Package 'SigCheck'

April 10, 2015

Type Package

Title Check a gene signature's classification performance against random signatures, permuted data, and known signatures.

Version 1.0.2

Author Justin Norden <jn333@cam.ac.uk> and Rory Stark <rory.stark@cruk.cam.ac.uk>

Maintainer Rory Stark <rory.stark@cruk.cam.ac.uk>

Description While gene signatures are frequently used to classify data (e.g. predict prognosis of cancer patients), it it not always clear how optimal or meaningful they are (cf David Venet, Jacques E. Dumont, and Vincent Detours' paper ``Most Random Gene Expression Signatures Are Significantly Associated with Breast Cancer Outcome''). Based partly on suggestions in that paper, SigCheck accepts a data set (as an ExpressionSet) and a gene signature, and compares its classification performance (using the MLInterfaces package) against a) random gene signatures of the same length; b) known, (related and unrelated) gene signatures; and c) permuted data.

License Artistic-2.0

LazyLoad yes

Depends R (>= 3.1.0), MLInterfaces, Biobase, e1071, BiocParallel

Imports graphics, stats, utils

Suggests BiocStyle, breastCancerNKI

biocViews GeneExpression, Classification, GeneSetEnrichment

R topics documented:

`SigCheck-package` *Check a gene signature's classification performance against random signatures, permuted data, and known signatures.*

Description

While gene signatures are frequently used to classify data (e.g. predict prognosis of cancer patients), it is not always clear how optimal or meaningful they are (cf David Venet, Jacques E. Dumont, Vincent Detours' paper "Most Random Gene Expression Signatures Are Significantly Associated with Breast Cancer Outcome"). Based on suggestions in that paper, SigCheck accepts a data set (as an ExpressionSet) and a gene signature, and compares its classification performance (using the MLInterfaces package) against a) random gene signatures of the same length; b) permuted data; and c) known, unrelated gene signatures.

Details

Package:	SigCheck
Type:	Package
Version:	1.0
Date:	2014-06-26
License:	Artistic-2.0

SigCheck provides a high-level function, sigCheck, that runs all the core functions in turn. The four core functions enable 1) a genes signature's baseline classification performance to be established (`sigCheckClassifier`), 2) compares performance against signatures composed of random genes (`sigCheckRandom`), 3) compares performance against known, and mostly unrelated, gene signatures (`sigCheckKnown`), and 4) compares performance against randomly permuted data (`sigCheckPermuted`).

At a minimum, SigCheck requires a data set (as an `ExpressionSet`) and a signature (a subset of features in the ExpressionSet). It uses the `MLearn` funciton formt he `MLInterfaces` package to build a classifier (using `link{smvI}` by default) and measure its performance against validation samples in the ExpressionSet; if no validation samples are specified, it uses leave-one-out (LOO) cross-validation to build multiple classifiers, each predicting one sample.

Output of each check includes the distribution of random performance scores (percentage of validation samples correctly classified) and the ranking of the passed signature in this distribution. A simple p-value calculation based on this rank is also returned.

Author(s)

Originally written by Justin Norden with Rory Stark at the University of Cambridge, Cancer Resaerch UK Cambridge Institute.

Maintainer: Rory Stark <rory.stark@cruk.cam.ac.uk>

References

Venet, David, Jacques E. Dumont, and Vincent Detours. "Most random gene expression signatures are significantly associated with breast cancer outcome." PLoS Computational Biology 7.10 (2011): e1002240.

---

knownSignatures *Previously identified gene signatures for use in* sigCheckKnown

---

Description

Previously identified gene signature sets. These include three signatures sets from Venet et. al.

Usage

```
data(knownSignatures)
```

Format

The data object knownSignatures is a list of sets of gene signatures. Each set is a list of gene signatures. Each signature is a vector of gene names.

Gene signature sets include:

- "cancer": 48 gene signatures derived from cancer samples, from Venet et. al.
- "proliferation": 5 gene signatures comprising genes associated with cell proliferation, including a "super signature", from Venet et. al.
- "non.cancer": 3 gene signatures derived from non-cancer sources, from Venet et. al.

Details

These data are taken directly from the supplemental material for Venet et. al "Most random gene expression signatures are significantly associated with breast cancer outcome".

