Package ‘DiscreteDatasets’

May 28, 2024

Type Package

Title Example Data Sets for Use with Discrete Statistical Tests

Version 0.1.1

Date 2024-05-28

Maintainer Florian Junge <diso.fbmn@h-da.de>

Description Provides several data sets for use with discrete statistical tests and discrete multiple testing procedures. Some of them are also available as a four-column version, so that each row represents a 2x2 table.

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 4.0)

Imports checkmate

URL https://github.com/DISOhda/DiscreteDatasets

BugReports https://github.com/DISOhda/DiscreteDatasets/issues

RoxygenNote 7.3.1

NeedsCompilation no

Author Christina Kihn [aut],
Sebastian Döhler [aut],
Florian Junge [cre, aut]

Repository CRAN

Date/Publication 2024-05-28 12:00:02 UTC

R topics documented:

`DiscreteDatasets-package` *DiscreteDatasets*

Description

This package contains example datasets for use with discrete statistical tests and discrete multiple testing procedures. Some of them are also available as a four-column version, so that each row represents a 2x2 table.

Author(s)

Maintainer: Florian Junge `<diso.fbmn@h-da.de>`

Authors:

- Christina Kihn
- Sebastian Döhler

See Also

Useful links:

- `https://github.com/DISOhda/DiscreteDatasets`
- Report bugs at `https://github.com/DISOhda/DiscreteDatasets/issues`

`airway` *Airway smooth muscle cells*

Description

Read counts per gene for airway smooth muscle cell lines RNA-Seq experiment

Usage

```
data("airway")

data("airway_treat")

data("airway_four_columns")
```

Format

airway is a data.frame with 63,677 rows and 8 columns. Each row corresponds to a specific gene and each column to a labeled sample.

airway_treat is a data.frame with 63,677 rows representing genes with the following two columns:

Treatment Number of reads for the specific gene in all treated samples.

NoTreatment Number of reads for the specific gene in all untreated samples.

Thus, each line describes a 2x2 table, e.g.:

ENSG00000000003	This gene	All other genes
Treatment	$X_{i,1}$	89,561,179 - $X_{i,1}$
No Treatment	$X_{i,2}$	85,955,244 - $X_{i,2}$

airway_four_columns is a data.frame with 63,677 rows representing genes with the following four columns:

Treatment.ThisGene Number of reads for the specific gene in all treated samples.

NoTreatment.ThisGene Number of reads for the specific gene in all untreated samples.

Treatment.AllOtherGenes Number of reads for all other genes in all treated samples.

NoTreatment.AllOtherGenes Number of reads for tall other genes in all untreated samples.

Thus, each line describes a 2x2 table, e.g.:

ENSG00000000003	This gene	All other genes
Treatment	$X_{i,1}$	$X_{i,3}$
No Treatment	$X_{i,2}$	$X_{i,4}$

Details

The cell lines of the even-numbered samples were treated with dexamethasone, whereas the cell lines of the odd-numbered samples were not. There were 89,561,179 reads for all treated samples and 85,955,244 for the untreated ones.

Note

The original airway dataset has been taken from the airway BioConductor package. Since the original data would require other BioConductor packages to access it, it has been reformatted to a standard data frame (with assay(airway)) which only contains the raw numeric data.

Source

FASTQ files from SRA, phenotypic data from GEO

References

Himes, B. E., Jiang, X., Wagner, P., Hu, R., Wang, Q., Klanderman, B., Whitaker, R. M., Duan, Q., Lasky-Su, J., Nikolos, C., Jester, W., Johnson, M., Panettieri, R. Jr., Tantisira, K. G., Weiss,

S. T., Lu, Q. (2014). RNA-Seq Transcriptome Profiling Identifies CRISPLD2 as a Glucocorticoid Responsive Gene that Modulates Cytokine Function in Airway Smooth Muscle Cells. *PLoS One* **9**(6). doi:10.1371/journal.pone.0099625

`amnesia` *Amnesia and other drug reactions in the MHRA pharmacovigilance spontaneous reporting system*

Description

For each of 2,446 drugs in the MHRA database (column 1), the number of cases with amnesia as an adverse event (column 2), and the number of cases with other adverse event for this drug (column 3). In total, 684,652 adverse drug reactions were reported, among them 2,044 cases of amnesia.

Usage

```
data("amnesia")

data("amnesia_four_columns")
```

Format

`amnesia` is a `data.frame` with 2,446 rows representing drugs with the following two columns:

AmnesiaCases Number of the amnesia cases reported for the drug.

OtherAdverseCases Number of other adverse drug reactions reported for the drug.

Thus, each line describes a 2x2 table, e.g.:

1-ANDROSTENEDIOL	This drug	All other drugs
Amnesia cases	$X_{i,1}$	2,044 - $X_{i,1}$
Other adverse cases	$X_{i,2}$	682,648 - $X_{i,2}$

`amnesia_four_columns` is a `data.frame` with 2,446 rows representing drugs with the following four columns:

AmnesiaCases.ThisDrug Number of the amnesia cases reported for the drug.

