

# Package ‘partitionComparison’

December 19, 2017

**Type** Package

**Title** Implements Measures for the Comparison of Two Partitions

**Version** 0.2.3

**Date** 2017-12-19

**Author** Fabian Ball [aut, cre, cph, ctb],  
Andreas Geyer-Schulz [cph]

**Maintainer** Fabian Ball <fabian.ball@kit.edu>

**Description** Provides several measures ((dis)similarity, distance/metric, correlation, entropy) for comparing two partitions of the same set of objects. The different measures can be assigned to three different classes: Pair comparison (containing the famous Jaccard and Rand indices), set based, and information theory based. Many of the implemented measures can be found in Albatineh AN, Niewiadomska-Bugaj M and Mihalko D (2006) <doi:10.1007/s00357-006-0017-z> and Meila M (2007) <doi:10.1016/j.jmva.2006.11.013>. Partitions are represented by vectors of class labels which allow a straightforward integration with existing clustering algorithms (e.g. kmeans()). The package is mostly based on the S4 object system.

**Depends** R (>= 3.2.0)

**License** MIT + file LICENSE

**RoxygenNote** 6.0.1

**RdMacros** Rdpack

**Imports** methods, Rdpack, lpSolve

**Suggests** testthat

## R topics documented:

partitionComparison-package . . . . .	3
adjustedRandIndex . . . . .	4
baulieu1 . . . . .	5
baulieu2 . . . . .	6
classificationErrorDistance . . . . .	7
compareAll . . . . .	8
computePairCoefficients . . . . .	9
czekanowski . . . . .	10

dongensMetric	11
entropy	12
fagerMcGowan	13
folwkesMallovsIndex	14
gammaStatistics	15
goodmanKruskal	16
gowerLegendre	17
hamann	18
jaccardCoefficient	19
kulczynski	20
larsenAone	21
lermanIndex	22
mcconnaughey	23
minkowskiMeasure	24
mirkinMetric	25
mutualInformation	26
N	27
N00	27
N01	28
N01p	28
N10	29
N10p	29
N11	30
N12	30
N21	31
normalizedLermanIndex	31
normalizedMutualInformation	32
PairCoefficients-class	33
Partition-class	34
pearson	34
peirce	35
projectionNumber	36
randIndex	37
registerPartitionVectorSignatures	38
rogersTanimoto	38
russelRao	39
rvCoefficient	40
sokalSneath1	41
sokalSneath2	42
sokalSneath3	43
variationOfInformation	44
wallaceI	45
wallaceII	46
[<-,Partition-method	47

---

partitionComparison-package

*partitionComparison: Implements Measures for the Comparison of Two Partitions*

---

## Description

Provides several measures ((dis)similarity, distance/metric, correlation, entropy) for comparing two partitions of the same set of objects. The different measures can be assigned to three different classes: Pair comparison (containing the famous Jaccard and Rand indices), set based, and information theory based. Many of the implemented measures can be found in Albatineh AN, Niewiadomska-Bugaj M and Mihalko D (2006) <doi:10.1007/s00357-006-0017-z> and Meila M (2007) <doi:10.1016/j.jmva.2006.11.013>. Partitions are represented by vectors of class labels which allow a straightforward integration with existing clustering algorithms (e.g. kmeans()). The package is mostly based on the S4 object system.

## Details

This package provides a large collection of measures to compare two partitions. Some survey articles for these measures are cited below, the seminal papers for each individual measure is provided with the function definition.

Most functionality is implemented as S4 classes and methods so that an adoption is easily possible for special needs and specifications. The main class is `Partition` which merely wraps an atomic vector of length  $n$  for storing the class label of each object. The computation of all measures is designed to work on vectors of class labels.

All partition comparison methods can be called in the same way: `<measure method>(p, q)` with `p`, `q` being the two partitions (as `Partition` instances). One often does not explicitly want to transform the vector of class labels (as output of another package's function/algorithm) into `Partition` instances before using measures from this package. For convenience, the function `registerPartitionVectorSignatures` exists which dynamically creates versions of all measures that will directly work with plain R vectors.

## Author(s)

**Maintainer:** Fabian Ball <fabian.ball@kit.edu> [copyright holder, contributor]

Other contributors:

- Andreas Geyer-Schulz <andreas.geyer-schulz@kit.edu> [copyright holder]

## References

Albatineh AN, Niewiadomska-Bugaj M and Mihalko D (2006). "On Similarity Indices and Correction for Chance Agreement." *Journal of Classification*, **23**(2), pp. 301 – 313. ISSN 0176-4268, doi: [10.1007/s003570060017z](https://doi.org/10.1007/s003570060017z).

Meila M (2007). "Comparing Clusterings – an Information Based Distance." *Journal of Multivariate Analysis*, **98**(5), pp. 873 – 895. doi: [10.1016/j.jmva.2006.11.013](https://doi.org/10.1016/j.jmva.2006.11.013).

## Examples

```
# Generate some data
set.seed(42)
data <- cbind(x=c(rnorm(50), rnorm(30, mean=5)), y=c(rnorm(50), rnorm(30, mean=5)))
# Run k-means with two/three centers
data.km2 <- kmeans(data, 2)
data.km3 <- kmeans(data, 3)

# Load this library
library(partitionComparison)
# Register the measures to take ANY input
registerPartitionVectorSignatures(environment())
# Compare the clusters
randIndex(data.km2$cluster, data.km3$cluster)
# [1] 0.8101266
```

---

adjustedRandIndex	<i>Adjusted Rand Index</i>
-------------------	----------------------------

---

## Description

Compute the Adjusted Rand Index (ARI)

$$\frac{2(N_{00}N_{11} - N_{10}N_{01})}{N'_{01}N_{12} + N'_{10}N_{21}}$$

## Usage

```
adjustedRandIndex(p, q)

## S4 method for signature 'Partition,Partition'
adjustedRandIndex(p, q)

## S4 method for signature 'PairCoefficients,missing'
adjustedRandIndex(p, q = NULL)
```

## Arguments

p	The partition $P$ or an instance of <a href="#">PairCoefficients</a>
q	The partition $Q$ or NULL

## Methods (by class)

- p = Partition, q = Partition: Compute given two partitions
- p = PairCoefficients, q = missing: Compute given the pair coefficients

## Author(s)

Fabian Ball <fabian.ball@kit.edu>

## References

Hubert L and Arabie P (1985). “Comparing Partitions.” *Journal of Classification*, **2**(1), pp. 193 – 218.

## Examples

```
isTRUE(all.equal(adjustedRandIndex(new("Partition", c(0, 0, 0, 1, 1)),
                                   new("Partition", c(0, 0, 1, 1, 1))), 1/6))
```

---

baulieu1	<i>Baulieu Index 1</i>
----------	------------------------

---

## Description

Compute the index 1 of Baulieu

$$\frac{N^2 - N(N_{10} + N_{01}) + (N_{10} - N_{01})^2}{N^2}$$

## Usage

```
baulieu1(p, q)

## S4 method for signature 'Partition,Partition'
baulieu1(p, q)

## S4 method for signature 'PairCoefficients,missing'
baulieu1(p, q = NULL)
```

## Arguments

p                      The partition  $P$  or an instance of [PairCoefficients](#)  
q                        The partition  $Q$  or NULL

## Methods (by class)

- p = Partition, q = Partition: Compute given two partitions
- p = PairCoefficients, q = missing: Compute given the pair coefficients

## Author(s)

Fabian Ball <fabian.ball@kit.edu>

## References

Baulieu FB (1989). “A Classification of Presence/Absence Based Dissimilarity Coefficients.” *Journal of Classification*, **6**(1), pp. 233–246. ISSN 0176-4268, 1432-1343, doi: [10.1007/BF01908601](https://doi.org/10.1007/BF01908601).

## Examples

```
isTRUE(all.equal(baulieu1(new("Partition", c(0, 0, 0, 1, 1)),
                          new("Partition", c(0, 0, 1, 1, 1))), 0.76))
```

baulieu2

*Baulieu Index 2***Description**

Compute the index 2 of Baulieu

$$\frac{N_{11}N_{00} - N_{10}N_{01}}{N^2}$$

**Usage**

```
baulieu2(p, q)
```

```
## S4 method for signature 'Partition,Partition'
baulieu2(p, q)
```

```
## S4 method for signature 'PairCoefficients,missing'
baulieu2(p, q = NULL)
```

**Arguments**

p                      The partition  $P$  or an instance of [PairCoefficients](#)

q                      The partition  $Q$  or NULL

**Methods (by class)**

- p = Partition, q = Partition: Compute given two partitions
- p = PairCoefficients, q = missing: Compute given the pair coefficients

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

**References**

Baulieu FB (1989). “A Classification of Presence/Absence Based Dissimilarity Coefficients.” *Journal of Classification*, **6**(1), pp. 233–246. ISSN 0176-4268, 1432-1343, doi: [10.1007/BF01908601](https://doi.org/10.1007/BF01908601).