Note

Other signatures of interest can be downlaoded at http://www.broad.mit.edu/gsea/downloads.jsp#msigdb.

Source

http://www.ploscompbiol.org/article/fetchSingleRepresentation.action?uri=info:doi/10.1371/journal.pcbi.1002240.s001

References

Venet, David, Jacques E. Dumont, and Vincent Detours. "Most random gene expression signatures are significantly associated with breast cancer outcome." PLoS Computational Biology 7.10 (2011): e1002240.

Examples

```
data(knownSignatures)
names(knownSignatures)
names(knownSignatures$cancer)
knownSignatures$cancer$VANTVEER
```

`nkiResults` *Precomputed list of results for a call to* `sigCheck` *using the* `breastCancerNKI` *dataset.*

Description

This object represents the results lists returned by a call to `sigCheck`. It is used by the vignette accompanying the `SigCheck` package as an example result. It was derived by running the code in the example below.

Usage

```
data(nkiResults)
```

Examples

```
## Not run:
# This is how nkiResults is built
library(breastCancerNKI)
data(nki)
nki = nki[,!is.na(nki$e.dmfs)]
data(knownSignatures)
nkiResults = sigCheck(nki, classes="e.dmfs", annotation="HUGO.gene.symbol",
                  signature=knownSignatures$cancer$VANTVEER,
                  validationSamples=275:319,
                  knownSignatures="cancer",nIterations=1000)

## End(Not run)

# Example usage of nkiResults
data(nkiResults)
nkiResults$checkClassifier
sigCheckPlot(nkiResults$checkRandom)
```

sigCheck — *Check classification potential of a gene signature against randomly selected gene signatures, known gene signatures, and permuted expression sets.*

Description

High-level function for package SigCheck that runs all available checks against a classification signature.

Usage

```
sigCheck(expressionSet, classes, signature, annotation, validationSamples,
         classifierMethod = svmI, nIterations = 10, knownSignatures="cancer",
         plotResults=TRUE)
```

Arguments

expressionSet	An ExpressionSet object containing the data to be checked, ncluding an expression matrix, feature labels, and samples.
classes	Specifies which label is to be used to determine the classification categories (must be one of varLabels(expressionSet)). There should be only two unique values in expressionSet$classes.
signature	A vector of feature labels specifying which features comprise the signature to be checked. These feature labels should match values as specified in the annotation parameter (default is row names in the expressionSet). Alternatively, this can be a integer vector of feature indexes.
annotation	Character string specifying which featureData field should be used as the annotation. If missing, the row names of the expressionSet are used as the feature names.
validationSamples	Optional specification, as a vector of sample indices, of what samples in the expressionSet should be used for validation. If present, a classifier will be trained, using the specified signature and classification method, on the non-validation samples, and it's performance evaluated by attempting to classify the validations samples. If missing, a leave-one-out (LOO) validation method will be used, where a separate classifier will be trained to classify each sample using the reaming samples.
classifierMethod	The MLInterfaces learnerSchema object indicating the machine learning method to use for classification. Default is svmI for linear Support Vector Machine classification. See MLearn for available methods.
nIterations	For random gene and permutation tests, the number of iterations to run to compare classification outcomes.

knownSignatures
: Either a character string specifying which set of signatures to use from the included sets in knownSignatures, or a list of previously identified signatures to compare performance against. Each element in the list should be a vector of feature labels. Default is to use the "cancer" signatures from the included knownSignatures data set, taken from Venet et. al.

plotResults
: if TRUE, will call sigCheckPlot four times to plot the results of all checks (laid out in a 2x2 plot matrix).

Details

First, sigCheck calls sigCheckClassifier to establish the baseline performance of the signature being checked.

Next, it calls sigCheckRandom to check the performance of randomly selected signatures.

This is followed by a call to sigCheckKnown to check the performance of the signature against a database of signatures previously identified to discriminate in other, generally unrelated domains.

Finally, two calls are made to sigCheckPermuted to check the performance of randomly permuted data; the first call permutes the rows (toPermute="features"), while the second call permutes the categories (toPermute="categories").

