# Package 'classSNitch'

September 9, 2015

**Title** Classifies RNA Structure Change in Chemical Mapping Data

**Version** 0.10.0

**Author** Chanin Tolson [aut, cre]

**Maintainer** Chanin Tolson <ctolson@email.unc.edu>

**URL** http://www.unc.edu/~ctolson/classSNitch/classSNitch.html

**Description** Mutations in RNA will create a riboSNitch, if important structural elements are disrupted.Recent ultra-high throughput techniques, such as SHAPE-MaP and PARS, enable the collection of structural RNA information on a genome-wide scale. With the ability to gather genome-wide structural information on RNA, it is important to accurately classify these structural data in order to identify those structural changes that result in a phenotypic outcome. We have developed an automated approach to classify structure change in SHAPE data. This method utilizes random forest classification on a set of pattern and magnitude parameters from the mutate and map SHAPE data set (or another user specified data set) to build a classifier. The classifier is then used to identify structure change in other SHAPE traces. Enabling scientists to identify structure change may help guide experiments that examine RNA structure and its role in biological processes.

**Depends** R (>= 3.2.2)

**License** GPL (>= 2)

**LazyData** true

**Imports** randomForest,
ROCR,
gplots,
changepoint,
dtw

## R topics documented:

---

`classifyRNA` *classifyRNA*

---

## Description

A function to build a classifier for RNA structure change

## Usage

```
classifyRNA(data=NULL, classes=2)
```

## Arguments

| | |
|---|---|
| `data` | Optional data to build the classifier. Default is pre-loaded data. |
| `classes` | An optional number indicating which class style to use. Only used when data is not supplied. Default is 2. |

## Details

This function builds a random forest classifier for RNA structure change in SHAPE data

This function builds a random forest classifier for RNA structure change using the randomForest package.

## Value

A classifyRNA object, based on randomForest object (see randomForest package)

**call** The original call to randomForest

**type** One of regression, classification, or unsupervised.

**predicted** The predicted values of the input data based on out-of-bag samples.

**importance** A matrix with nclass + 2 columns. The first nclass columns are the class-specific measures computed as mean descrease in accuracy. The nclass + 1st column is the mean descrease in accuracy over all classes. The last column is the mean decrease in Gini index.

**importanceSD** The standard errors of the permutation-based importance measure. A p by nclass + 1 matrix corresponding to the first nclass + 1 columns of the importance matrix.

**ntree** Number of trees grown.

**mtry** Number of predictors sampled for spliting at each node.

**forest** A list that contains the entire forest

**err.rate** Vector error rates of the prediction on the input data, the i-th element being the (OOB) error rate for all trees up to the i-th.

**confusion** The confusion matrix of the prediction (based on OOB data).

**votes** A matrix with one row for each input data point and one column for each class, giving the fraction or number of (OOB) votes from the random forest.

**oob.times** Number of times cases are out-of-bag (and thus used in computing OOB error estimate)

**proximity** A matrix of proximity measures among the input (based on the frequency that pairs of data points are in the same terminal nodes).

## Note

Organization of the data file: header=TRUE, tab-delimited .txt file

- "column 1" RNA name
- "column 2" Classification
- "column 3" Magnitude predictor
- "column 4" Pattern predictor

Options for classes:

- "1" none v. local v. global
- "2" none v. local/global
- "3" global v. none/local
- "4" local v. global

The default data has been gathered from the RNA Mapping Database mutate and map experiments.

## Author(s)

Chanin Tolson

## References

A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. R News 2(3), 18–22 (randomForest package)

RNA Mapping Database

## See Also

`getChangeParams` `predict.classifyRNA`

## Examples

```
#build classifier
rf = classifyRNA(classes=2)
#get confusion matrix
rf$confusion
```

`classify_default` *classify_default*

## Description

classify_default

## References

RNA Mapping Database

`classSNitch` *Package for the autonomous classification of RNA structure change*

## Description

Mutations in RNA will create a riboSNitch, if important structural elements are disrupted. Recent ultra-high throughput techniques, such as SHAPE-MaP and PARS, enable the collection of structural RNA information on a genome-wide scale. With the ability to gather genome-wide structural information on RNA, it is important to accurately classify these structural data in order to identify those structural changes that result in a phenotypic outcome. We have developed an automated approach to classify structure change in SHAPE data. This method utilizes random forest classification on a set of pattern and magnitude parameters from the mutate-and-map SHAPE data set (or another user specified data set) to build a classifier. The classifier is then used to identify structure change in other SHAPE traces. Enabling scientists to identify structure change may help guide experiments that examine RNA structure and its role in biological processes.