**Examples**

```
isTRUE(all.equal(baulieu2(new("Partition", c(0, 0, 0, 1, 1)),
                          new("Partition", c(0, 0, 1, 1, 1))), 0.04))
```

---

classificationErrorDistance  
*Classification Error Distance*

---

### Description

Compute the classification error distance

$$1 - \frac{1}{n} \max_{\sigma} \sum_{C \in \mathcal{P}} |C \cap \sigma(C)|$$

with  $\sigma$  a weighted matching between the clusters of both partitions. The nodes are the classes of each partition, the weights are the overlap of objects.

### Usage

```
classificationErrorDistance(p, q)
```

```
## S4 method for signature 'Partition,Partition'
classificationErrorDistance(p, q)
```

### Arguments

p	The partition $P$
q	The partition $Q$

### Methods (by class)

- p = Partition, q = Partition: Compute given two partitions

### Hint

This measure is implemented using [lp.assign](#) from the lpSolve package to compute the maximal matching of a weighted bipartite graph.

### Author(s)

Fabian Ball <fabian.ball@kit.edu>

### References

Meila M and Heckerman D (2001). “An Experimental Comparison of Model-Based Clustering Methods.” *Machine Learning*, **42**(1), pp. 9 – 29.

Meila M (2005). “Comparing Clusterings: An Axiomatic View.” In *Proceedings of the 22Nd International Conference on Machine Learning*, series ICML '05, pp. 577–584. ISBN 978-1-59593-180-1, doi: [10.1145/1102351.1102424](https://doi.org/10.1145/1102351.1102424).

### Examples

```
isTRUE(all.equal(classificationErrorDistance(new("Partition", c(0, 0, 0, 1, 1)),
                                              new("Partition", c(0, 0, 1, 1, 1))), 0.2))
```

---

compareAll

---

*Compare two partitions with all measures*


---

## Description

Compute the comparison between two partitions for all available measures.

## Usage

```
compareAll(p, q)
```

```
## S4 method for signature 'Partition,Partition'
```

```
compareAll(p, q)
```

## Arguments

p                      The partition  $P$

q                      The partition  $Q$

## Value

Instance of `data.frame` with columns measure and value

## Methods (by class)

- `p = Partition, q = Partition`: Compare given two `Partition` instances

## Warning

This method will identify every generic S4 method that has a signature "Partition", "Partition" (including signatures with following "missing" parameters, e.g. "Partition", "Partition", "missing") as a partition comparison measure, **except** this method itself (otherwise: infinite recursion). This means one has to take care when defining other methods with the same signature in order not to produce unwanted side-effects!

## Author(s)

Fabian Ball <fabian.ball@kit.edu>

## Examples

```
compareAll(new("Partition", c(0, 0, 0, 1, 1)), new("Partition", c(0, 0, 1, 1, 1)))
## Not run:
```

	measure	value
1	adjustedRandIndex	0.166666667
2	baulieu1	0.760000000
3	baulieu2	0.040000000
4	classificationErrorDistance	0.200000000
5	czekanowski	0.500000000
6	dongensMetric	2.000000000
7	fagerMcGowan	0.250000000
8	folwkesMallowsIndex	0.500000000
9	gammaStatistics	0.166666667



```

10         goodmanKruskal 0.33333333
11         gowerLegendre 0.75000000
12         hamann 0.20000000
13         jaccardCoefficient 0.33333333
14         kulczynski 0.50000000
15         larsenAone 0.80000000
16         lermanIndex 0.43643578
17         mcconnaughey 0.00000000
18         minkowskiMeasure 1.00000000
19         mirkinMetric 8.00000000
20         mutualInformation 0.29110316
21         normalizedLermanIndex 0.16666667
22         normalizedMutualInformation 0.43253806
23         pearson 0.00694444
24         peirce 0.16666667
25         randIndex 0.60000000
26         rogersTanimoto 0.42857142
27         russelRao 0.20000000
28         rvCoefficient 0.69230769
29         sokalSneath1 0.58333333
30         sokalSneath2 0.20000000
31         sokalSneath3 0.33333333
32         variationOfInformation 0.76381700
33         wallaceI 0.50000000
34         wallaceII 0.50000000

```

```
## End(Not run)
```

---

```
computePairCoefficients
```

*Compute the four coefficients  $N_{11}$ ,  $N_{10}$ ,  $N_{01}$ ,  $N_{00}$*

---

## Description

Given two object partitions  $P$  and  $Q$ , of same length  $n$ , each of them described as a vector of cluster ids, compute the four coefficients ( $N_{11}$ ,  $N_{10}$ ,  $N_{01}$ ,  $N_{00}$ ) all of the pair comparison measures are based on.

## Usage

```
computePairCoefficients(p, q)
```

## Arguments

<code>p</code>	The partition $P$
<code>q</code>	The partition $Q$

## Author(s)

Fabian Ball <fabian.ball@kit.edu>

**Examples**

```
pc <- computePairCoefficients(new("Partition", c(0, 0, 0, 1, 1)),
                             new("Partition", c(0, 0, 1, 1, 1)))
isTRUE(all.equal(N11(pc), 2))
isTRUE(all.equal(N10(pc), 2))
isTRUE(all.equal(N01(pc), 2))
isTRUE(all.equal(N00(pc), 4))
```

czezanowski

*Czezanowski Index***Description**

Compute the Czezanowski index

$$\frac{2N_{11}}{2N_{11} + N_{10} + N_{01}}$$

**Usage**

```
czezanowski(p, q)

## S4 method for signature 'Partition,Partition'
czezanowski(p, q)

## S4 method for signature 'PairCoefficients,missing'
czezanowski(p, q = NULL)
```

**Arguments**

p                      The partition  $P$  or an instance of [PairCoefficients](#)  
q                        The partition  $Q$  or NULL

**Methods (by class)**

- p = Partition, q = Partition: Compute given two partitions
- p = PairCoefficients, q = missing: Compute given the pair coefficients

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

**References**

Czezanowski J (1932). "Coefficient of Racial Likeness" Und „Durchschnittliche Differenz." *Anthropologischer Anzeiger*, **9**(3/4), pp. 227–249.

**Examples**

```
isTRUE(all.equal(czezanowski(new("Partition", c(0, 0, 0, 1, 1)),
                             new("Partition", c(0, 0, 1, 1, 1))), 0.5))
```

---

dongensMetric	<i>Dongen's Metric</i>
---------------	------------------------

---

**Description**

Compute Dongen's metric

$$2n - \sum_{C \in P} \max_{D \in Q} |C \cap D| - \sum_{D \in Q} \max_{C \in P} |C \cap D|$$

**Usage**

```
dongensMetric(p, q)
```

```
## S4 method for signature 'Partition,Partition'
dongensMetric(p, q)
```

**Arguments**

p	The partition $P$
q	The partition $Q$

**Methods (by class)**

- p = Partition, q = Partition: Compute given two partitions

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

**References**

van Dongen S (2000). "Performance Criteria For Graph Clustering And Markov Cluster Experiments." Technical Report INS-R 0012, CWI.