Value

A list containing five elements:

- $checkClassifier is the result list returned by sigCheckClassifier.
- $checkRandom is the result list returned by sigCheckRandom.
- $checkKnown is the result list returned by sigCheckKnown.
- $checkPermutedFeatures is the result list returned by sigCheckPermuted with toPermute="features".
- $checkPermutedCategories is the result list returned by sigCheckPermuted with toPermute="categories".

Author(s)

Justin Norden with Rory Stark

References

Venet, David, Jacques E. Dumont, and Vincent Detours. "Most random gene expression signatures are significantly associated with breast cancer outcome." PLoS Computational Biology 7.10 (2011): e1002240.

See Also

sigCheckClassifier, sigCheckRandom, sigCheckPermuted, sigCheckKnown, MLearn

Examples

```
library(breastCancerNKI)
data(nki)
nki <- nki[,!is.na(nki$e.dmfs)]
data(knownSignatures)
results <- sigCheck(nki, classes="e.dmfs",
                    annotation="HUGO.gene.symbol",
                    signature=knownSignatures$cancer$VANTVEER,
                    validationSamples=275:319, nIterations=5)
```

`sigCheckClassifier` *Establish baseline classification performance for a signature*

Description

Compute classification performance of a signature by training one or more classifiers and testing their ability to predict validation samples.

Usage

```
sigCheckClassifier(expressionSet, classes, signature, annotation,
                   validationSamples, classifierMethod = svmI, ...)
```

Arguments

`expressionSet`	An `ExpressionSet` object containing the data to be checked, including an expression matrix, feature labels, and samples.
`classes`	Specifies which label is to be used to determine the classification categories (must be one of `varLabels(expressionSet)`). There should be only two unique values in `expressionSet$classes`.
`signature`	A vector of feature labels specifying which features comprise the signature to be checked. These feature labels should match values as specified in the `annotation` parameter (default is row names in the expressionSet). Alternatively, this can be a integer vector of feature indexes.
`annotation`	Character string specifying which `featureData` field should be used as the annotation. If missing, the row names of the expressionSet are used as the feature names.
`validationSamples`	Optional specification, as a vector of sample indices, of what samples in the `expressionSet` should used for validation. If present, a classifier will be trained, using the specified signature and classification method, on the non-validation samples, and it's performance evaluated by attempting to classify the validations samples. If missing, a leave-one-out (LOO) validation method will be used, where a separate classifier will be trained to classify each sample using the remaining samples.

`classifierMethod`
: The MLInterfaces learnerSchema object indicating the machine learning method to use for classification. Default is `svmI` for linear Support Vector Machine classification. See `MLearn` for available methods.

`...`
: additional parameters to be passed to `MLearn` in support of the classification method specified in `classifierMethod`.

Details

If `validationSamples` are specified, the `MLInterfaces` package is used to train a classifier on the remaining samples. By default, a Support Vector Machine classifier is used, but any machine learning approach supported by `MLearn` can be specified. Baseline performance is measured by the percentage of the validation samples classified correctly (a confusion matrix of the results is also returned). If the validationSamples are not specified, a leave-one-out (LOO) approach is deployed, whereby each sample in turn is used as the validation sample, resulting in as many classifiers being trained as there are samples.

Value

A list with three elements:

- `$sigPerformance` is the percentage of validationSamples correctly classified (or, in the LOO case, the percentage of total samples correctly classified by classifiers trained using the remaining samples.)
- `$confusion` is a confusion matrix in the form of a table showing how many samples in each class were correctly or incorrectly classified, corresponding to True Positives, True Negative, False Positives, and False Negatives.
- `$modePerformance` is the percentage of validationSamples correctly classified by a "mode" classifier (or, in the LOO case, the percentage of total samples correctly classified by a "mode" classifier, which is equal the number of samples with the more-frequent category.) The "mode" classifier always predicts the category that appears most often in the training set. If the training set is balanced between categories, one category will always be predicted.