## Author(s)

Chanin Tolson

## References

A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. R News 2(3), 18–22 (randomForest package)

RNA Mapping Database

## See Also

`getChangeParams` `classifyRNA`

## Examples

```
#get change parameters
library("ROCR")
library("gplots")

data("shape_ex")
sample_shape = shape_ex
sample = getChangeParams(sample_shape[2:nrow(sample_shape),], base=sample_shape[1,], trim=5)

#predict change
data("magpat_ex")
sample_class = magpat_ex
cr = classifyRNA(sample_class, classes=2)
cr_pred = predict(cr, sample, type="response")

#plot ROC curve
col = 2
data("mutmap")
data = mutmap
data = data[-which(is.na(data[,col]),arr.ind=TRUE),]
predobj = prediction(cr$votes[,2], data[,col])
perfobj = performance(predobj, tpr, fpr)
aucobj = performance(predobj, auc)
plot(perfobj@x.values[[1]], perfobj@y.values[[1]], lwd=2,
     type="l", xlab="Specificity", ylab="Sensitivity")
points(c(-1,2),c(-1,2), col="red", type="l")
text(0.8, 0.2, paste("AUC: ", format(aucobj@y.values, digits=2), sep=""), cex=1)
```

---

`getChangeParams` *getChangeParams*

---

## Description

A function to get magnitude and pattern change parameters

## Usage

```
getChangeParams(sample, base=NULL, margin=1, trim=0, high=NULL,
   tol=0.1, point=rep(0,nrow(sample)), window=ncol(sample), outfile=NULL, append=F)
```

## Arguments

| | |
|---|---|
| `sample` | A numeric matrix containing values to be compared (e.g. a set of mutant SHAPE traces). |
| `base` | An optional numeric vector containing the values to which the samples are to be compared (e.g. a wild type SHAPE trace). Default is the first trace in each file. |
| `margin` | An optional number indicating if the samples are organized by rows or columns, where 1 indicates rows and 2 indicates columns. Default is 1. |
| `trim` | An optional number indicating the number of nucleotides to be trimed from the ends. Default is 0. |
| `high` | An optional number indicating the reactivity above which reactivities are considered high. Default is third quartile of the sample in each file. |

| | |
|---|---|
| tol | An optional number indicating the tolerance for the change. Default is 0.1. |
| point | An optional numerical vector indicating the location of disruption (e.g. mutation point) |
| window | An optional number indicating the number of columns around the disruption to calculate. Default is the entire trace. |
| outfile | An optional string indicating the name of the output file. The output file will consist of two columns (magnitude change and pattern change). Default will not output a file. |
| append | An optional boolean to append the file if an outfile is given. Default is FALSE. |

## Details

This calculates the magnitude and pattern change in SHAPE reactivity.

This function normalizes and reduces the noise in the sample. The magnitude, pattern, location, timewarp and trace change are calculated for the sample using the magnitudeChange, patternChange, locationChange, timewarpChange and traceChange functions.

## Value

**"outmat"** A three column numeric matrix for magnitude, pattern, location and timewarp change.

**"outfile"** An optional output file for the matrix.

## Author(s)

Chanin Tolson

## See Also

magnitudeChange patternChange normalize reduceNoise classifyRNA predict.classifyRNA locationChange timewarpChange traceChange

## Examples

```
#input files
data("shape_ex")
#get change parameters
params = getChangeParams(shape_ex, trim=5, outfile="out.txt")
```

---

# locationChange *locationChange*

---

## Description

A function to get the location change between samples

## Usage

```
locationChange(sample, point=rep(0,nrow(sample)), base=sample[1,], margin=1)
```

### Arguments

| | |
|---|---|
| sample | A numeric matrix containing values to be compared (e.g. a set of mutant SHAPE traces). |
| point | An optional numeric vector containing the location of the disruption (e.g. the mutation in an RNA) |
| base | An optional numeric vector containing the values to which the samples are to be compared (e.g. a wildtype SHAPE trace). Default is the first trace in sample. |
| margin | An optional number indicating if the samples are organized by rows or columns, where 1 indicates rows and 2 indicates columns. Default is 1. |

### Details

This function determines

This function calculates the distance of change from the disruption using changepoint analysis to determine the location of change.