**See Also**

[projectionNumber](#)

**Examples**

```
isTRUE(all.equal(dongensMetric(new("Partition", c(0, 0, 0, 1, 1)),
                                new("Partition", c(0, 0, 1, 1, 1))), 2))
```

entropy

*Entropy***Description**

Compute the Shannon entropy

$$-\sum_i p_i \log_b p_i$$

**Usage**

```
entropy(x, log_base)
```

```
## S4 method for signature 'numeric,numeric'
entropy(x, log_base)
```

```
## S4 method for signature 'Partition,numeric'
entropy(x, log_base)
```

```
## S4 method for signature 'ANY,missing'
entropy(x, log_base = exp(1))
```

**Arguments**

x	A probability distribution
log_base	Optional base of the logarithm (default: <i>e</i> )

**Methods (by class)**

- x = Partition, log\_base = numeric: Entropy of a partition represented by x

**Hint**

This method is used internally for measures based on information theory

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

**Examples**

```
isTRUE(all.equal(entropy(c(.5, .5)), log(2)))
isTRUE(all.equal(entropy(c(.5, .5), 2), 1))
isTRUE(all.equal(entropy(c(.5, .5), 4), .5))

# Entropy of a partition
isTRUE(all.equal(entropy(new("Partition", c(0, 0, 1, 1, 1))), entropy(c(2/5, 3/5))))
```

fagerMcGowan

*Fager & McGowan Index***Description**

Compute the index of Fager and McGowan

$$\frac{N_{11}}{\sqrt{N_{21}N_{12}}} - \frac{1}{2\sqrt{N_{21}}}$$

**Usage**

```
fagerMcGowan(p, q)
```

```
## S4 method for signature 'Partition,Partition'
fagerMcGowan(p, q)
```

```
## S4 method for signature 'PairCoefficients,missing'
fagerMcGowan(p, q = NULL)
```

**Arguments**

p                      The partition  $P$  or an instance of [PairCoefficients](#)

q                      The partition  $Q$  or NULL

**Methods (by class)**

- p = Partition, q = Partition: Compute given two partitions
- p = PairCoefficients, q = missing: Compute given the pair coefficients

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

**References**

Fager EW and McGowan JA (1963). “Zooplankton Species Groups in the North Pacific Co-Occurrences of Species Can Be Used to Derive Groups Whose Members React Similarly to Water-Mass Types.” *Science*, **140**(3566), pp. 453–460.

**Examples**

```
isTRUE(all.equal(fagerMcGowan(new("Partition", c(0, 0, 0, 1, 1)),
                             new("Partition", c(0, 0, 1, 1, 1))), 0.25))
```

---

folwkesMallowsIndex	<i>Folwkes &amp; Mallows Index</i>
---------------------	------------------------------------

---

## Description

Compute the index of Folwkes and Mallows

$$\sqrt{\frac{N_{11}}{N_{21}} \frac{N_{11}}{N_{12}}}$$

which is a combination of the two Wallace indices.

## Usage

```
folwkesMallowsIndex(p, q)
```

```
## S4 method for signature 'Partition,Partition'
```

```
folwkesMallowsIndex(p, q)
```

```
## S4 method for signature 'PairCoefficients,missing'
```

```
folwkesMallowsIndex(p, q = NULL)
```

## Arguments

p	The partition $P$ or an instance of <a href="#">PairCoefficients</a>
q	The partition $Q$ or NULL

## Methods (by class)

- p = Partition, q = Partition: Compute given two partitions
- p = PairCoefficients, q = missing: Compute given the pair coefficients

## Author(s)

Fabian Ball <fabian.ball@kit.edu>

## References

Fowlkes EB and Mallows CL (1983). “A Method for Comparing Two Hierarchical Clusterings.” *Journal of the American Statistical Association*, **78**(383), pp. 553 – 569.

## See Also

[wallaceI](#) [wallaceII](#)

## Examples

```
isTRUE(all.equal(folwkesMallowsIndex(new("Partition", c(0, 0, 0, 1, 1)),
                                     new("Partition", c(0, 0, 1, 1, 1))), 0.5))
```

---

gammaStatistics
Gamma Statistics

---

## Description

Compute the Gamma statistics

$$\frac{N_{11}N_{00} - N_{10}N_{01}}{\sqrt{N_{21}N_{12}N'_{10}N'_{01}}}$$

## Usage

```
gammaStatistics(p, q)
```

```
## S4 method for signature 'Partition,Partition'
```

```
gammaStatistics(p, q)
```

```
## S4 method for signature 'PairCoefficients,missing'
```

```
gammaStatistics(p, q = NULL)
```

## Arguments

p                      The partition  $P$  or an instance of [PairCoefficients](#)

q                        The partition  $Q$  or NULL

## Methods (by class)

- p = Partition, q = Partition: Compute given two partitions
- p = PairCoefficients, q = missing: Compute given the pair coefficients

## Author(s)

Fabian Ball <fabian.ball@kit.edu>

## References

Yule GU (1900). “On the Association of Attributes in Statistics: With Illustrations from the Material of the Childhood Society, \&c.” *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, **194**, pp. 257–319.

## Examples

```
isTRUE(all.equal(gammaStatistics(new("Partition", c(0, 0, 0, 1, 1)),
                                new("Partition", c(0, 0, 1, 1, 1))), 1/6))
```

goodmanKruskal

*Goodman & Kruskal Index***Description**

Compute the index of Goodman and Kruskal

$$\frac{N_{11}N_{00} - N_{10}N_{01}}{N_{11}N_{00} + N_{10}N_{01}}$$

**Usage**

```
goodmanKruskal(p, q)
```

```
## S4 method for signature 'Partition,Partition'
goodmanKruskal(p, q)
```

```
## S4 method for signature 'PairCoefficients,missing'
goodmanKruskal(p, q)
```

**Arguments**

p                      The partition  $P$  or an instance of [PairCoefficients](#)

q                      The partition  $Q$  or NULL

**Methods (by class)**

- p = Partition, q = Partition: Compute given two partitions
- p = PairCoefficients, q = missing: Compute given the pair coefficients

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

**References**

Goodman LA and Kruskal WH (1954). “Measures of Association for Cross Classifications.” *Journal of the American Statistical Association*, **49**(268), pp. 732–764. ISSN 0162-1459, doi: [10.1080/01621459.1954.10501231](https://doi.org/10.1080/01621459.1954.10501231).

**Examples**

```
isTRUE(all.equal(goodmanKruskal(new("Partition", c(0, 0, 0, 1, 1)),
                                new("Partition", c(0, 0, 1, 1, 1))), 1/3))
```



gowerLegendre

*Gower & Legendre Index***Description**

Compute the index of Gower and Legendre

$$\frac{N_{11} + N_{00}}{N_{11} + \frac{1}{2}(N_{10} + N_{01}) + N_{00}}$$

**Usage**

```
gowerLegendre(p, q)
```

```
## S4 method for signature 'Partition,Partition'
```

```
gowerLegendre(p, q)
```

```
## S4 method for signature 'PairCoefficients,missing'
```

```
gowerLegendre(p, q)
```

**Arguments**

p                      The partition  $P$  or an instance of [PairCoefficients](#)

q                      The partition  $Q$  or NULL

**Methods (by class)**

- p = Partition, q = Partition: Compute given two partitions
- p = PairCoefficients, q = missing: Compute given the pair coefficients

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

**References**

Gower JC and Legendre P (1986). “Metric and Euclidean Properties of Dissimilarity Coefficients.” *Journal of Classification*, **3**(1), pp. 5–48. ISSN 0176-4268, 1432-1343, doi: [10.1007/BF01896809](https://doi.org/10.1007/BF01896809).

**Examples**

```
isTRUE(all.equal(gowerLegendre(new("Partition", c(0, 0, 0, 1, 1)),
                                new("Partition", c(0, 0, 1, 1, 1))), 0.75))
```

---

hamann	<i>Hamann Coefficient</i>
--------	---------------------------

---

## Description

Compute the Hamann coefficient

$$\frac{(N_{11} + N_{00}) - (N_{10} + N_{01})}{N}$$

## Usage

```
hamann(p, q)

## S4 method for signature 'Partition,Partition'
hamann(p, q)

## S4 method for signature 'PairCoefficients,missing'
hamann(p, q = NULL)
```

## Arguments

p	The partition $P$ or an instance of <a href="#">PairCoefficients</a>
q	The partition $Q$ or NULL

## Methods (by class)

- p = Partition, q = Partition: Compute given two partitions
- p = PairCoefficients, q = missing: Compute given the pair coefficients

## Author(s)

Fabian Ball <fabian.ball@kit.edu>

## References

Hamann U (1961). “Merkmalsbestand Und Verwandtschaftsbeziehungen Der Farinosae: Ein Beitrag Zum System Der Monokotyledonen.” *Willdenowia*, **2**(5), pp. 639–768. ISSN 0511-9618.