Author(s)

Justin Norden with Rory Stark

See Also

`sigCheck`, `sigCheckRandom`, `sigCheckPermuted`, `sigCheckKnown`, `MLearn`

Examples

```
library(breastCancerNKI)
data(nki)
nki <- nki[,!is.na(nki$e.dmfs)]
data(knownSignatures)
results <- sigCheckClassifier(nki, classes="e.dmfs",
                              signature=knownSignatures$cancer$VANTVEER,
                              annotation="HUGO.gene.symbol")
```

sigCheckKnown *Check classification performance of signature against a panel of known gene signatures*

Description

Compare the classification performance of a known panel of gene signatures to the signature being checked. By default, a panel of gene signatures from Venet et. al. is used.

Usage

```
sigCheckKnown(expressionSet, classes, signature, annotation, validationSamples,
            classifierMethod = svmI, classifierScore, knownSignatures="cancer")
```

Arguments

expressionSet	An ExpressionSet object containing the data to be checked, including an expression matrix, feature labels, and samples.
classes	Specifies which label is to be used to determine the classification categories (must be one of varLabels(expressionSet)). There should be only two unique values in expressionSet$classes.
signature	A vector of feature labels specifying which features comprise the signature to be checked. These feature labels should match values as specified in the annotation parameter (default is row names in the expressionSet). Alternatively, this can be a integer vector of feature indexes.
annotation	Character string specifying which featureData field should be used as the annotation. If missing, the row names of the expressionSet are used as the feature names.
validationSamples	Optional specification, as a vector of sample indices, of what samples in the should used for validation. If present, a classifier will be trained, using the specified signature and classification method, on the non-validation samples, and it's performance evaluated by attempting to classify the validations samples. If missing, a leave-one-out (LOO) validation method will be used, where a separate classifier will be trained to classify each sample using the remaining samples.
classifierMethod	The MLInterfaces learnerSchema object indicating the machine learning method to use for classification. Default is svmI for linear Support Vector Machine classification. See MLearn for available methods.
classifierScore	A performance measure of the baseline classifier. Generally the classifierScore element of the result list returned by sigCheckClassifier. If missing, sigCheckClassifier will be called to establish baseline performance.

knownSignatures
: Either a character string specifying which set of signatures to use from the included sets in knownSignatures, or a list of previously identified signatures to compare performance against. Each element in the list should be a vector of feature labels. Default is to use the "cancer" signatures from the included knownSignatures data set, taken from Venet et. al.

Details

sigCheckClassifier is called for each of the known signatures.

Value

A list with six elements:

- $sigPerformance is the percentage of validationSamples correctly classified (or, in the LOO case, the percentage of total samples correctly classified by classifiers trained using the remaining samples.)
- $modePerformance is the percentage of validationSamples correctly classified by a "mode" classifier (or, in the LOO case, the percentage of total samples correctly classified by a "mode" classifier, which is equal the number of samples with the more-frequent category.) The "mode" classifier always predicts the category that appears most often in the training set. If the training set is balanced between categories, one category will always be predicted.
- $known is a character string indicating which gene signature set was checked. Either one of the sets in knownSignatures, or the string "user specified".
- $knownSigs is the number of signatures evaluated (equal to length(knownSignatures), minus any signatures with zero features matching the labels in expressionSet.)
- $rank is the performance rank of the primary signature classifier on the original dataset amongst the performances of the known signatures on the same dataset.
- $performanceKnown is a vector of performance scores (proportion of the validation set correctly predicted) for each known signature on the dataset.

Author(s)

Justin Norden with Rory Stark

References

Venet, David, Jacques E. Dumont, and Vincent Detours. "Most random gene expression signatures are significantly associated with breast cancer outcome." PLoS Computational Biology 7.10 (2011): e1002240.

See Also

knownSignatures, sigCheck, sigCheckClassifier, sigCheckRandom, sigCheckPermuted, MLearn

Examples

```
library(breastCancerNKI)
data(nki)
nki <- nki[,!is.na(nki$e.dmfs)]
data(knownSignatures)
results <- sigCheckKnown(nki, classes="e.dmfs",
                         signature=knownSignatures$cancer$VANTVEER,
                         annotation="HUGO.gene.symbol",
                         validationSamples=275:319)
```

`sigCheckPermuted` *Check classification performance of signature on randomly permuted data*

Description

Performance of a classification signature on intact data is compared to performance in permuted data, either by feature (expression values of each feature permuted across samples), samples (expression values of all features permuted within each sample), or categories (permuted assignment of samples to classification categories).