### Value

A numeric vector of location changes.

### Author(s)

Chanin Tolson

### See Also

getChangeParams

### Examples

```
#sample data
sample = matrix(sample(1:100), ncol=10)
#normalize
samp_norm = normalize(sample)
#reduce noise
samp_nreduce = reduceNoise(samp_norm, trim=1, high=4)
#get location change
loc = locationChange(samp_nreduce)
```

---

## magnitudeChange *magnitudeChange*

---

### Description

A function to get the magnitude change between samples

### Usage

```
magnitudeChange(sample, base=sample[1,], margin=1)
```

## Arguments

| | |
|---|---|
| sample | A numeric matrix containing values to be compared (e.g. a set of mutant SHAPE traces). |
| base | An optional numeric vector containing the values to which the samples are to be compared (e.g. a wildtype SHAPE trace). Default is the first trace in sample. |
| margin | An optional number indicating if the samples are organized by rows or columns, where 1 indicates rows and 2 indicates columns. Default is 1. |

## Details

This function compares the magnitude change between samples.

This function calculates the Pearson correlation coefficient between the base vector and each row (or column) in sample.

## Value

A numeric vector of magnitude changes.

## Author(s)

Chanin Tolson

## See Also

getChangeParams

## Examples

```
#sample data
sample = matrix(sample(1:100), ncol=10)
#normalize
samp_norm = normalize(sample)
#reduce noise
samp_nreduce = reduceNoise(samp_norm, trim=1, high=4)
#get magnitude change
mag = magnitudeChange(samp_nreduce)
```

---

# magpat_ex *Magnitude and Pattern Change in example RNAs*

---

## Description

Example dataset with the calculated magnitude, pattern, location and timewarp change for a set of RNAs. SHAPE traces from the RNA Mapping Database. Parameters calculated using getChangeParameters function.

## Usage

```
magpat_ex
```

### Format

```
data.frame: 2670 obs. of  6 variables:
 $ class           : int  1 1 NA 1 1 1 NA NA NA 1 ...
 $ magnitude.change: num  0.0947 0.0949 0.0949 0.0938 0.0942 ...
 $ pattern.change  : num  0.0821 0.0886 0.087 0.0879 0.0862 ...
 $ location.change : num  0 0 0 0 0 ...
 $ timewarp.change : num  2.47 2.02 2.13 2.49 2.3 ...
 $ trace.change    : num  2.52 2.81 3.42 3.6 4.23 ...
```

### References

RNA Mapping Database

### Examples

```
data("magpat_ex")
```

---

## mutmap — *Consensus classification, magnitude change, and pattern change in example RNAs*

---

### Description

Example dataset with the calculated magnitude, pattern, location and timewarp change for a set of RNAs. SHAPE traces from the RNA Mapping Database. Parameters calculated using getChangeParameters function. Consensus classification determined using crowd-sourced manual classification.

### Usage

```
mutmap
```

### Format

```
data.frame: 2670 obs. of  12 variables:
 $ class           : int  1 1 NA 1 1 1 NA NA NA 1 ...
 $ magnitude.change: int  1 1 NA 1 1 1 NA NA NA 1 ...
 $ pattern.change  : int  0 0 NA 0 0 0 NA NA NA 0 ...
 $ location.change : int  NA NA NA NA NA NA NA NA NA NA ...
 $ timewarp.change : num  1 NA 0.75 1 1 NA NA NA 0.75 NA ...
 $ trace.change    : num  0 0 NA 0 0 0 NA NA NA 0 ...
 $ NA              : num  NA NA NA NA NA NA NA NA NA NA ...
 $ NA              : num  0.0947 0.0949 0.0949 0.0938 0.0942 ...
 $ NA              : num  0.0821 0.0886 0.087 0.0879 0.0862 ...
 $ NA              : num  0 0 0 0 0 ...
 $ NA              : num  2.47 2.02 2.13 2.49 2.3 ...
 $ NA              : num  2.52 2.81 3.42 3.6 4.23 ...
```