## Examples

```
isTRUE(all.equal(hamann(new("Partition", c(0, 0, 0, 1, 1)),
                        new("Partition", c(0, 0, 1, 1, 1))), 0.2))
```

---

jaccardCoefficient	<i>Jaccard Coefficient</i>
--------------------	----------------------------

---

## Description

Compute the Jaccard coefficient

$$\frac{N_{11}}{N_{11} + N_{10} + N_{01}}$$

## Usage

```
jaccardCoefficient(p, q)

## S4 method for signature 'Partition,Partition'
jaccardCoefficient(p, q)

## S4 method for signature 'PairCoefficients,missing'
jaccardCoefficient(p, q = NULL)
```

## Arguments

p	The partition $P$ or an instance of <a href="#">PairCoefficients</a>
q	The partition $Q$ or NULL

## Methods (by class)

- p = Partition, q = Partition: Compute given two partitions
- p = PairCoefficients, q = missing: Compute given the pair coefficients

## Author(s)

Fabian Ball <fabian.ball@kit.edu>

## References

Jaccard P (1908). “Nouvelles Recherches Sur La Distribution Florale.” *Bulletin de la Société Vaudoise des Sciences Naturelles*, **44**(163), pp. 223 – 270.

## Examples

```
isTRUE(all.equal(jaccardCoefficient(new("Partition", c(0, 0, 0, 1, 1)),
                                     new("Partition", c(0, 0, 1, 1, 1))), 1/3))
```

---

 kulczynski

*Kulczynski Index*


---

### Description

Compute the Kulczynski index

$$\frac{1}{2} \left( \frac{N_{11}}{N_{21}} + \frac{N_{11}}{N_{12}} \right)$$

### Usage

```
kulczynski(p, q)
```

```
## S4 method for signature 'Partition,Partition'
kulczynski(p, q)
```

```
## S4 method for signature 'PairCoefficients,missing'
kulczynski(p, q = NULL)
```

### Arguments

p                      The partition  $P$  or an instance of [PairCoefficients](#)

q                      The partition  $Q$  or NULL

### Methods (by class)

- p = Partition, q = Partition: Compute given two partitions
- p = PairCoefficients, q = missing: Compute given the pair coefficients

### Author(s)

Fabian Ball <fabian.ball@kit.edu>

### References

Kulczynski S (1927). “Zespoly Roslin w Pieninach.” *Bull. Intern. Acad. Pol. Sci. Lett. Cl. Sci. Math. Nat., B (Sci. Nat.)*, **1927**(Suppl 2), pp. 57–203.

### Examples

```
isTRUE(all.equal(kulczynski(new("Partition", c(0, 0, 0, 1, 1)),
                             new("Partition", c(0, 0, 1, 1, 1))), 0.5))
```

larsenAone

*Larsen & Aone Measure***Description**

Compute the measure of Larsen and Aone

$$\frac{1}{|\mathcal{P}|} \sum_{C \in \mathcal{P}} \max_{D \in \mathcal{Q}} \frac{2|C \cap D|}{|C| + |D|}$$

**Usage**

```
larsenAone(p, q)
```

```
## S4 method for signature 'Partition,Partition'
larsenAone(p, q)
```

**Arguments**

p                      The partition  $P$

q                        The partition  $Q$

**Methods (by class)**

- p = Partition, q = Partition: Compute given two partitions

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

**References**

Larsen B and Aone C (1999). “Fast and Effective Text Mining Using Linear-Time Document Clustering.” In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, series KDD '99, pp. 16–22. ISBN 1-58113-143-7, doi: [10.1145/312129.312186](https://doi.org/10.1145/312129.312186).

**Examples**

```
isTRUE(all.equal(larsenAone(new("Partition", c(0, 0, 0, 1, 1)),
                             new("Partition", c(0, 0, 1, 1, 1))), 0.8))
```

lermanIndex

*Lerman Index***Description**

Compute the Lerman index

$$\frac{N_{11} - E(N_{11})}{\sqrt{\sigma^2(N_{11})}}$$

**Usage**

```
lermanIndex(p, q, c = NULL)
```

```
## S4 method for signature 'Partition,Partition,missing'
lermanIndex(p, q, c = NULL)
```

```
## S4 method for signature 'Partition,Partition,PairCoefficients'
lermanIndex(p, q, c = NULL)
```

**Arguments**

p                      The partition  $P$

q                      The partition  $Q$

c                      [PairCoefficients](#) or NULL

**Methods (by class)**

- p = Partition, q = Partition, c = missing: Compute given two partitions
- p = Partition, q = Partition, c = PairCoefficients: Compute given the partitions and pair coefficients

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

**References**

- Lerman IC (1988). “Comparing Partitions (Mathematical and Statistical Aspects).” In Bock H (ed.), *Classification and Related Methods of Data Analysis*, pp. 121 – 132.
- Hubert L and Arabie P (1985). “Comparing Partitions.” *Journal of Classification*, **2**(1), pp. 193 – 218.
- Denœud L and Guénoche A (2006). “Comparison of Distance Indices Between Partitions.” In Batagelj V, Bock H, Ferligoj A and Žiberna A (eds.), *Data Science and Classification*, series Studies in Classification, Data Analysis, and Knowledge Organization, pp. 21–28. Springer Berlin Heidelberg. ISBN 978-3-540-34415-5 978-3-540-34416-2.

**See Also**

[normalizedLermanIndex](#)

**Examples**

```
isTRUE(all.equal(lermanIndex(new("Partition", c(0, 0, 0, 1, 1)),
                             new("Partition", c(0, 0, 1, 1, 1))), 2/sqrt(21)))
```

---

mcconnaughey	<i>McConnaughey Index</i>
--------------	---------------------------

---

**Description**

Compute the McConnaughey index

$$\frac{N_{11}^2 - N_{10}N_{01}}{N_{21}N_{12}}$$

**Usage**

```
mcconnaughey(p, q)

## S4 method for signature 'Partition,Partition'
mcconnaughey(p, q)

## S4 method for signature 'PairCoefficients,missing'
mcconnaughey(p, q = NULL)
```

**Arguments**

p                      The partition  $P$  or an instance of [PairCoefficients](#)  
q                        The partition  $Q$  or NULL

**Methods (by class)**

- p = Partition, q = Partition: Compute given two partitions
- p = PairCoefficients, q = missing: Compute given the pair coefficients

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

**References**

McConnaughey BH and Laut LP (1964). *The Determination and Analysis of Plankton Communities*. Lembaga Penelitian Laut.

**Examples**

```
isTRUE(all.equal(mcconnaughey(new("Partition", c(0, 0, 0, 1, 1)),
                             new("Partition", c(0, 0, 1, 1, 1))), 0))
```

---

minkowskiMeasure	<i>Minkowski Measure</i>
------------------	--------------------------

---

## Description

Compute the Minkowski measure

$$\sqrt{\frac{N_{10} + N_{01}}{N_{11} + N_{10}}}$$

## Usage

```
minkowskiMeasure(p, q)

## S4 method for signature 'Partition,Partition'
minkowskiMeasure(p, q)

## S4 method for signature 'PairCoefficients,missing'
minkowskiMeasure(p, q = NULL)
```

## Arguments

p	The partition $P$ or an instance of <a href="#">PairCoefficients</a>
q	The partition $Q$ or NULL

## Methods (by class)

- p = Partition, q = Partition: Compute given two partitions
- p = PairCoefficients, q = missing: Compute given the pair coefficients

## Author(s)

Fabian Ball <fabian.ball@kit.edu>

## References

Minkowski H (1911). *Gesammelte Abhandlungen von Hermann Minkowski, Zweiter Band*, number 2. B. G. Teubner, Leipzig, Berlin.

## Examples

```
isTRUE(all.equal(minkowskiMeasure(new("Partition", c(0, 0, 0, 1, 1)),
                                   new("Partition", c(0, 0, 1, 1, 1))), 1))
```



---

mirkinMetric	<i>Mirkin Metric</i>
--------------	----------------------

---

## Description

Compute the Mirkin metric

$$2(N_{10} + N_{01})$$

## Usage

```
mirkinMetric(p, q)

## S4 method for signature 'Partition,Partition'
mirkinMetric(p, q)

## S4 method for signature 'PairCoefficients,missing'
mirkinMetric(p, q = NULL)
```

## Arguments

p	The partition $P$ or an instance of <a href="#">PairCoefficients</a>
q	The partition $Q$ or NULL

## Methods (by class)

- p = Partition, q = Partition: Compute given two partitions
- p = PairCoefficients, q = missing: Compute given the pair coefficients

## Author(s)

Fabian Ball <fabian.ball@kit.edu>

## References

Mirkin BG and Chernyi LB (1970). “Measurement of the Distance Between Partitions of a Finite Set of Objects.” *Automation and Remote Control*, **31**(5), pp. 786 – 792.