Usage

```
sigCheckPermuted(expressionSet, classes, signature,
                 annotation, validationSamples,
                 classifierMethod = svmI, nIterations = 10, classifierScore,
                 toPermute="features")
```

Arguments

`expressionSet`	An `ExpressionSet` object containing the data to be checked, including an expression matrix, feature labels, and samples.
`classes`	Specifies which label is to be used to determine the classification categories (must be one of `varLabels(expressionSet)`). There should be only two unique values in `expressionSet$classes`.
`signature`	A vector of feature labels specifying which features comprise the signature to be checked. These feature labels should match values as specified in the `annotation` parameter (default is row names in the expressionSet). Alternatively, this can be a integer vector of feature indexes.
`annotation`	Character string specifying which `featureData` field should be used as the annotation. If missing, the row names of the expressionSet are used as the feature names.
`validationSamples`	Optional specification, as a vector of sample indices, of what samples in the should used for validation. If present, a classifier will be trained, using the specified signature and classification method, on the non-validation samples, and

its performance evaluated by attempting to classify the validations samples. If missing, a leave-one-out (LOO) validation method will be used, where a separate classifier will be trained to classify each sample using the remaining samples.

`classifierMethod`
: The MLInterfaces learnerSchema object indicating the machine learning method to use for classification. Default is `svmI` for linear Support Vector Machine classification. See `MLearn` for available methods.

`nIterations`
: The number of permutations to test and compare classification outcomes.

`classifierScore`
: A performance measure of the baseline classifier. Generally the `classifierScore` element of the result list returned by `sigCheckClassifier`. If missing, `sigCheckClassifier` will be called to establish baseline performance.

`toPermute`
: Character string or vector of strings indicating what should be permuted. Allowable values:

  - `"features"`: the expression values for each feature will be permuted (permutation by row).
  - `"samples"`: the expression values for each sample will be permuted (permutation by column).
  - `"categories"`: the values in `classes` will be permuted.

Details

Any combination of `permuteFeatures`, `permuteSamples`, and `permuteCategories` can be specified. Performance for each signature is determined by calling `sigCheckClassifier`.

Value

A list with six elements:

- `$sigPerformance` is the percentage of validationSamples correctly classified (or, in the LOO case, the percentage of total samples correctly classified by classifiers trained using the remaining samples.)
- `$modePerformance` is the percentage of validationSamples correctly classified by a "mode" classifier (or, in the LOO case, the percentage of total samples correctly classified by a "mode" classifier, which is equal the number of samples with the more-frequent category.) The "mode" classifier always predicts the category that appears most often in the training set. If the training set is balanced between categories, one category will always be predicted.
- `$permute` is a character string or string of character strings detailing what aspects of the data were permuted (equal to `toPermute`.)
- `$tests` is the number of tests run (equal to `nIterations`.)
- `$rank` is the performance rank of the primary signature classifier on the unpermuted dataset amongst the performance of the signature on permuted datasets.
- `$performancePermuted` is a vector of performance scores (proportion of the validation set correctly predicted) for each permuted dataset.

Author(s)

Justin Norden with Rory Stark

See Also

sigCheck, sigCheckClassifier, sigCheckRandom, sigCheckKnown, MLearn

Examples

```
library(breastCancerNKI)
data(nki)
nki <- nki[,!is.na(nki$e.dmfs)]
data(knownSignatures)
results <- sigCheckPermuted(nki, classes="e.dmfs",
                            signature=knownSignatures$cancer$VANTVEER,
                            annotation="HUGO.gene.symbol",
                            validationSamples=275:319,
                            toPermute="features")
```

---

sigCheckPlot *Plot results of a signature check*

Description

Plots a histogram of the classification performance scores for a check, showing how it compares to classification performance of the signature being checked, as well as to a hypothetical classifier that uses the mode of the training samples.

Usage

```
sigCheckPlot(checkResults, ...)
```

Arguments

checkResults	The list value returned by sigCheckRandom, sigCheckKnown, or sigCheckPermuted (or one the elements of the list returned by sigCheck). Can also be the list returned by sigCheck, which which case each of the four results lists will be plotted in turn.
...	Additional arguments to be passed to the plot function.