### References

RNA Mapping Database

## Examples

```
data("mutmap")
```

---

`normalize` *normalize*

---

## Description

A Between-Sample Normalization Function

## Usage

```
normalize(sample, base=sample[1,], margin=1)
```

## Arguments

| | |
|---|---|
| `sample` | A numeric matrix containing values to be normalized (e.g. a set of mutant SHAPE traces). |
| `base` | An optional numeric vector containing the values to which the sample is to be normalized (e.g. a wild type SHAPE trace). Default is the first trace in sample. |
| `margin` | An optional number indicating if sample is organized by rows or columns, where 1 indicates rows and 2 indicates columns. Default is 1. |

## Details

This function performs between-sample normalization.

This function normalizes the average value of the base vector to 1.5. Each row (or column) in sample is then normalized by minimizing the absolute difference between the base and the sample row (or column).

## Value

A normalized numeric matrix with the same dimensions as sample.

## Author(s)

Chanin Tolson

## See Also

`getChangeParams`

## Examples

```
#sample data
sample = matrix(sample(1:100), ncol=10)
#normalize
samp_norm = normalize(sample)

#sample data
sample = matrix(sample(1:100), ncol=10)
base = sample(1:100, size=10)
```

```
#normalize
samp_norm = normalize(sample, base)
```

---

patternChange *patternChange*

---

## Description

A function to get the pattern change between samples

## Usage

```
patternChange(sample, base=sample[1,], margin=1, tol=0.1)
```

## Arguments

| | |
|---|---|
| sample | A numeric matrix containing values to be compared (e.g. a set of mutant SHAPE traces). |
| base | An optional numeric vector containing the value to which the samples are to be compared (e.g. a wild type SHAPE trace). Default is the first trace in sample. |
| margin | An optional number indicating if the samples are organized by rows or columns, where 1 indicates rows and 2 indicates columns. Default is 1. |
| tol | An optional number indicating the tolerance for the change. Default is 0.1. |

## Details

This function compares the pattern change between samples.

The pattern for a single SHAPE reacitivity trace is the pattern of increase in reactivity or decrease in reactivity between nucleotides. If the change is less than the tolerance value, it is considered a none change. The pattern change value is the Pearson correlation coefficient between the base vector pattern and the pattern of each row (or column) in sample.

## Value

A numeric vector of pattern changes.

## Author(s)

Chanin Tolson

## See Also

getChangeParams

## Examples

```
#sample data
sample = matrix(sample(1:100), ncol=10)
#normalize
samp_norm = normalize(sample)
#reduce noise
samp_nreduce = reduceNoise(samp_norm, trim=1, high=4)
#get patterne change
pat = patternChange(samp_nreduce)
```

---

`predict.classifyRNA` *predict.classifyRNA*

---

## Description

A function to classify rna structure change

## Usage

```
## S3 method for class classifyRNA
predict(object, sample = NULL, ...)
```

## Arguments

| | |
|---|---|
| `object` | An object of classifyRNA (see classifyRNA function). |
| `sample` | An optional matrix of predictors for magnitude, pattern, location, timewarp and trace change (e.g. output from getChangeParams()) |
| `...` | Further arguments passed to or from other methods. |

## Details

This function predicts RNA structure change in SHAPE data

This function predicts RNA structure change in SHAPE data using a random forest classifier.

## Value

A list of "response", "vote" or "prob" predictions.

**"response"** Predicted classes (classes with majority vote)

**"vote"** Vote count fraction (one column for each class and one row for each input)

**"class"** Class probabilities (one column for each class and one row for each input)

## Author(s)

Chanin Tolson

## References

A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. R News 2(3), 18–22 (randomForest package)

## See Also

getChangeParams classifyRNA

## Examples

```
#input data
data("magpat_ex")
#build classifier
cr = classifyRNA(classes=2)
#get prediction
cr_pred = predict(cr, magpat_ex[,2:6])
```

---

reduceNoise *reduceNoise*

---

## Description

A function to reduce noise in SHAPE data

## Usage

```
reduceNoise(sample, base=sample[1,], margin=1, trim=0, high=boxplot(sample)$stats[4])
```

## Arguments

| | |
|---|---|
| sample | A numeric matrix containing reactivity scores to be compared (e.g. a set of mutant SHAPE traces). |
| base | An optional numeric vector containing the reactivity score to which the samples are to be compared (e.g. a wild type SHAPE trace). Default is the first trace in sample. |
| margin | An optional number indicating if the samples are organized by rows or columns, where 1 indicates rows and 2 indicates columns. Default is 1. |
| trim | An optional number indicating the number of nucleotides to be trimmed from the each end. Default is 0. |
| high | An optional number indicating the value above which reactivities are considered high. Default is third quartile of sample. |

## Details

This function removes peaks that are high in both the sample and the comparison SHAPE traces.

