Rep. 2013 Jan 31;3(1):246-59

Author(s)

Damiano Fantini

References

More info at: <http://www.biotechworld.it/bioinf/2017/01/22/mutsignatures/>

Examples

```
tmp <- getTestRunArgs()
extractSignatures(mutCountMatrix = tmp$mutCount.obj$mutCounts[1:10,1:3], params = tmp$params)
```

filterOutIterations *Remove Iterations that Generated Outlier Results*

Description

Core function called from within a deconvoluteMutCounts() call

Usage

```
filterOutIterations(wall, hall, cnt.errors, cnt.reconstructed, params)
```

Arguments

wall	numeric matrix combining w results from all iterations
hall	numeric matrix combining h results from all iterations
cnt.errors	numeric matrix combining all residuals from all iterations
cnt.reconstructed	numeric matrix combining fitted values from all iterations
params	object (list) including all parameters for running the analysis

Value

list including all data required for running the subsequent stability check

Note

Affiliation: Meeks Lab, Department of Urology, Feinberg School of Medicine, Northwestern University, Chicago, IL 60611 USA Software: Original MATLAB framework was described here: Cell Rep. 2013 Jan 31;3(1):246-59

Author(s)

Damiano Fantini

References

More info at: <http://www.biotechworld.it/bioinf/2017/01/22/mutsignatures/>

Examples

```
# No examples are provided for this function  
message("no examples")
```

getTestRunArgs *Generate Arguments for Running Examples and Mock Runs*

Description

this function generates objects that can be used for running the examples in the package documentation files as well as some simple runs

Usage

```
getTestRunArgs(testN = 1)
```

Arguments

testN numeric, index corresponding to the type of test input to generate

Value

list of objects to be used as arguments for running the example codes of the package

Note

Affiliation: Meeks Lab, Department of Urology, Feinberg School of Medicine, Northwestern University, Chicago, IL 60611 USA Software: Original MATLAB framework was described here: Cell Rep. 2013 Jan 31;3(1):246-59

Author(s)

Damiano Fantini

References

More info at: <http://www.biotechworld.it/bioinf/2017/01/22/mutsignatures/>

Examples

```
getTestRunArgs(1)
```

leadZeros

Add Leading Zeros to a Number

Description

Converts a numeric to character and adds leading zeros, in order to allow consistent sorting. Currently, this function is not used by the core framework functions anymore.

Usage

```
leadZeros(n, m)
```

Arguments

n	number (integer) to be transformed
m	maximum number that will be used to define how many leading zeros to attach

Value

input number "n" converted to character and modified by attaching the required number of leading zeros

Note

Affiliation: Meeks Lab, Department of Urology, Feinberg School of Medicine, Northwestern University, Chicago, IL 60611 USA Software: Original MATLAB framework was described here: Cell Rep. 2013 Jan 31;3(1):246-59

Author(s)

Damiano Fantini

References

More info at: <http://www.biotechworld.it/bioinf/2017/01/22/mutsignatures/>

Examples

```
leadZeros(15, 99999)
```

`plotSignatureExposures`

Generate Plot Signature

Description

Plot Signature Contributions of All Samples in a Dataset

Usage

```
plotSignatureExposures(mutSign.analysis.result)
```

Arguments

```
mutSign.analysis.result  
object (list) including the results of a decipherMutationalProcesses() analysis
```

Details

Data are plotted using ggplot2

Value

ggplot2 object

Note

Affiliation: Meeks Lab, Department of Urology, Feinberg School of Medicine, Northwestern University, Chicago, IL 60611 USA Software: Original MATLAB framework was described here: Cell Rep. 2013 Jan 31;3(1):246-59

Author(s)

Damiano Fantini

References

More info at: <http://www.biotechworld.it/bioinf/2017/01/22/mutsignatures/>

Examples

```
tmp <- getTestRunArgs(6)  
gp <- plotSignatureExposures(tmp)  
gp + ggtitle("de novo mutSignatures") + scale_fill_manual(values = c("#377eb8", "#4daf4a"))
```

`removeWeak`*Remove Mutation Types Not Meeting the Threshold*

Description

Remove mutation types that account for a total number of mutations below a defined threshold.

