Package ‘SUITOR’

July 16, 2025

Title Selecting the number of mutational signatures through cross-validation

Version 1.11.0

Date 2022-05-10

Description An unsupervised cross-validation method to select the optimal number of mutational signatures. A data set of mutational counts is split into training and validation data.Signatures are estimated in the training data and then used to predict the mutations in the validation data.

Imports stats, utils, graphics, ggplot2, BiocParallel

Depends R (>= 4.2.0)

License GPL-2

biocViews Genetics, Software, SomaticMutation

Suggests devtools, MutationalPatterns, RUnit, BiocManager, BiocGenerics, BiocStyle, knitr, rmarkdown

NeedsCompilation yes

BugReports `https://github.com/wheelerb/SUITOR/issues`

VignetteBuilder knitr

git_url https://git.bioconductor.org/packages/SUITOR

git_branch devel

git_last_commit f49f293

git_last_commit_date 2025-04-15

Repository Bioconductor 3.22

Date/Publication 2025-07-15

Author DongHyuk Lee [aut],
Bin Zhu [aut],
Bill Wheeler [cre]

Maintainer Bill Wheeler <`wheelerb@imsweb.com`>

Contents

`SUITOR-package` *Number of mutational signatures*

Description

To select the number of mutational signatures through cross-validation.

Details

SUITOR (Selecting the nUmber of mutatIonal signaTures thrOugh cRoss-validation), an unsupervised cross-validation method that requires little assumptions and no numerical approximations to select the optimal number of signatures without overfitting the data. The full dataset of mutation counts is split into a training set and a validation set; for a given number of signatures, these signatures are estimated in the training set and then they are used to predict the mutations in the validation set. Multiple candidate numbers of signatures are considered; and the number of signatures which predicts most closely the mutations in the validation set is selected.

The two main functions in this package are `suitor` and `suitorExtractWH`.

Author(s)

Donghyuk Lee <dhyuklee@pusan.ac.kr> and Bin Zhu <bin.zhu@nih.gov>

References

Lee, D., Wang, D., Yang, X., Shi, J., Landi, M., Zhu, B. (2021) SUITOR: selecting the number of mutational signatures through cross-validation. bioRxiv, doi: https://doi.org/10.1101/2021.07.28.454269.

`getSummary` *Compute summary results*

Description

Compute summary results and the optimal rank from the matrix containing all results.

Usage

```
getSummary(obj, NC, NR=96)
```

Arguments

`obj`	Matrix containing all results in the return list from `suitor`.
`NC`	The number of columns in `data` when `suitor` was called.
`NR`	The number of rows in `data` when `suitor` was called. The default is 96.

Details

The input matrix `obj` must have column 1 as the rank, column 2 as the value of `k` in `1:k.fold`, column 4 as the training errors, and column 5 as the testing errors.

Value

A list containing the objects:

- `rank`: The optimal rank
- `all.results`: Matrix containing training and testing errors for all values of seeds, ranks, folds. NA values appear for runs in which the EM algorithm did not converge.
- summary: Data frame of summarized results for each possible rank created from `all.results`. The `MSErr` column is defined as sqrt({fold1 + ... +foldK}/{nrow(data)*ncol(data)})

Author(s)

Donghyuk Lee <dhyuklee@pusan.ac.kr> and Bin Zhu <bin.zhu@nih.gov>

See Also

`plotErrors`

Examples

```
data(SimData, package="SUITOR")
data(results, package="SUITOR")
ret <- getSummary(results$all.results, ncol(SimData))
ret$summary
ret$rank
```

`plotData` *Example data for plotting*

Description

A data frame with columns Rank, Type, and MSErr

See Also

`suitor`

Examples

```
data(plotData, package="SUITOR")

plotData
```

`plotErrors` *Plot train and test errors*

Description

Plot train and test errors

Usage

```
plotErrors(x)
```

Arguments

`x`	Data frame of `summary` results in the return list from `suitor` or from `getSummary`, or a data frame with columns `Rank`, `Type`, and `MSErr`.

Details

The optimal rank is the minimum at which the test error is attained, and appears as a red dot on the graph.