## Examples

```
isTRUE(all.equal(mirkinMetric(new("Partition", c(0, 0, 0, 1, 1)),
                               new("Partition", c(0, 0, 1, 1, 1))), 8))
```

---

mutualInformation	<i>Mutual Information</i>
-------------------	---------------------------

---

**Description**

Compute the mutual information

$$\sum_{C \in P} \sum_{D \in Q} \frac{|C \cap D|}{n} \log n \frac{|C \cap D|}{|C||D|}$$

**Usage**

```
mutualInformation(p, q)
```

```
## S4 method for signature 'Partition,Partition'
mutualInformation(p, q)
```

**Arguments**

p	The partition $P$
q	The partition $Q$

**Methods (by class)**

- p = Partition, q = Partition: Compute given two partitions

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

**References**

Vinh NX, Epps J and Bailey J (2010). “Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance.” *Journal of Machine Learning Research*, **11**, pp. 2837 – 2854.

**See Also**

[normalizedMutualInformation](#)

**Examples**

```
isTRUE(all.equal(mutualInformation(new("Partition", c(0, 0, 0, 1, 1)),
  new("Partition", c(0, 0, 1, 1, 1))), 4/5*log(5/3) + 1/5*log(5/9)))
```

---

N	<i>Method to retrieve the complex coefficient <math>N</math></i>
---	--

---

**Description**

It is defined as  $N = N_{11} + N_{10} + N_{01} + N_{00}$  which equals  $\binom{n}{2}$  with  $n$  the number of objects

**Usage**

```
N(obj)

## S4 method for signature 'PairCoefficients'
N(obj)
```

**Arguments**

obj                      Instance of [PairCoefficients](#)

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

---

N00	<i>Method to retrieve the coefficient <math>N_{00}</math></i>
-----	---

---

**Description**

Method to retrieve the coefficient  $N_{00}$

**Usage**

```
N00(obj)

## S4 method for signature 'PairCoefficients'
N00(obj)
```

**Arguments**

obj                      Instance of [PairCoefficients](#)

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

---

N01	<i>Method to retrieve the coefficient <math>N_{01}</math></i>
-----	---

---

**Description**

Method to retrieve the coefficient  $N_{01}$

**Usage**

N01(obj)

```
## S4 method for signature 'PairCoefficients'
N01(obj)
```

**Arguments**

obj                      Instance of [PairCoefficients](#)

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

---

N01p	<i>Method to retrieve the complex coefficient <math>N'_{01}</math></i>
------	--

---

**Description**

It is defined as  $N'_{01} = N_{00} + N_{01}$

**Usage**

N01p(obj)

```
## S4 method for signature 'PairCoefficients'
N01p(obj)
```

**Arguments**

obj                      Instance of [PairCoefficients](#)

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

---

N10	<i>Method to retrieve the coefficient <math>N_{10}</math></i>
-----	---

---

**Description**

Method to retrieve the coefficient  $N_{10}$

**Usage**

```
N10(obj)

## S4 method for signature 'PairCoefficients'
N10(obj)
```

**Arguments**

obj                      Instance of [PairCoefficients](#)

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

---

N10p	<i>Method to retrieve the complex coefficient <math>N'_{10}</math></i>
------	--

---

**Description**

It is defined as  $N'_{10} = N_{00} + N_{10}$

**Usage**

```
N10p(obj)

## S4 method for signature 'PairCoefficients'
N10p(obj)
```

**Arguments**

obj                      Instance of [PairCoefficients](#)

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

---

N11	<i>Method to retrieve the coefficient <math>N_{11}</math></i>
-----	---

---

**Description**

Method to retrieve the coefficient  $N_{11}$

**Usage**

`N11(obj)`

```
## S4 method for signature 'PairCoefficients'
N11(obj)
```

**Arguments**

`obj` Instance of [PairCoefficients](#)

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

---

N12	<i>Method to retrieve the complex coefficient <math>N_{12}</math></i>
-----	---

---

**Description**

It is defined as  $N_{12} = N_{11} + N_{01}$

**Usage**

`N12(obj)`

```
## S4 method for signature 'PairCoefficients'
N12(obj)
```

**Arguments**

`obj` Instance of [PairCoefficients](#)

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

N21

*Method to retrieve the complex coefficient  $N_{21}$* **Description**

It is defined as  $N_{21} = N_{11} + N_{10}$

**Usage**

N21(obj)

```
## S4 method for signature 'PairCoefficients'
N21(obj)
```

**Arguments**

obj                      Instance of [PairCoefficients](#)

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

normalizedLermanIndex *Normalized Lerman Index***Description**

Compute the normalized Lerman index

$$L(P, Q) / \sqrt{L(P, P)L(Q, Q)}$$

where  $L$  is the Lerman index.

**Usage**

normalizedLermanIndex(p, q, c = NULL)

```
## S4 method for signature 'Partition,Partition,missing'
normalizedLermanIndex(p, q, c = NULL)
```

```
## S4 method for signature 'Partition,Partition,PairCoefficients'
normalizedLermanIndex(p, q,
  c = NULL)
```

**Arguments**

p                      The partition  $P$   
 q                      The partition  $Q$   
 c                      [PairCoefficients](#) or NULL

**Methods (by class)**

- p = Partition, q = Partition, c = missing: Compute given two partitions
- p = Partition, q = Partition, c = PairCoefficients: Compute given the partitions and pair coefficients

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

**References**

Lerman IC (1988). “Comparing Partitions (Mathematical and Statistical Aspects).” In Bock H (ed.), *Classification and Related Methods of Data Analysis*, pp. 121 – 132.

Hubert L and Arabie P (1985). “Comparing Partitions.” *Journal of Classification*, **2**(1), pp. 193 – 218.

**See Also**

[lermanIndex](#)

**Examples**

```
isTRUE(all.equal(normalizedLermanIndex(new("Partition", c(0, 0, 0, 1, 1)),
                                         new("Partition", c(0, 0, 1, 1, 1))), 1/6))
```

---

normalizedMutualInformation

*Normalized Mutual Information*

---

**Description**

Compute the mutual information (*MI*) which is normalized either by the minimum/maximum partition entropy (*H*)

$$\frac{MI(P, Q)}{\varphi(H(P), H(Q))}, \varphi \in \{\min, \max\}$$

or the sum

$$\frac{2 \cdot MI(P, Q)}{H(P) + H(Q)}$$

**Usage**

```
normalizedMutualInformation(p, q, type = c("min", "max", "sum"))
```

```
## S4 method for signature 'Partition,Partition,character'
normalizedMutualInformation(p, q,
  type = c("min", "max", "sum"))
```

```
## S4 method for signature 'Partition,Partition,missing'
normalizedMutualInformation(p, q,
  type = NULL)
```



**Arguments**

p	The partition $P$
q	The partition $Q$
type	One of "min" (default), "max" or "sum"

**Methods (by class)**

- p = Partition, q = Partition, type = character: Compute given two partitions
- p = Partition, q = Partition, type = missing: Compute given two partitions with type="min"

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

**References**

Kvalseth TO (1987). "Entropy and Correlation: Some Comments." *IEEE Transactions on Systems, Man and Cybernetics*, **17**(3), pp. 517 – 519. ISSN 0018-9472, doi: [10.1109/TSMC.1987.4309069](https://doi.org/10.1109/TSMC.1987.4309069).

**See Also**

[mutualInformation](#), [entropy](#)

**Examples**

```
isTRUE(all.equal(normalizedMutualInformation(
  new("Partition", c(0, 0, 0, 1, 1)),
  new("Partition", c(0, 0, 1, 1, 1)), "min"),
normalizedMutualInformation(
  new("Partition", c(0, 0, 0, 1, 1)),
  new("Partition", c(0, 0, 1, 1, 1)), "max")
))
```

---

PairCoefficients-class

*S4 class to represent coefficients of object pairs for the comparison of two object partitions (say  $P$  and  $Q$ ).*

---

**Description**

S4 class to represent coefficients of object pairs for the comparison of two object partitions (say  $P$  and  $Q$ ).

**Slots**

- N11 The number of object pairs that are in both partitions together in a cluster
- N00 The number of object pairs that are in no partition together in a cluster
- N10 The number of object pairs that are only in partition  $P$  together in a cluster
- N01 The number of object pairs that are only in partition  $Q$  together in a cluster

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

**See Also**

[N11 N10 N01 N00](#)

---

Partition-class	<i>Simple S4 class to represent a partition of objects as vector of class labels.</i>
-----------------	---

---

**Description**

This class is a wrapper around a vector but allows only the atomic vectors logical, numeric, integer, complex, character, raw. The reason for this is that only those types seem to make sense as class labels. Furthermore, class labels are immutable.

