Package ‘jaccard’

June 14, 2018

Type Package

Title Test Similarity Between Binary Data using Jaccard/Tanimoto Coefficients

Version 0.1.0

Date 2018-06-06

Author Neo Christopher Chung <nchchung@gmail.com>, Błażej Miasojedow <bmiasojedow@gmail.com>, Michał Startek <M.Startek@mimuw.edu.pl>, Anna Gambin <aniag@mimuw.edu.pl>

Maintainer Neo Christopher Chung <nchchung@gmail.com>

Description Calculate statistical significance of Jaccard/Tanimoto similarity coefficients for binary data.

biocViews

License GPL-2

Encoding UTF-8

LazyData true

Imports Rcpp (>= 0.12.6), qvalue, dplyr, magrittr

LinkingTo Rcpp

NeedsCompilation yes

SystemRequirements C++11

RoxygenNote 6.0.1

Repository CRAN

Date/Publication 2018-06-14 17:53:00 UTC

R topics documented:

jaccard ................................................................. 2
jaccard.ev ............................................................ 3
jaccard.rahman ....................................................... 3
jaccard.test ............................................................ 4
**jaccard**

Compute a Jaccard/Tanimoto similarity coefficient

**Usage**

\[
jaccard(x, y, center = FALSE, px = NULL, py = NULL)
\]

**Arguments**

- **x**: a binary vector (e.g., fingerprint)
- **y**: a binary vector (e.g., fingerprint)
- **center**: whether to center the Jaccard/Tanimoto coefficient by its expectation
- **px**: probability of successes in \( x \) (optional)
- **py**: probability of successes in \( y \) (optional)

**Value**

\( jaccard \) returns a Jaccard/Tanimoto coefficient.

**Examples**

```r
set.seed(1234)
x = rbinom(100, 1, .5)
y = rbinom(100, 1, .5)
jaccard(x, y)
```
Compute an expected Jaccard/Tanimoto similarity coefficient under independence

**Description**

Compute an expected Jaccard/Tanimoto similarity coefficient under independence.

**Usage**

```
jaccard.ev(x, y, px = NULL, py = NULL)
```

**Arguments**

- `x`: a binary vector (e.g., fingerprint)
- `y`: a binary vector (e.g., fingerprint)
- `px`: probability of successes in `x` (optional)
- `py`: probability of successes in `y` (optional)

**Value**

`jaccard.ev` returns an expected value.

**Examples**

```
set.seed(1234)
x = rbinom(100,1,.5)
y = rbinom(100,1,.5)
jaccard.ev(x,y)
```

Compute p-value using an extreme value distribution

**Description**

Rahman et al. (2014) proposes a method to compute a p-value of a Jaccard/Tanimoto coefficient using an extreme value distribution. Their paper provides the following description: The mean (mu) and s.d. (sigma) of the similarity scores are used to define the z score, \( z = (T - \mu) / \sigma \). For the purpose of calculating the P value, only hits with \( T > 0 \) are considered. The P value \( w \) is derived from the z score using an extreme value distribution \( P = 1 - \exp(-e^{-z}\pi/\sqrt{6}) - G'(1) \), where the Euler-Mascheroni constant \( G'(1) = 0.577215665 \).

**Usage**

```
jaccard.rahman(j)
```
jaccard.test

Arguments

j a numeric vector of observed Jaccard coefficients (uncentered)

Value

jaccard.rahman returns a numeric vector of p-values

References


jaccard.test Test for Jaccard/Tanimoto similarity coefficients

Description

Compute statistical significance of Jaccard/Tanimoto similarity coefficients between binary vectors, using four different methods.

Usage

jaccard.test(x, y, method = "mca", px = NULL, py = NULL, verbose = TRUE, ...)

Arguments

x a binary vector (e.g., fingerprint)
y a binary vector (e.g., fingerprint)
method a method to compute a p-value ("mca", "bootstrap", "asymptotic", or "exact")
px probability of successes in x (optional)
py probability of successes in y (optional)
verbose whether to print progress messages
... optional arguments for specific computational methods

Details

There exist four methods to compute p-values of Jaccard/Tanimoto similarity coefficients: mca, bootstrap, asymptotic, and exact. This is simply a wrapper function for corresponding four functions in this package: jaccard.test.mca, jaccard.test.bootstrap, jaccard.test.asymptotic, and jaccard.test.exact.

We recommend using either mca or bootstrap methods, since the exact solution is slow for a moderately large vector and asymptotic approximation may be inaccurate depending on the input vector size. The bootstrap method uses resampling with replacement binary vectors to compute a
The `mca` method uses the measure concentration algorithm that estimates the multinomial distribution with a known error bound (specified by an optional argument `accuracy`).

### Value

`jaccard.test` returns a list mainly consisting of

- `statistics`: centered Jaccard/Tanimoto similarity coefficient
- `pvalue`: p-value
- `expectation`: expectation

### Optional arguments for method="bootstrap"

- `fix`: whether to fix (i.e., not resample) `x` and/or `y`
- `B`: a total bootstrap iteration
- `seed`: a seed for a random number generator

### Optional arguments for method="mca"

- `accuracy`: an error bound on approximating a multinomial distribution
- `error.type`: an error type on approximating a multinomial distribution ("average", "upper", "lower")
- `seed`: a seed for the random number generator.

