

Package: DiffXTables (via r-universe)

August 28, 2024

Title Pattern Analysis Across Contingency Tables

Version 0.1.3

Date 2021-05-15

Author Ruby Sharma [aut] (<<https://orcid.org/0000-0001-7774-4065>>),
Joe Song [aut, cre] (<<https://orcid.org/0000-0002-6883-6547>>)

Maintainer Joe Song <joemsong@cs.nmsu.edu>

Description Statistical hypothesis testing of pattern heterogeneity via differences in underlying distributions across multiple contingency tables. Five tests are included: the comparative chi-squared test (Song et al. 2014) <[doi:10.1093/nar/gku086](https://doi.org/10.1093/nar/gku086)> (Zhang et al. 2015) <[doi:10.1093/nar/gkv358](https://doi.org/10.1093/nar/gkv358)>, the Sharma-Song test (Sharma et al. 2021) <[doi:10.1093/bioinformatics/btab240](https://doi.org/10.1093/bioinformatics/btab240)>, the heterogeneity test, the marginal-change test (Sharma et al. 2020) <[doi:10.1145/3388440.3412485](https://doi.org/10.1145/3388440.3412485)>, and the strength test (Sharma et al. 2020) <[doi:10.1145/3388440.3412485](https://doi.org/10.1145/3388440.3412485)>. Under the null hypothesis that row and column variables are statistically independent and joint distributions are equal, their test statistics all follow an asymptotically chi-squared distribution. A comprehensive type analysis categorizes the relation among the contingency tables into type null, 0, 1, and 2 (Sharma et al. 2020) <[doi:10.1145/3388440.3412485](https://doi.org/10.1145/3388440.3412485)>. They can identify heterogeneous patterns that differ in either the first order (marginal) or the second order (differential departure from independence). Second-order differences reveal more fundamental changes than first-order differences across heterogeneous patterns.

Depends R (>= 3.0), pander

Imports Rdpack (>= 0.6-1)

RdMacros Rdpack

Suggests Ckmeans.1d.dp, FunChisq, knitr, rmarkdown, testthat

License LGPL (>= 3)

Encoding UTF-8

NeedsCompilation no

VignetteBuilder knitr**RoxygenNote** 7.1.1**Date/Publication** 2021-05-16 04:30:02 UTC**Repository** https://joemsong.r-universe.dev**RemoteUrl** https://github.com/cran/DiffXTables**RemoteRef** HEAD**RemoteSha** 4bba27f406f49d22b69a7674c1015bdc48c16dcb

Contents

cp.chisq.test	2
heterogeneity.test	4
marginal.change.test	6
sharma.song.test	7
simulate_diff_tables	9
strength.test	11
type.analysis	13

Index	16
--------------	-----------

cp.chisq.test	<i>Comparative Chi-Squared Test for Difference Across Contingency Tables</i>
---------------	--

Description

Across given contingency tables, the test admits any type of differences in either the joint or marginal distributions underlying the tables.

Usage

```
cp.chisq.test(
  tables, method=c("chisq", "nchisq", "default", "normalized"),
  log.p = FALSE
)
```

Arguments

tables	a list of at least two nonnegative matrices or data frames representing contingency tables of the same dimensions.
method	a character string to specify the method to compute the chi-squared statistic and its p-value. The default is "chisq". See Details. Note: "default" and "normalized" are deprecated.
log.p	logical; if TRUE, the p-value is given as $\log(p)$. Taking the log improves the accuracy when p-value is close to zero. The default is FALSE.

Details

The comparative chi-squared test determines whether the patterns underlying multiple contingency tables are heterogeneous. Its null test statistic is proved to asymptotically follow the chi-squared distribution (Song et al. 2014; Zhang et al. 2015). This test is different from the heterogeneity test (Zar 2009).