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

**Examples**

```
p <- new("Partition", c(0, 0, 1, 1, 1))
q <- new("Partition", c("a", "a", "b", "b", "b"))

## Not run:
# This won't work:
new("Partition", c(list("a"), "a", "b", "b", "b"))
p[2] <- 2

## End(Not run)
```

---

pearson	<i>Pearson Index</i>
---------	----------------------

---

**Description**

Compute the Pearson index

$$\frac{N_{11}N_{00} - N_{10}N_{01}}{N_{21}N_{12}N'_{01}N'_{10}}$$

**Usage**

```
pearson(p, q)

## S4 method for signature 'Partition,Partition'
pearson(p, q)

## S4 method for signature 'PairCoefficients,missing'
pearson(p, q)
```

**Arguments**

- p                      The partition  $P$  or an instance of [PairCoefficients](#)
- q                      The partition  $Q$  or NULL

**Methods (by class)**

- p = Partition, q = Partition: Compute given two partitions
- p = PairCoefficients, q = missing: Compute given the pair coefficients

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

**References**

Pearson K (1926). “On the Coefficient of Racial Likeness.” *Biometrika*, **18**(1/2), pp. 105–117. ISSN 0006-3444, doi: [10.2307/2332498](https://doi.org/10.2307/2332498).

**Examples**

```
isTRUE(all.equal(pearson(new("Partition", c(0, 0, 0, 1, 1)),
                        new("Partition", c(0, 0, 1, 1, 1))), 1/144))
```

---

peirce	<i>Peirce Index</i>
--------	---------------------

---

**Description**

Compute the Peirce index

$$\frac{N_{11}N_{00} - N_{10}N_{01}}{N_{21}N'_{01}}$$

**Usage**

```
peirce(p, q)

## S4 method for signature 'Partition,Partition'
peirce(p, q)

## S4 method for signature 'PairCoefficients,missing'
peirce(p, q = NULL)
```

**Arguments**

- p                      The partition  $P$  or an instance of [PairCoefficients](#)
- q                      The partition  $Q$  or NULL

**Methods (by class)**

- p = Partition, q = Partition: Compute given two partitions
- p = PairCoefficients, q = missing: Compute given the pair coefficients

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

**References**

Peirce CS (1884). “The Numerical Measure of the Success of Predictions.” *Science*, **4**(93), pp. 453–454.

**Examples**

```
isTRUE(all.equal(peirce(new("Partition", c(0, 0, 0, 1, 1)),
                        new("Partition", c(0, 0, 1, 1, 1))), 1/6))
```

---

projectionNumber

*Compute the projection number of two partitions*

---

**Description**

Given two partitions (p, q) represented as vectors of cluster ids, compute the projection number which is the sum of maximum cluster overlaps for all clusters of  $P$  to any cluster of  $Q$ .

**Usage**

```
projectionNumber(p, q)
```

**Arguments**

p	Partition $P$
q	Partition $Q$

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

**See Also**

[dongensMetric](#)

**Examples**

```
isTRUE(all.equal(projectionNumber(c(0, 0, 0, 1, 1), c(0, 0, 1, 1, 1)), 4))
```

---

randIndex	<i>Rand Index</i>
-----------	-------------------

---

### Description

Compute the Rand index

$$\frac{N_{11} + N_{00}}{N}$$

### Usage

```
randIndex(p, q)
```

```
## S4 method for signature 'Partition,Partition'
randIndex(p, q)
```

```
## S4 method for signature 'PairCoefficients,missing'
randIndex(p, q = NULL)
```

### Arguments

p                      The partition  $P$  or an instance of [PairCoefficients](#)

q                      The partition  $Q$  or NULL

### Methods (by class)

- p = Partition, q = Partition: Compute given two partitions
- p = PairCoefficients, q = missing: Compute given the pair coefficients

### Author(s)

Fabian Ball <fabian.ball@kit.edu>

### References

Rand WM (1971). “Objective Criteria for the Evaluation of Clustering Algorithms.” *Journal of the American Statistical Association*, **66**(336), pp. 846 – 850.

### Examples

```
isTRUE(all.equal(randIndex(new("Partition", c(0, 0, 0, 1, 1)),
                           new("Partition", c(0, 0, 1, 1, 1))), 0.6))
```

---

registerPartitionVectorSignatures

*Make comparison measures usable with any vectors*


---

### Description

The comparison measures are defined to use the class [Partition](#) as parameters. If you do not want to explicitly convert an arbitrary vector of class labels (probably as a result from another package's algorithm) into a [Partition](#) instance, calling this function will create methods for all measures that allow "ANY" input which is implicitly converted to [Partition](#).

### Usage

```
registerPartitionVectorSignatures(e)
```

### Arguments

e                      The environment to register the methods in (mostly environment() is fine)

### Author(s)

Fabian Ball <fabian.ball@kit.edu>

### Examples

```
library(partitionComparison)
randIndex(new("Partition", c(0, 0, 0, 1, 1)), new("Partition", c(0, 0, 1, 1, 1)))
# [1] 0.6
## Not run: randIndex(c(0, 0, 0, 1, 1), c(0, 0, 1, 1, 1))
# Error in (function (classes, fdef, mtable) :
# unable to find an inherited method for function 'randIndex' for signature '"numeric", "numeric"'
registerPartitionVectorSignatures(environment())
randIndex(c(0, 0, 0, 1, 1), c(0, 0, 1, 1, 1))
# [1] 0.6
```

---

rogersTanimoto

*Rogers & Tanimoto Index*


---

### Description

Compute the index of Rogers and Tanimoto

$$\frac{N_{11} + N_{00}}{N_{11} + 2(N_{10} + N_{01}) + N_{00}}$$

## Usage

```
rogersTanimoto(p, q)

## S4 method for signature 'Partition,Partition'
rogersTanimoto(p, q)

## S4 method for signature 'PairCoefficients,missing'
rogersTanimoto(p, q)
```

## Arguments

p                      The partition  $P$  or an instance of [PairCoefficients](#)  
q                        The partition  $Q$  or NULL

## Methods (by class)

- p = Partition, q = Partition: Compute given two partitions
- p = PairCoefficients, q = missing: Compute given the pair coefficients

## Author(s)

Fabian Ball <fabian.ball@kit.edu>

## References

Rogers DJ and Tanimoto TT (1960). “A Computer Program for Classifying Plants.” *Science*, **132**(3434), pp. 1115–1118. ISSN 0036-8075, 1095-9203, doi: [10.1126/science.132.3434.1115](https://doi.org/10.1126/science.132.3434.1115).

## Examples

```
isTRUE(all.equal(rogersTanimoto(new("Partition", c(0, 0, 0, 1, 1)),
                                new("Partition", c(0, 0, 1, 1, 1))), 3/7))
```

---

russelRao

*Russel & Rao Index*

---

## Description

Compute the index of Russel and Rao

$$\frac{N_{11}}{N}$$

## Usage

```
russelRao(p, q)

## S4 method for signature 'Partition,Partition'
russelRao(p, q)

## S4 method for signature 'PairCoefficients,missing'
russelRao(p, q = NULL)
```





**Methods (by class)**

- `p = Partition, q = Partition`: Compute the RV coefficient given two partitions
- `p = PairCoefficients, q = missing`: Compute the RV coefficient given the pair coefficients

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

**References**

Robert P and Escoufier Y (1976). “A Unifying Tool for Linear Multivariate Statistical Methods: The RV- Coefficient.” *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, **25**(3), pp. 257 – 265. ISSN 00359254.

Youness G and Saporta G (2004). “Some Measures of Agreement between Close Partitions.” *Student*, **51**, pp. 1 – 12.

**Examples**

```
isTRUE(all.equal(rvCoefficient(new("Partition", c(0, 0, 0, 1, 1)),
                                new("Partition", c(0, 0, 1, 1, 1))), 9/13))
```

---

sokalSneath1

*Sokal & Sneath Index 1*


---

**Description**

Compute the index 1 of Sokal and Sneath

$$\frac{1}{4} \left( \frac{N_{11}}{N_{21}} + \frac{N_{11}}{N_{12}} + \frac{N_{00}}{N'_{10}} + \frac{N_{00}}{N'_{01}} \right)$$

**Usage**

```
sokalSneath1(p, q)
```

```
## S4 method for signature 'Partition,Partition'
sokalSneath1(p, q)
```

```
## S4 method for signature 'PairCoefficients,missing'
sokalSneath1(p, q = NULL)
```

**Arguments**

`p`                      The partition  $P$  or an instance of [PairCoefficients](#)

`q`                      The partition  $Q$  or NULL

**Methods (by class)**

- `p = Partition, q = Partition`: Compute given two partitions
- `p = PairCoefficients, q = missing`: Compute given the pair coefficients

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

**References**

Sokal RR and Sneath PHA (1963). *Principles of numerical taxonomy*.. Freeman, San Francisco.