This function reduces the noise in SHAPE data. For positions where both the base vector and the sample row (or column) is above the high value, the position in the sample row (or column) is set equal to that position in the base vector. The function trims the data by setting the ends of sample equal to the ends of the base vector.

## Value

A noise reduced numeric matrix with the same dimensions as sample.

## Author(s)

Chanin Tolson

## See Also

getChangeParams

## Examples

```
#sample data
sample = matrix(sample(1:100), ncol=10)
#normalize
samp_norm = normalize(sample)
#reduce noise
samp_nreduce = reduceNoise(samp_norm, trim=1, high=4)
```

---

shape_ex *16S rRNA four-way junction wild-type and mutant SHAPE data*

---

## Description

Example SHAPE traces from 16SFWJ_1M7_0001 Mutate and Map experiment in the RNA Mapping Database.

## Usage

```
shape_ex
```

## Format

```
 num [1:111, 1:110] 18.89 11.57 3.69 13.7 10.68 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:111] "16SFWJ_1M7_0001_WT" "16SFWJ_1M7_0001_G126C" "16SFWJ_1M7_0001_G127C" "16SFWJ
  ..$ : chr [1:110] "V3" "V4" "V5" "V6" ...
```

## References

RNA Mapping Database

## Examples

```
data("shape_ex")
```

`timewarpChange` *timewarpChange*

## Description

A function to get the timewarp between samples

## Usage

```
timewarpChange(sample, base=sample[1,], margin=1)
```

## Arguments

| | |
|---|---|
| `sample` | A numeric matrix containing values to be compared (e.g. a set of mutant SHAPE traces). |
| `base` | An optional numeric vector containing the values to which the samples are to be compared (e.g. a wildtype SHAPE trace). Default is the first trace in sample. |
| `margin` | An optional number indicating if the samples are organized by rows or columns, where 1 indicates rows and 2 indicates columns. Default is 1. |

## Details

This function compares the timewarp between samples.

This function calculates the timewarp between the base vector and each row (or column) in sample.

## Value

A numeric vector of timewarp changes.

## Author(s)

Chanin Tolson

## See Also

`getChangeParams`

## Examples

```
#sample data
sample = matrix(sample(1:100), ncol=10)
#normalize
samp_norm = normalize(sample)
#reduce noise
samp_nreduce = reduceNoise(samp_norm, trim=1, high=4)
#get magnitude change
tw = timewarpChange(samp_nreduce)
```

traceChange *traceChange*

## Description

A function to get the average trace between samples

## Usage

```
traceChange(sample, base=sample[1,], margin=1,
   point=rep(0,nrow(sample)), window=ncol(sample))
```

## Arguments

| | |
|---|---|
| sample | A numeric matrix containing values to be compared (e.g. a set of mutant SHAPE traces). |
| base | An optional numeric vector containing the values to which the samples are to be compared (e.g. a wildtype SHAPE trace). Default is the first trace in sample. |
| margin | An optional number indicating if the samples are organized by rows or columns, where 1 indicates rows and 2 indicates columns. Default is 1. |
| point | An optional numeric vector containing the location of the disruption (e.g. the mutation in an RNA) |
| window | An optional number indicating the number of columns around the disruption to calculate. Default is the entire trace. |

## Details

This function compares the average trace difference between samples.

This function calculates the average trace difference between the base vector and each row (or column) in sample.

## Value

A numeric vector of trace changes.

## Author(s)

Chanin Tolson

## See Also

getChangeParams

## Examples

```
#sample data
sample = matrix(sample(1:100), ncol=10)
#normalize
samp_norm = normalize(sample)
#reduce noise
samp_nreduce = reduceNoise(samp_norm, trim=1, high=4)
```

```
#get trace change
tc = traceChange(samp_nreduce)
```

# Index