Two methods are provided to compute the chi-squared statistic and its p-value. When method = "chisq" (or "default"), the p-value is computed using the chi-squared distribution; when method = "nchisq" (or "normalized") a normalized statistic is obtained by shifting and scaling the original chi-squared test statistic and a p-value is computed using the standard normal distribution (George et al. 2005). The normalized test is more conservative on the degrees of freedom.

Either test statistic is minimized to zero if and only if the input tables are linearly scaled versions of each other.

The test is recommended to determine whether multiple contingency tables have the same distributions, regardless of independence of row and column variables in each table.

Value

A list with class "hstest" containing the following components:

statistic	chi-squared test statistic if method = "chisq" (equivalent to "default"), or normalized test statistic if method = "nchisq" (equivalent to "normalized").
parameter	degrees of freedom of the chi-squared statistic.
p.value	p-value of the comparative chi-squared test. By default, it is computed by the chi-squared distribution (method = "chisq" or "default"). If method = "nchisq" (or "normalized"), it is the p-value of the normalized chi-squared statistic using the standard normal distribution.

Author(s)

Joe Song

References

George EP, Hunter JS, Hunter WG, Bins R, Kirilin IV K, Carroll D (2005). *Statistics for Experimenters: Design, Innovation, and Discovery*. Wiley, New York.

Song M, Zhang Y, Katzaroff AJ, Edgar BA, Buttitta L (2014). "Hunting complex differential gene interaction patterns across molecular contexts." *Nucleic Acids Research*, **42**(7), e57. doi: [10.1093/nar/gku086](https://doi.org/10.1093/nar/gku086), <https://doi.org/10.1093/nar/gku086>.

Zar JH (2009). *Biostatistical Analysis*, 5th edition. Prentice Hall, New Jersey.

Zhang Y, Liu ZL, Song M (2015). "ChiNet uncovers rewired transcription subnetworks in tolerant yeast for advanced biofuels conversion." *Nucleic Acids Research*, **43**(9), 4393–4407. doi: [10.1093/nar/gkv358](https://doi.org/10.1093/nar/gkv358), <https://doi.org/10.1093/nar/gkv358>.

See Also

The Sharma-Song test [sharma.song.test](#).

The heterogeneity test [heterogeneity.test](#).

Examples

```
# Two second-order differential tables:
tables <- list(
  matrix(c(4,0,0,
          0,4,0,
          0,0,4), nrow=3),
  matrix(c(0,4,4,
          4,0,4,
          4,4,0), nrow=3)
)
cp.chisq.test(tables)

# Three tables differ in the first-order but not second-order:
tables <- list(
  matrix(c(2, 4, 6, 8,
          3, 6, 9, 12,
          4, 8, 12, 16), nrow=4),
  matrix(c( 2, 1, 3, 7,
          2, 1, 3, 7,
          10, 5, 15, 35), nrow=4),
  matrix(c(40, 16, 72, 16,
          45, 18, 81, 18,
          25, 10, 45, 10), nrow=4)
)
cp.chisq.test(tables)
```

heterogeneity.test *Heterogeneity Test for Difference Across Contingency Tables*

Description

Across given contingency tables, the test admits any type of differences in either the joint or marginal distributions of the tables.

Usage

```
heterogeneity.test(tables)
```

Arguments

tables a list of at least two non-negative matrices or data frames representing contingency tables of the same dimensions.

Details

The heterogeneity test determines whether the patterns underlying multiple contingency tables are heterogeneous or differential. The chi-squared distribution is used for the null distribution of its test statistic (Zar, 2010).

Value

A list with class "htest" containing the following components:

statistic	heterogeneity test statistic.
parameter	degrees of freedom of used for the null distribution of the heterogeneity test statistic.
p.value	p-value of the heterogeneity test, computed using the chi-squared distribution.

References

Zar, J. H. (2010) *Biostatistical Analysis*, 5th Ed., New Jersey: Prentice Hall.

See Also

The comparative chi-squared test [cp.chisq.test](#).