### See Also

`jaccard.test.bootstrap` `jaccard.test.mca` `jaccard.test.exact` `jaccard.test.asymptotic`

### Examples

```r
set.seed(1234)
x = rbinom(100L, 1L, .5)
y = rbinom(100L, 1L, .5)
jaccard.test(x, y, method="bootstrap")
jaccard.test(x, y, method="mca")
jaccard.test(x, y, method="exact")
jaccard.test(x, y, method="asymptotic")
```
jaccard.test.asymptotic

Compute p-value using an asymptotic approximation

Description

Compute statistical significance of Jaccard/Tanimoto similarity coefficients.

Usage

jaccard.test.asymptotic(x, y, px = NULL, py = NULL, verbose = TRUE)

Arguments

x       a binary vector (e.g., fingerprint)
y       a binary vector (e.g., fingerprint)
px      probability of successes in x (optional)
py      probability of successes in y (optional)
verbose whether to print progress messages

Value

jaccard.test.asymptotic returns a list consisting of

statistics centered Jaccard/Tanimoto similarity coefficient
pvalue   p-value
expectation expectation

Examples

set.seed(1234)
x = rbinom(100,1,.5)
y = rbinom(100,1,.5)
jaccard.test.asymptotic(x,y)
jaccard.test.bootstrap

Compute p-value using the bootstrap procedure

Description

Compute statistical significance of Jaccard/Tanimoto similarity coefficients.

Usage

jaccard.test.bootstrap(x, y, px = NULL, py = NULL, verbose = TRUE, fix = "x", B = 1000, seed = NULL)

Arguments

x      a binary vector (e.g., fingerprint)
y      a binary vector (e.g., fingerprint)
px     probability of successes in x (optional)
py     probability of successes in y (optional)
verbose whether to print progress messages
fix     whether to fix (i.e., not resample) x and/or y
B      a total bootstrap iteration
seed   a seed for a random number generator

Value

jaccard.test.bootstrap returns a list consisting of

statistics  centered Jaccard/Tanimoto similarity coefficient
pvalue      p-value
expectation expectation

Examples

set.seed(1234)
x = rbinom(100,1,.5)
y = rbinom(100,1,.5)
jaccard.test.bootstrap(x,y,B=500)
### jaccard.test.exact

**Compute p-value using the exact solution**

**Description**

Compute statistical significance of Jaccard/Tanimoto similarity coefficients.

**Usage**

```r
jaccard.test.exact(x, y, px = NULL, py = NULL, verbose = true)
```

**Arguments**

- `x`: a binary vector (e.g., fingerprint)
- `y`: a binary vector (e.g., fingerprint)
- `px`: probability of successes in `x` (optional)
- `py`: probability of successes in `y` (optional)
- `verbose`: whether to print progress messages

**Value**

`jaccard.test.exact` returns a list consisting of:

- `statistics`: centered Jaccard/Tanimoto similarity coefficient
- `pvalue`: p-value
- `expectation`: expectation

**Examples**

```r
set.seed(1234)
x = rbinom(100,1,.5)
y = rbinom(100,1,.5)
jaccard.test.exact(x,y)
```

### jaccard.test.mca

**Compute p-value using the Measure Concentration Algorithm**

**Description**

Compute statistical significance of Jaccard/Tanimoto similarity coefficients.

**Usage**

```r
jaccard.test.mca(x, y, px = NULL, py = NULL, accuracy = 1e-05, error.type = "average", verbose = TRUE)
```
**jaccard.test.pairwise**

**Arguments**

- **x**: a binary vector (e.g., fingerprint)
- **y**: a binary vector (e.g., fingerprint)
- **px**: probability of successes in x (optional)
- **py**: probability of successes in y (optional)
- **accuracy**: an error bound on approximating a multinomial distribution
- **error.type**: an error type on approximating a multinomial distribution ("average", "upper", "lower")
- **verbose**: whether to print progress messages

**Value**

- **jaccard.test.mca** returns a list consisting of
  - **statistics**: centered Jaccard/Tanimoto similarity coefficient
  - **pvalue**: p-value
  - **expectation**: expectation

**Examples**

```r
set.seed(1234)
x = rbinom(100,1,.5)
y = rbinom(100,1,.5)
jaccard.test.mca(x,y,accuracy = 1e-05)
```

---

**jaccard.test.pairwise**  *Pair-wise tests for Jaccard/Tanimoto similarity coefficients*

**Description**

Given a data matrix, it computes pair-wise Jaccard/Tanimoto similarity coefficients and p-values among rows (variables). For fine controls, use "jaccard.test".

**Usage**

```r
jaccard.test.pairwise(dat, method = "mca", verbose = TRUE,
compute.qvalue = TRUE, ...)
```

**Arguments**

- **dat**: a data matrix
- **method**: a method to compute a p-value ("mca", "bootstrap", "asymptotic", or "exact")
- **verbose**: whether to print progress messages
- **compute.qvalue**: whether to compute q-values
- **...**: optional arguments for specific computational methods
jaccard.test.pairwise

Value

jaccard.test.pairwise returns a list of matrices

- statistics: Jaccard/Tanimoto similarity coefficients
- pvalues: p-values
- qvalues: q-values

See Also

jaccard.test
Index

jaccard, 2
jaccard.ev, 3
jaccard.rahman, 3
jaccard.test, 4, 10
jaccard.test.asymptotic, 4, 5, 6
jaccard.test.bootstrap, 4, 5, 7
jaccard.test.exact, 4, 5, 8
jaccard.test.mca, 4, 5, 8
jaccard.test.pairwise, 9