**Examples**

```
isTRUE(all.equal(sokalSneath1(new("Partition", c(0, 0, 0, 1, 1)),
                             new("Partition", c(0, 0, 1, 1, 1))), 7/12))
```

---

sokalSneath2	<i>Sokal &amp; Sneath Index 2</i>
--------------	-----------------------------------

---

**Description**

Compute the index 2 of Sokal and Sneath

$$\frac{N_{11}}{N_{11} + 2(N_{10} + N_{01})}$$

**Usage**

```
sokalSneath2(p, q)

## S4 method for signature 'Partition,Partition'
sokalSneath2(p, q)

## S4 method for signature 'PairCoefficients,missing'
sokalSneath2(p, q = NULL)
```

**Arguments**

p                      The partition  $P$  or an instance of [PairCoefficients](#)

q                      The partition  $Q$  or NULL

**Methods (by class)**

- p = Partition, q = Partition: Compute given two partitions
- p = PairCoefficients, q = missing: Compute given the pair coefficients

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

**References**

Sokal RR and Sneath PHA (1963). *Principles of numerical taxonomy*.. Freeman, San Francisco.

**Examples**

```
isTRUE(all.equal(sokalSneath2(new("Partition", c(0, 0, 0, 1, 1)),
                             new("Partition", c(0, 0, 1, 1, 1))), 0.2))
```

---

sokalSneath3	<i>Sokal &amp; Sneath Index 3</i>
--------------	-----------------------------------

---

**Description**

Compute the index 3 of Sokal and Sneath

$$\frac{N_{11}N_{00}}{\sqrt{N_{21}N_{12}N'_{01}N'_{10}}}$$

**Usage**

```
sokalSneath3(p, q)

## S4 method for signature 'Partition,Partition'
sokalSneath3(p, q)

## S4 method for signature 'PairCoefficients,missing'
sokalSneath3(p, q = NULL)
```

**Arguments**

p	The partition $P$ or an instance of <a href="#">PairCoefficients</a>
q	The partition $Q$ or NULL

**Methods (by class)**

- p = Partition, q = Partition: Compute given two partitions
- p = PairCoefficients, q = missing: Compute given the pair coefficients

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

**References**

Sokal RR and Sneath PHA (1963). *Principles of numerical taxonomy*. Freeman, San Francisco.

**Examples**

```
isTRUE(all.equal(sokalSneath3(new("Partition", c(0, 0, 0, 1, 1)),
                             new("Partition", c(0, 0, 1, 1, 1))), 1/3))
```

---

variationOfInformation

*Variation of Information*


---

## Description

Compute the variation of information

$$H(P) + H(Q) - 2MI(P, Q)$$

where  $MI$  is the mutual information,  $H$  the partition entropy

## Usage

```
variationOfInformation(p, q)
```

```
## S4 method for signature 'Partition,Partition'
variationOfInformation(p, q)
```

## Arguments

p	The partition $P$
q	The partition $Q$

## Methods (by class)

- p = Partition, q = Partition: Compute given two partitions

## Author(s)

Fabian Ball <fabian.ball@kit.edu>

## References

Meila M (2003). “Comparing Clusterings by the Variation of Information.” In Schölkopf B and Warmuth MK (eds.), *Learning Theory and Kernel Machines*, volume 2777 series Lecture Notes in Computer Science, pp. 173 – 187. Springer Berlin / Heidelberg. ISBN 978-3-540-40720-1.

Meila M (2007). “Comparing Clusterings – an Information Based Distance.” *Journal of Multivariate Analysis*, **98**(5), pp. 873 – 895. doi: [10.1016/j.jmva.2006.11.013](https://doi.org/10.1016/j.jmva.2006.11.013).

## See Also

[mutualInformation](#), [entropy](#)

## Examples

```
isTRUE(all.equal(variationOfInformation(new("Partition", c(0, 0, 0, 1, 1)),
                                         new("Partition", c(0, 0, 1, 1, 1))),
               0.763817))
```

wallaceI

*Wallace I***Description**

Compute Wallace' index I

$$\frac{N_{11}}{N_{21}}$$

**Usage**

```
wallaceI(p, q)
```

```
## S4 method for signature 'Partition,Partition'
```

```
wallaceI(p, q)
```

```
## S4 method for signature 'PairCoefficients,missing'
```

```
wallaceI(p, q = NULL)
```

**Arguments**

p                      The partition *P* or an instance of [PairCoefficients](#)

q                        The partition *Q* or NULL

**Methods (by class)**

- p = Partition, q = Partition: Compute given two partitions
- p = PairCoefficients, q = missing: Compute given the pair coefficients

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

**References**

Wallace DL (1983). "A Method for Comparing Two Hierarchical Clusterings: Comment." *Journal of the American Statistical Association*, **78**(383), pp. 569 – 576.

**See Also**

[folwkesMallowsIndex](#)

**Examples**

```
isTRUE(all.equal(wallaceI(new("Partition", c(0, 0, 0, 1, 1)),
                           new("Partition", c(0, 0, 1, 1, 1))), 0.5))
```

wallaceII

*Wallace II***Description**

Compute Wallace' index II

$$\frac{N_{11}}{N_{12}}$$

**Usage**

```
wallaceII(p, q)
```

```
## S4 method for signature 'Partition,Partition'
```

```
wallaceII(p, q)
```

```
## S4 method for signature 'PairCoefficients,missing'
```

```
wallaceII(p, q = NULL)
```

**Arguments**

p                      The partition  $P$  or an instance of [PairCoefficients](#)

q                        The partition  $Q$  or NULL

**Methods (by class)**

- p = Partition, q = Partition: Compute given two partitions
- p = PairCoefficients, q = missing: Compute given the pair coefficients

**Author(s)**

Fabian Ball <fabian.ball@kit.edu>

**References**

Wallace DL (1983). "A Method for Comparing Two Hierarchical Clusterings: Comment." *Journal of the American Statistical Association*, **78**(383), pp. 569 – 576.

**See Also**

[folwkesMallowsIndex](#)

**Examples**

```
isTRUE(all.equal(wallaceII(new("Partition", c(0, 0, 0, 1, 1)),
                           new("Partition", c(0, 0, 1, 1, 1))), 0.5))
```

---

[<-,Partition-method    *Subsetting [Partition](#) instances*

---

### Description

This method overrides the standard subsetting to prevent alteration (makes partitions, i.e. class labels, immutable).

### Usage

```
## S4 replacement method for signature 'Partition'  
x[i, j] <- value
```

### Arguments

x	A <a href="#">Partition</a> instance
i	<a href="#">Extract</a>
j	<a href="#">Extract</a>
value	<a href="#">Extract</a>