The Sharma-Song test [sharma.song.test](#).

Examples

```
# Two second-order differential tables:
tables <- list(
  matrix(c(4,0,0,
           0,4,0,
           0,0,4), nrow=3),
  matrix(c(0,4,4,
           4,0,4,
           4,4,0), nrow=3)
)
heterogeneity.test(tables)

# Three tables differ in the first-order but not second-order:
tables <- list(
  matrix(c(2, 4, 6, 8,
           3, 6, 9, 12,
           4, 8, 12, 16), nrow=4),
  matrix(c( 2, 1, 3, 7,
           2, 1, 3, 7,
           10, 5, 15, 35), nrow=4),
  matrix(c(40, 16, 72, 16,
           45, 18, 81, 18,
           25, 10, 45, 10), nrow=4)
)
heterogeneity.test(tables)
```

marginal.change.test *Test for Marginal Change Across Contingency Tables*

Description

The test detects change in either row or column marginal distributions across given contingency tables.

Usage

```
marginal.change.test(tables)
```

Arguments

tables a list of at least two non-negative matrices or data frames representing contingency tables of the same dimensions.

Details

The marginal change test determines whether the patterns underlying multiple input contingency tables have changed row or column marginal distributions. Its test statistic is proved to asymptotically follow the chi-squared distribution under the null hypothesis of same row and marginal distributions across tables (Sharma et al. 2020).

The test statistic is minimized to zero if and only if observed row marginal distributions are the same across tables and so do the column marginal distributions.

Value

A list with class "hctest" containing the following components:

statistic the chi-squared test statistic.
parameter the degrees of freedom of the null chi-squared distribution.
p.value the p-value for the test, computed using the chi-squared distribution.

Author(s)

Ruby Sharma and Joe Song

References

Sharma R, Luo X, Kumar S, Song M (2020). "Three Co-Expression Pattern Types across Microbial Transcriptional Networks of Plankton in Two Oceanic Waters." In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, BCB '20. ISBN 9781450379649, doi: [10.1145/3388440.3412485](https://doi.org/10.1145/3388440.3412485), <https://doi.org/10.1145/3388440.3412485>.

See Also

[sharma.song.test](#), [strength.test](#), and [type.analysis](#).

Examples

```
# Two first-order differential tables:
tables <- list(
  matrix(c(30,0,0,
           0,10,0,
           0,0,20), nrow=3),
  matrix(c(10,0,0,
           0,20,0,
           0,0,30), nrow=3)
)
marginal.change.test(tables)

# Tables differ in the second-order but not first-order:
tables <- list(
  matrix(c(4,0,0,
           0,4,0,
           0,0,4), nrow=3),
  matrix(c(0,0,4,
           0,4,0,
           4,0,0), nrow=3)
)
marginal.change.test(tables)
```

sharma.song.test

Sharma-Song Test for Second-Order Difference Across Contingency Tables

Description

The test detects differential departure from independence via second-order difference in joint distributions underlying two or more contingency tables.

Usage

```
sharma.song.test(
  tables, null.table.marginal = c("observed", "uniform"),
  compensated = FALSE
)
```

Arguments

`tables` a list of at least two non-negative matrices or data frames representing contingency tables of the same dimensions.

`null.table.marginal` a character string to specify marginal distributions of null tables. The options are "observed"(default) and "uniform".

`compensated` a logical value to compensate for the Cochran's condition. It is only used if `null.table.marginal="observed"`. Default is FALSE.

Details

The Sharma-Song test determines whether the patterns underlying multiple input contingency tables are second-order differential. The test statistic measures differential departure from independence. Its null test statistic is proved to asymptotically follow the chi-squared distribution. For full detail of the test, see (Sharma et al. 2021).

If `null.table.marginal` is set to "observed", the null hypothesis uses the observed marginals. The `compensated` parameter, if set to TRUE, adds a small constant to each entry of the tables to address the Cochran's condition that the expected count in any table entry is 5 or less.