Details

Draws a line plot version of a histogram, with the x-axis representing the range of classification performance scores computed in the check, and the y-axis representing how many times that score was obtained. In addition, vertical lines are plotted representing the classification performance of the originally specified signature (solid red line) and the performance of a classifier that always predicts the mode value of the training samples (dotted red line).

Value

none

Note

By default, sigCheck will call this function for all checks it runs.

Author(s)

Rory Stark with Justin Norden

See Also

sigCheck, sigCheckRandom, sigCheckKnown, sigCheckPermuted

Examples

```
library(breastCancerNKI)
data(nki)
nki <- nki[,!is.na(nki$e.dmfs)]
data(knownSignatures)
results <- sigCheckRandom(nki, classes="e.dmfs",
                          signature=knownSignatures$cancer$VANTVEER,
                          annotation="HUGO.gene.symbol",
                          validationSamples=275:319, nIterations=25)
sigCheckPlot(results)
```

---

sigCheckRandom *Check classification performance of signatures composed of randomly selected features*

Description

Performance of a classification signature is compared to signatures composed of the same number of randomly-selected features.

Usage

```
sigCheckRandom(expressionSet, classes, signature,
               annotation, validationSamples,
               classifierMethod = svmI, nIterations = 10, classifierScore)
```

Arguments

expressionSet	An ExpressionSet object containing the data to be checked, including an expression matrix, feature labels, and samples.
classes	Specifies which label is to be used to determine the classification categories (must be one of varLabels(expressionSet)). There should be only two unique values in expressionSet$classes.
signature	A vector of feature labels specifying which features comprise the signature to be checked. These feature labels should match values as specified in the annotation parameter (default is row names in the expressionSet). Alternatively, this can be a integer vector of feature indexes.
annotation	Character string specifying which featureData field should be used as the annotation. If missing, the row names of the expressionSet are used as the feature names.
validationSamples	Optional specification, as a vector of sample indices, of what samples should used for validation. If present, a classifier will be trained, using the specified signature and classification method, on the non-validation samples, and it's performance evaluated by attempting to classify the validations samples. If missing, a leave-one-out (LOO) validation method will be used, where a separate classifier will be trained to classify each sample using the remaining samples.
classifierMethod	The MLInterfaces learnerSchema object indicating the machine learning method to use for classification. Default is svmI for linear Support Vector Machine classification. See MLearn for available methods.
nIterations	The number of permutations to test and compare classification outcomes.
classifierScore	A performance measure of the baseline classifier. Generally the classifierScore element of the result list returned by sigCheckClassifier. If missing, sigCheckClassifier will be called to establish baseline performance.

Details

First, the number of features in the passed signature that match features in the dataset is calculated. Next, nIterations signatures are generated and tested, each consisting of the same number of randomly selected features. Performance for each signature is determined by calling sigCheckClassifier.

Value

A list with five elements:

- $sigPerformance is the percentage of validationSamples correctly classified (or, in the LOO case, the percentage of total samples correctly classified by classifiers trained using the remaining samples.)
- $modePerformance is the percentage of validationSamples correctly classified by a "mode" classifier (or, in the LOO case, the percentage of total samples correctly classified by a "mode" classifier, which is equal the number of samples with the more-frequent category.) The "mode"

classifier always predicts the category that appears most often in the training set. If the training set is balanced between categories, one category will always be predicted.

- `$tests` is the number of tests run (equal to `nIterations`.)
- `$rank` is the performance rank of the primary signature classifier amongst the performance of the random signatures.
- `$performanceRandom` is a vector of performance scores (proportion of the validation set correctly predicted) for each random signature.

Author(s)

Justin Norden with Rory Stark

See Also

`sigCheck`, `sigCheckClassifier`, `sigCheckPermuted`, `sigCheckKnown`, `MLearn`

Examples

```
library(breastCancerNKI)
data(nki)
nki <- nki[,!is.na(nki$e.dmfs)]
data(knownSignatures)
results <- sigCheckRandom(nki, classes="e.dmfs",
                          signature=knownSignatures$cancer$VANTVEER,
                          annotation="HUGO.gene.symbol",
                          validationSamples=275:319)
```

Index