### Author(s)

Fabian Ball <fabian.ball@kit.edu>

# Index

[\[<-](#), Partition-method, [47](#)  
[adjustedRandIndex](#), [4](#)  
[adjustedRandIndex](#), PairCoefficients, missing-method  
     ([adjustedRandIndex](#)), [4](#)  
[adjustedRandIndex](#), Partition, Partition-method  
     ([adjustedRandIndex](#)), [4](#)  
[baulieu1](#), [5](#)  
[baulieu1](#), PairCoefficients, missing-method  
     ([baulieu1](#)), [5](#)  
[baulieu1](#), Partition, Partition-method  
     ([baulieu1](#)), [5](#)  
[baulieu2](#), [6](#)  
[baulieu2](#), PairCoefficients, missing-method  
     ([baulieu2](#)), [6](#)  
[baulieu2](#), Partition, Partition-method  
     ([baulieu2](#)), [6](#)  
[classificationErrorDistance](#), [7](#)  
[classificationErrorDistance](#), Partition, Partition-method  
     ([classificationErrorDistance](#)),  
     [7](#)  
[compareAll](#), [8](#)  
[compareAll](#), Partition, Partition-method  
     ([compareAll](#)), [8](#)  
[computePairCoefficients](#), [9](#)  
[czezanowski](#), [10](#)  
[czezanowski](#), PairCoefficients, missing-method  
     ([czezanowski](#)), [10](#)  
[czezanowski](#), Partition, Partition-method  
     ([czezanowski](#)), [10](#)  
[data.frame](#), [8](#)  
[dongensMetric](#), [11](#), [36](#)  
[dongensMetric](#), Partition, Partition-method  
     ([dongensMetric](#)), [11](#)  
[entropy](#), [12](#), [33](#), [44](#)  
[entropy](#), ANY, missing-method ([entropy](#)), [12](#)  
[entropy](#), numeric, numeric-method  
     ([entropy](#)), [12](#)  
[entropy](#), Partition, numeric-method  
     ([entropy](#)), [12](#)  
[Extract](#), [47](#)  
[fagerMcGowan](#), [13](#)  
[fagerMcGowan](#), PairCoefficients, missing-method  
     ([fagerMcGowan](#)), [13](#)  
[fagerMcGowan](#), Partition, Partition-method  
     ([fagerMcGowan](#)), [13](#)  
[folwkesMallowsIndex](#), [14](#), [45](#), [46](#)  
[folwkesMallowsIndex](#), PairCoefficients, missing-method  
     ([folwkesMallowsIndex](#)), [14](#)  
[folwkesMallowsIndex](#), Partition, Partition-method  
     ([folwkesMallowsIndex](#)), [14](#)  
[gammaStatistics](#), [15](#)  
[gammaStatistics](#), PairCoefficients, missing-method  
     ([gammaStatistics](#)), [15](#)  
[gammaStatistics](#), Partition, Partition-method  
     ([gammaStatistics](#)), [15](#)  
[goodmanKruskal](#), [16](#)  
[goodmanKruskal](#), PairCoefficients, missing-method  
     ([goodmanKruskal](#)), [16](#)  
[goodmanKruskal](#), Partition, Partition-method  
     ([goodmanKruskal](#)), [16](#)  
[gowerLegendre](#), [17](#)  
[gowerLegendre](#), PairCoefficients, missing-method  
     ([gowerLegendre](#)), [17](#)  
[gowerLegendre](#), Partition, Partition-method  
     ([gowerLegendre](#)), [17](#)  
[hamann](#), [18](#)  
[hamann](#), PairCoefficients, missing-method  
     ([hamann](#)), [18](#)  
[hamann](#), Partition, Partition-method  
     ([hamann](#)), [18](#)  
[jaccardCoefficient](#), [19](#)  
[jaccardCoefficient](#), PairCoefficients, missing-method  
     ([jaccardCoefficient](#)), [19](#)  
[jaccardCoefficient](#), Partition, Partition-method  
     ([jaccardCoefficient](#)), [19](#)  
[kulczynski](#), [20](#)  
[kulczynski](#), PairCoefficients, missing-method  
     ([kulczynski](#)), [20](#)  
[kulczynski](#), Partition, Partition-method  
     ([kulczynski](#)), [20](#)



- larsenAone, [21](#)
- larsenAone, Partition, Partition-method  
(larsenAone), [21](#)
- lermanIndex, [22](#), [32](#)
- lermanIndex, Partition, Partition, missing-method  
(lermanIndex), [22](#)
- lermanIndex, Partition, Partition, PairCoefficients-method  
(lermanIndex), [22](#)
- lp.assign, [7](#)
- mcconnaughey, [23](#)
- mcconnaughey, PairCoefficients, missing-method  
(mcconnaughey), [23](#)
- mcconnaughey, Partition, Partition-method  
(mcconnaughey), [23](#)
- minkowskiMeasure, [24](#)
- minkowskiMeasure, PairCoefficients, missing-method  
(minkowskiMeasure), [24](#)
- minkowskiMeasure, Partition, Partition-method  
(minkowskiMeasure), [24](#)
- mirkinMetric, [25](#)
- mirkinMetric, PairCoefficients, missing-method  
(mirkinMetric), [25](#)
- mirkinMetric, Partition, Partition-method  
(mirkinMetric), [25](#)
- mutualInformation, [26](#), [33](#), [44](#)
- mutualInformation, Partition, Partition-method  
(mutualInformation), [26](#)
- N, [27](#)
- N, PairCoefficients-method (N), [27](#)
- N00, [27](#), [34](#)
- N00, PairCoefficients-method (N00), [27](#)
- N01, [28](#), [34](#)
- N01, PairCoefficients-method (N01), [28](#)
- N01p, [28](#)
- N01p, PairCoefficients-method (N01p), [28](#)
- N10, [29](#), [34](#)
- N10, PairCoefficients-method (N10), [29](#)
- N10p, [29](#)
- N10p, PairCoefficients-method (N10p), [29](#)
- N11, [30](#), [34](#)
- N11, PairCoefficients-method (N11), [30](#)
- N12, [30](#)
- N12, PairCoefficients-method (N12), [30](#)
- N21, [31](#)
- N21, PairCoefficients-method (N21), [31](#)
- normalizedLermanIndex, [22](#), [31](#)
- normalizedLermanIndex, Partition, Partition, missing-method  
(normalizedLermanIndex), [31](#)
- normalizedLermanIndex, Partition, Partition, PairCoefficients-method  
(normalizedLermanIndex), [31](#)
- normalizedMutualInformation, [26](#), [32](#)
- normalizedMutualInformation, Partition, Partition, character  
(normalizedMutualInformation), [32](#)
- normalizedMutualInformation, Partition, Partition, missing  
(normalizedMutualInformation), [32](#)
- PairCoefficients, [4-6](#), [10](#), [13-20](#), [22-25](#),  
[27-31](#), [35](#), [37](#), [39-43](#), [45](#), [46](#)
- PairCoefficients  
(PairCoefficients-class), [33](#)
- PairCoefficients-class, [33](#)
- Partition, [3](#), [8](#), [38](#), [47](#)
- Partition (Partition-class), [34](#)
- Partition-class, [34](#)
- partitionComparison  
(partitionComparison-package), [3](#)
- partitionComparison-package, [3](#)
- pearson, [34](#)
- pearson, PairCoefficients, missing-method  
(pearson), [34](#)
- pearson, Partition, Partition-method  
(pearson), [34](#)
- peirce, [35](#)
- peirce, PairCoefficients, missing-method  
(peirce), [35](#)
- peirce, Partition, Partition-method  
(peirce), [35](#)
- projectionNumber, [11](#), [36](#)
- randIndex, [37](#)
- randIndex, PairCoefficients, missing-method  
(randIndex), [37](#)
- randIndex, Partition, Partition-method  
(randIndex), [37](#)
- registerPartitionVectorSignatures, [3](#),  
[38](#)
- rogersTanimoto, [38](#)
- rogersTanimoto, PairCoefficients, missing-method  
(rogersTanimoto), [38](#)
- rogersTanimoto, Partition, Partition-method  
(rogersTanimoto), [38](#)
- russelRao, [39](#)
- russelRao, PairCoefficients, missing-method  
(russelRao), [39](#)
- russelRao, Partition, Partition-method  
(russelRao), [39](#)
- rvCoefficient, [40](#)
- rvCoefficient, PairCoefficients, missing-method  
(rvCoefficient), [40](#)
- rvCoefficient, Partition, Partition-method  
(rvCoefficient), [40](#)

sokalSneath1, [41](#)  
sokalSneath1, PairCoefficients, missing-method  
    (sokalSneath1), [41](#)  
sokalSneath1, Partition, Partition-method  
    (sokalSneath1), [41](#)  
sokalSneath2, [42](#)  
sokalSneath2, PairCoefficients, missing-method  
    (sokalSneath2), [42](#)  
sokalSneath2, Partition, Partition-method  
    (sokalSneath2), [42](#)  
sokalSneath3, [43](#)  
sokalSneath3, PairCoefficients, missing-method  
    (sokalSneath3), [43](#)  
sokalSneath3, Partition, Partition-method  
    (sokalSneath3), [43](#)  
  
variationOfInformation, [44](#)  
variationOfInformation, Partition, Partition-method  
    (variationOfInformation), [44](#)  
  
wallaceI, [14](#), [45](#)  
wallaceI, PairCoefficients, missing-method  
    (wallaceI), [45](#)  
wallaceI, Partition, Partition-method  
    (wallaceI), [45](#)  
wallaceII, [14](#), [46](#)  
wallaceII, PairCoefficients, missing-method  
    (wallaceII), [46](#)  
wallaceII, Partition, Partition-method  
    (wallaceII), [46](#)