If the `null.table.marginal` is set to "uniform", the null tables are set to have uniform marginals. No longer suffering from the Cochran's condition, it detects second-order differential patterns with additional robustness.

Value

A list with class "htest" containing the following components:

`statistic` the Sharma-Song chi-squared test statistic.

`parameter` degrees of freedom of the chi-squared test statistic.

`p.value` p-value of the Sharma-Song test, computed using the chi-squared distribution.

Author(s)

Ruby Sharma and Joe Song

References

Sharma R, Kumar S, Song M (2021). "Fundamental gene network rewiring at the second order within and across mammalian systems." *Bioinformatics*. doi: [10.1093/bioinformatics/btab240](https://doi.org/10.1093/bioinformatics/btab240), <https://doi.org/10.1093/bioinformatics/btab240>.

See Also

[cp.chisq.test](#), [heterogeneity.test](#), [strength.test](#), [marginal.change.test](#), and [type.analysis](#).

Examples

```
# Two second-order differential tables:
tables <- list(
  matrix(c(4,0,0,
           0,4,0,
           0,0,4), nrow=3),
  matrix(c(0,4,4,
```



```

      4,0,4,
      4,4,0), nrow=3)
)
sharma.song.test(tables)

# Three tables differ in the first-order but not second-order:
tables <- list(
  matrix(c(2, 4, 6, 8,
           3, 6, 9, 12,
           4, 8, 12, 16), nrow=4),
  matrix(c( 2, 1, 3, 7,
           2, 1, 3, 7,
           10, 5, 15, 35), nrow=4),
  matrix(c(40, 16, 72, 16,
           45, 18, 81, 18,
           25, 10, 45, 10), nrow=4)
)
sharma.song.test(tables, null.table.marginal = "uniform")

```

simulate_diff_tables *Simulating Contingency Tables that Differ in Distribution*

Description

Generate contingency tables that are first-order, second-order or full-order differential in the joint distribution of row and column variables.

Usage

```

simulate_diff_tables(
  K = 2, nrow = 3, ncol = 3, n = 100, B = 100,
  type = c("second-order", "first-order", "full-order")
)

```

Arguments

K	the number of tables that are differential. It must be an integer greater than one.
nrow	the number of rows for all tables to be generated. It must be an integer greater than one.
ncol	the number of columns for all tables to be generated. It must be an integer greater than one.
n	the sample size for each table to be generated. It must be a positive integer.
B	the number of iterations indicating the level of differentiability. It must be a positive integer. The greater the value, the stronger the differentiability across tables.
type	the type of differential tables to be generated. Options are "first-order", "second-order" (default), and "full-order". See Details.

Details

The function randomly generates contingency tables differential in the joint distribution of the row and column variables. Specifically, three types of differential contingency tables can be simulated:

First-order differential contingency tables only differ in row or column marginal distribution. Such tables are differential in joint distribution, but different from second-order differential tables.

Second-order differential contingency tables differ in joint distribution. The difference is not attributed to row or column marginal distributions.

Full-order differential contingency tables are tables that are both first-order and second-order differential.

The simulation starts with randomly generated probability tables where row and column variables are independent. The probability tables are modified to K tables such that they represent specific distributions that strictly satisfy the type requirement. Finally, contingency tables are generated using multinomial distribution using these probability tables parameters and the required sample size.

Value

A list containing the following components:

contingency.tables

a list of K contingency tables that are differential in joint distribution according to the type argument. They contain non-negative integers following the multinomial distribution with probability parameters from probability.tables.

probability.tables

a list of K tables representing randomly generated differential joint probabilities that reflect the specified type.

method

a string that specifies the type of the differential tables.