Value

NULL

Author(s)

Donghyuk Lee <dhyuklee@pusan.ac.kr> and Bin Zhu <bin.zhu@nih.gov>

Examples

```
data(plotData, package="SUITOR")
plotErrors(plotData)
```

`results` *suitor return object*

Description

An object returned from the suitor function for examples

See Also

`suitor`

Examples

```
data(results, package="SUITOR")

results
```

`SimData` *Data for examples*

Description

Example input data and results

Details

Contains an example input data object of size 96 by 300. It is generated by `rpois` with mean WH where W (96 by 8) is profile of 8 signatures (SBS 4, 6, 7a, 9, 17b, 22, 26, 39) obtained from https://cancer.sanger.ac.uk/cosmic/signatures/SBS and H (8 by 300) is rounded integer generated from a uniform distribution between 0 and 100 with some randomly selected cells being set to zero.

See Also

`suitor`

Examples

```
data(SimData, package="SUITOR")

# Display a subset of data objects
SimData[1:5, 1:5]
```

`suitor` *suitor*

Description

Selecting the number of mutational signatures through cross-validation

Usage

```
suitor(data, op=NULL)
```

Arguments

`data`	Data frame or matrix containing mutational signatures. This object must contain non-negative values
`op`	List of options (see details). The default is NULL.

Details

The algorithm finds the optimal rank by applying k-fold cross validation.

Options list op:

Name	Description	Default Value
em.eps	EM algorithm stopping tolerance	1e-5
get.summary	0 or 1 to create summary results	1
k.fold	Number of folds	10
max.iter	Maximum number of iterations in EM algorithm	2000
max.rank	Maximum rank	10
min.rank	Minimum rank	1
min.value	Minimum value of matrix before factorizing	1e-4
BPPARAM	See `BiocParallelParam`	NULL
n.starts	Number of starting points	30
plot	0 or 1 to produce an error plot	1
print	0 or 1 to print info	1
kfold.vec	Vector of values in 1:k.fold when running on a cluster	NULL

Parallel computing

The `BiocParallel` package is used for parallel computing. If `BPPARAM` = `NULL`, then `BPPARAM` will be set to `SerialParam`.

Utilizing a cluster

When running on a cluster, the option `get.summary` should be set to 0. For fastest running jobs, set the options `min.rank` = `max.rank`, `kfold.vec` to a single integer in `1:k.fold`, and `n.starts` to 1.

Value

A list containing the objects:

- `rank`: The optimal rank
- `all.results`: Matrix containing training and testing errors for all values of seeds, ranks, folds.
- summary: Data frame of summarized results for each possible rank created from `all.results`. The `MSErr` column is defined as sqrt({fold1 + ... +foldK}/{nrow(data)*ncol(data)})

Author(s)

Donghyuk Lee <dhyuklee@pusan.ac.kr> and Bin Zhu <bin.zhu@nih.gov>

See Also

`getSummary`, `plotErrors`

Examples

```
data(SimData, package="SUITOR")

# Using the default options will take several minutes to run
ret <- suitor(SimData)
```

`suitorExtractWH` *suitorExtractWH*

Description

Extract the matrix of activities (exposures) and matrix of signatures

Usage

```
suitorExtractWH(data, rank, op=NULL)
```

Arguments

`data`	Data frame or matrix containing mutational signatures. This object must contain non-negative values
`rank`	Integer > 0
`op`	List of options (see details). The default is NULL.

Details

Options list op:

Name	**Description**	**Default Value**
min.value	Minimum value of matrix before factorizing	1e-4
BPPARAM	See `BiocParallelParam`	NULL
n.starts	Number of starting points	30
print	0 or 1 to print info	1

Parallel computing

The `BiocParallel` package is used for parallel computing. If `BPPARAM = NULL`, then `BPPARAM` will be set to `SerialParam`.

Value

A list containing the objects:

- `H`: Matrix of activities (exposures)
- `W`: Matrix of signatures

Author(s)

Donghyuk Lee <dhyuklee@pusan.ac.kr> and Bin Zhu <bin.zhu@nih.gov>

See Also

`suitor`

Examples

```
data(SimData, package="SUITOR")

suitorExtractWH(SimData, 2)
```

Index