Usage

```
removeWeak(input.mutCounts, params)
```

Arguments

`input.mutCounts`

numeric matrix of Mutation Counts

`params`

object (list) including all parameters required for running the analysis

Value

list including two elements

`removed.mutset` numeric vector including the ids of the removed rows. If `removed.mutset == -1`, no rows were removed

`output.mutCounts`

filtered Mutation Count matrix

Note

Affiliation: Meeks Lab, Department of Urology, Feinberg School of Medicine, Northwestern University, Chicago, IL 60611 USA Software: Original MATLAB framework was described here: Cell Rep. 2013 Jan 31;3(1):246-59

Author(s)

Damiano Fantini

References

More info at: <http://www.biotechworld.it/bioinf/2017/01/22/mutsignatures/>

Examples

```
tmp <- getTestRunArgs()
removeWeak(input.mutCounts = tmp$mutCount.obj$mutCounts, params = tmp$params)
```

setMutClusterParams *Set Parameters for Extracting Mutational Signatures*

Description

Create an object (list) including all parameters required for running the mutSignatures framework

Usage

```
setMutClusterParams(num.processes.toextract = 2, tot.iterations = 4,  
tot.cores = 5, remove.weak.muttypes = 0.01,  
remove.last.percent = 0.07, process.distance = "cosine",  
tot.Replicates = 100, eps = 2.2204e-16,  
stopconv = 10000, niter = 1e+06)
```

Arguments

num.processes.toextract	number of signatures to extract
tot.iterations	number of iterations per core
tot.cores	number of cores to use for the analysis
remove.weak.muttypes	threshold for filtering out under-represented mutation types
remove.last.percent	threshold for removing outlier iteration results
process.distance	method for calculating distances. Default method is 'cosine'
tot.Replicates	number of replicates while checking stability
eps	close-to-zero numeric value for replacing zeros and preventing negative values in the matrix
stopconv	number of non-changing iterations before stopping NMF
niter	total number of iterations for NMF

Value

Object (list) including all parameters for running the analysis

Note

Affiliation: Meeks Lab, Department of Urology, Feinberg School of Medicine, Northwestern University, Chicago, IL 60611 USA Software: Original MATLAB framework was described here: Cell Rep. 2013 Jan 31;3(1):246-59

Author(s)

Damiano Fantini

References

More info at: <http://www.biotechworld.it/bioinf/2017/01/22/mutsignatures/>

Examples

```
setMutClusterParams()
```

setMutCountObject	<i>Create a Mutation Count Object Suitable for Signatures Extraction</i>
-------------------	--------------------------------------------------------------------------

Description

Create a Mutation Count Object that can be used as data input for running the Signatures Extraction Framework

Usage

```
setMutCountObject(mutCountMatrix, mutationTypes = NULL,  
sampleNames = NULL, datasetName = NULL)
```

Arguments

mutCountMatrix	numeric matrix of mutation counts
mutationTypes	vector including the mutation types
sampleNames	vector including sample names
datasetName	name of the dataset

Value

list including the mutation Count Matrix and input data for de-novo extracting Mutational Signatures

Note

Affiliation: Meeks Lab, Department of Urology, Feinberg School of Medicine, Northwestern University, Chicago, IL 60611 USA Software: Original MATLAB framework was described here: Cell Rep. 2013 Jan 31;3(1):246-59

Author(s)

Damiano Fantini

References

More info at: <http://www.biotechworld.it/bioinf/2017/01/22/mutsignatures/>

Examples

```
tmp <- getTestRunArgs()[["mutCount.obj"]][["mutCounts"]]
tmp.obj <- setMutCountObject(tmp)
tmp.obj
```

silhouetteMLB

Silhouette Analysis

Description

Analyze the clustering quality and generate a Silhouette Plot

Usage

```
silhouetteMLB(data, fac, method = "cosine", plot = TRUE)
```

Arguments

data	numeric matrix
fac	clustering factor
method	method to be used as distance function
plot	logic, if TRUE plot a silhouette plot during the analysis

Value

numeric vector including the silhouette values of the data points in the input matrix

Note

Affiliation: Meeks Lab, Department of Urology, Feinberg School of Medicine, Northwestern University, Chicago, IL 60611 USA Software: Original MATLAB framework was described here: Cell Rep. 2013 Jan 31;3(1):246-59

Author(s)

Damiano Fantini

References

More info at: <http://www.biotechworld.it/bioinf/2017/01/22/mutsignatures/>

Examples

```
tmp <- getTestRunArgs(4)
silhouetteMLB(data = tmp$data, fac = tmp$fac)
```

Index

`addWeak`, [3](#)

`bootstrapCancerGenomes`, [4](#)

`citation`, [5](#)

`decipherMutationalProcesses`, [5](#)
`deconvoluteMutCounts`, [6](#)
`do.nmf`, [7](#)

`evaluateStability`, [8](#)
`extractSignatures`, [9](#)

`filterOutIterations`, [10](#)

`getTestRunArgs`, [11](#)

`leadZeros`, [12](#)

`mutSignatures (mutSignatures-package)`, [2](#)
`mutSignatures-package`, [2](#)

`plotSignatureExposures`, [13](#)

`removeWeak`, [14](#)

`setMutClusterParams`, [15](#)
`setMutCountObject`, [16](#)
`silhouetteMLB`, [17](#)