Author(s)

Ruby Sharma and Joe Song

See Also

Differential tables are simulated to evaluate the following tests for comparing contingency tables:

The Sharma-Song test [sharma.song.test](#)

The comparative chi-squared test [cp.chisq.test](#)

The heterogeneity test [heterogeneity.test](#)

The [simulate_tables](#) function in package **FunChisq** can generate a variety of tables.

Examples

```
# Three first-order differential tables:
res <- simulate_diff_tables(K = 3, nrow = 4, ncol = 3, n = 150, B = 200, type = "first-order")
print(res)

# Two second-order differential tables:
```

```

res <- simulate_diff_tables(K = 2, nrow = 2, ncol = 5, n = 100, B = 100, type = "second-order")
print(res)

# Four full-order differential tables:
res <- simulate_diff_tables(K = 4, nrow = 3, ncol = 4, n = 250, B = 200, type = "full-order")
print(res)

```

strength.test

Strength Test for Association in Multiple Contingency Tables

Description

The test determines the total strength of association in multiple contingency tables.

Usage

```
strength.test(tables)
```

Arguments

`tables` a list of at least two non-negative matrices or data frames representing contingency tables.

Details

The strength test determines total amount of association in multiple input contingency tables. Its test statistic asymptotically follows the chi-squared distribution under the null hypothesis of each table having independent row and column variables (Sharma et al. 2020).

The test statistic is minimized to zero if and only if row and column variables are empirically independent of each other in every table.

This test is considered a zeroth-order test in the function `type.analysis` that characterizes the difference across multiple contingency tables.

Value

A list with class "htest" containing the following components:

<code>statistic</code>	the strength test statistic.
<code>parameter</code>	the degrees of freedom of null chi-squared distribution.
<code>p.value</code>	the p-value for the test, computed using the null chi-squared distribution.

Author(s)

Ruby Sharma and Joe Song

References

Sharma R, Luo X, Kumar S, Song M (2020). “Three Co-Expression Pattern Types across Microbial Transcriptional Networks of Plankton in Two Oceanic Waters.” In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, BCB '20. ISBN 9781450379649, doi: [10.1145/3388440.3412485](https://doi.org/10.1145/3388440.3412485), <https://doi.org/10.1145/3388440.3412485>.

See Also

A second-order different test [sharma.song.test](#). A first-order difference test [marginal.change.test](#). The comparative type of differences across contingency tables [type.analysis](#).

Examples

```
# Both tables have strong association:
tables <- list(
  matrix(c(30,0,0,
           0,10,0,
           0,0,20), nrow=3),
  matrix(c(10,0,0,
           0,20,0,
           0,0,30), nrow=3)
)
strength.test(tables)

# One table has strong association:
tables <- list(
  matrix(c(4,0,0,
           0,4,0,
           0,0,4), nrow=3),
  matrix(c(4,0,4,
           8,4,8,
           4,0,4), nrow=3)
)
strength.test(tables)

# Both tables has no association:
tables <- list(
  matrix(c(4,0,4,
           8,4,8,
           4,0,4), nrow=3),
  matrix(c(4,0,4,
           8,4,8,
           4,0,4), nrow=3)
)
strength.test(tables)
```

type.analysis	<i>Comprehensive Type Analysis for Difference Across Contingency Tables</i>
---------------	---

Description

Four types (0, 1, 2, null) are assigned to a collection of contingency tables to categorize their differences.

Usage

```
type.analysis(tables, alpha = 0.05)
```

Arguments

tables	a list of at least two non-negative matrices or data frames representing contingency tables of the same dimensions.
alpha	a numerical value representing the significance level for all involved hypothesis tests. Default is 0.05.

Details

The function determines whether differences across patterns underlying multiple input contingency tables are type 0, 1, 2, or null. The function calls [strength.test](#), [marginal.change.test](#), and [sharma.song.test](#) to obtain three p-values and use them to decide on the type of difference among input contingency tables (Sharma et al. 2020).

Type null: association absent from the input tables. These tables can still differ in joint distribution, but a strong relationship is lacking between row and column variables. No mechanisms are implied.

Type 0: association present and patterns are conserved. These tables show strong row and column association but have no difference in distribution. Conserved mechanisms with conserved trajectories are implied.

Type 1: association present; the difference is up to the first order. These tables show strong row and column association, differ in marginal distribution, and do not differ deviation from joint distribution to product of marginals. Conserved mechanisms with differential trajectories are implied. Differences in trajectory can be due to changed stimuli.

Type 2: association present; the difference is up to the second order. These tables show strong row and column association and differ in deviation from joint distribution to the product of marginals. Differential mechanisms are implied.

Value

A list with class "DiffXTableTypeAnalysis" containing the following components:

Zeroth.order	a list with class "htest" containing the test statistic, degrees of freedom and p-value of the strength.test .
--------------	--

First.order	a list with class "htest" containing the test statistic, degrees of freedom and p-value of the marginal.change.test.
Second.order	a list with class "htest" containing the test statistic, degrees of freedom and p-value of the sharma.song.test.
Type	the type of differences across the input contingency tables. Possible values are 0, 1, 2, and NULL.

Author(s)

Ruby Sharma and Joe Song

References

Sharma R, Luo X, Kumar S, Song M (2020). "Three Co-Expression Pattern Types across Microbial Transcriptional Networks of Plankton in Two Oceanic Waters." In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, BCB '20. ISBN 9781450379649, doi: [10.1145/3388440.3412485](https://doi.org/10.1145/3388440.3412485), <https://doi.org/10.1145/3388440.3412485>.

See Also

[strength.test](#), [marginal.change.test](#), and [sharma.song.test](#).

A related function is the comparative chi-squared test [cp.chisq.test](#).

Examples

```
# Type-null tables:
tables <- list(
  matrix(c(7, 4, 1,
           3, 6, 9,
           2, 4, 4), nrow=3),
  matrix(c(2, 1, 2,
           2, 0, 8,
           6, 2, 7), nrow=3)
)
type.analysis(tables, alpha = 0.05)

# Type-0 tables:
tables <- list(
  matrix(c(30,0,0,
           0,10,0,
           0,0,20), nrow=3),
  matrix(c(30,0,0,
           0,10,0,
           0,0,20), nrow=3)
)
type.analysis(tables, alpha = 0.05)

# Type-1 differential tables:
tables <- list(
```

```
matrix(c(30,0,0,
        0,10,0,
        0,0,20), nrow=3),
matrix(c(10,0,0,
        0,20,0,
        0,0,30), nrow=3)
)
type.analysis(tables, alpha = 0.05)

# Type-2 differential tables:
tables <- list(
  matrix(c(4,0,0,
          0,4,0,
          0,0,4), nrow=3),
  matrix(c(0,4,4,
          4,0,4,
          4,4,0), nrow=3)
)
type.analysis(tables, alpha = 0.05)
```

Index

- * **datagen**
 - simulate_diff_tables, 9
 - * **distribution**
 - marginal.change.test, 6
 - sharma.song.test, 7
 - strength.test, 11
 - type.analysis, 13
 - * **htest**
 - cp.chisq.test, 2
 - heterogeneity.test, 4
 - marginal.change.test, 6
 - sharma.song.test, 7
 - strength.test, 11
 - type.analysis, 13
 - * **nonparametric**
 - cp.chisq.test, 2
 - heterogeneity.test, 4
 - marginal.change.test, 6
 - sharma.song.test, 7
 - strength.test, 11
 - type.analysis, 13
- cp.chisq.test, 2, 5, 8, 10, 14
- heterogeneity.test, 4, 4, 8, 10
- marginal.change.test, 6, 8, 12–14
- sharma.song.test, 4, 5, 7, 7, 10, 12–14
- simulate_diff_tables, 9
- simulate_tables, 10
- strength.test, 7, 8, 11, 13, 14
- type.analysis, 7, 8, 12, 13