AmnesiaCases.AllOtherDrugs Number of the amnesia cases reported for all other drugs.

OtherAdverseCases.ThisDrug Number of other adverse drug reactions reported for the drug.

OtherAdverseCases.AllOtherDrugs Number of other adverse drug reactions reported for all other drugs.

Thus, each line describes a 2x2 table:

1-ANDROSTENEDIOL	This drug	All other drugs
Amnesia cases	$X_{i,1}$	$X_{i,3}$
Other adverse cases	$X_{i,2}$	$X_{i,4}$

Details

The data was collected from the Drug Analysis Prints published by the Medicines and Healthcare products Regulatory Agency (MHRA), by Heller & Gur. See references for more details.

Note

The original `amnesia` dataset has been taken from the `discreteMTP` package, which is no longer available on CRAN. It has been reformatted such that the names in first column are now row descriptions; this way, the actual contents of the table are purely numeric.

Source

Drug Analysis Prints on MHRA site

References

R. Heller and H. Gur (2011). False discovery rate controlling procedures for discrete tests. arXiv pre-print arXiv:1112.4627v2 link.

`disorderdetection` *Disorder Detection data*

Description

For earlier recognition of diseases, multiple variations of the human base sequence get studied. The so-called coverage of each base is calculated to detect duplicates, deletions and insertions in the base sequence. To find these variations a hypothesis-test gets performed for each base in the tested area. The null-hypothesis being that the coverage of the base is as expected under the null-hypothesis (expected coverage Cb can be calculated using a given formula, following a poisson distribution). If the observed coverage is exceptionally high or low the null-hypothesis gets rejected. For each type of variation there is a different formula to calculate the expected coverages. The expected coverages in this data set were calculated using the formula for a local test without GC-correction.

Usage

```
data("disorderdetection")
```

Format

A data frame with 315 rows representing a base sequence with the following 2 columns:

`observed frequencies`	Observed coverage of each base
`expected frequencies`	Expected coverage of each base

Details

The data was collected from the "Goodness-of-fit tests for disorder detection in NGS experiments" published by the Biometrical Journal , by Jiménez-Otero, de Uña-Álvarez and Pardo-Fernández. See references for more details.

References

Jiménez-Otero N, de Uña-Álvarez J, Pardo-Fernández JC (2019). Goodness-of-fit tests for disorder detection in NGS experiments. *Biometrical Journal*, **61**(2), pp. 424-441. doi:10.1002/bimj.201700284.

---

hiv *HIV data*

Description

This data set has been analyzed and provided by the listed reference. Examined were two groups with different types of HIV (Type B and Type C), each consisting of 73 participants. Within both groups the number of amino-acid mutations at each position was determined.

Usage

```
data("hiv")

data("hiv_four_columns")
```

Format

`hiv` is a `data.frame` with 118 rows and the following two columns:

TypeC Number of test subjects with HIV type C and mutated i-th amino acid.

TypeB Number of test subjects with HIV type B and mutated i-th amino acid.

Thus, each row describes a 2x2 table:

Subject 1	Mutation	No mutation
Type C	$X_{i,1}$	73 - $X_{i,1}$
Type B	$X_{i,2}$	73 - $X_{i,2}$

`hiv_four_columns` is a `data.frame` with 118 rows and the following four columns:

TypeC.Mutation Number of test subjects with HIV type C and mutated i-th amino acid.

TypeB.Mutation Number of test subjects with HIV type B and mutated i-th amino acid.

TypeC.NoMutation Number of test subjects with HIV type C and non-mutated i-th amino acid.

TypeB.NoMutation Number of test subjects with HIV type B and non-mutated i-th amino acid.

Thus, each row describes a 2x2 table:

Subject 1	mutation	no mutation
Type C	$X_{i,1}$	$X_{i,3}$
Type B	$X_{i,2}$	$X_{i,4}$

Note

The original `hiv` dataset has been taken from the `fdrDiscreteNull` package, where it is named `hivdata`.

References

Gilbert, P. B. (2005). A modified false discovery rate multiple-comparisons procedure for discrete data, applied to human immunodeficiency virus genetics. *Journal of the Royal Statistical Society*, **54**(1), pp. 143-158. doi:10.1111/j.14679876.2005.00475.x

---

`listerdata` *Lister data*

Description

This dataset has been analyzed and provided by the listed reference. There are around 22,000 cytosines, each of which is under two conditions. For each cytosine under each condition, there is only one replicate. The discrete count for each replicate can be modeled by binomial distribution, and Fisher's exact test can be applied to assess if a cytosine is differentially methylated. The filtered data `lister` contains cytosines whose total counts for both lines are greater than 5 and whose count for each line does not exceed 25.

Usage

```
data("listerdata")

data("listerdata_four_columns")
```

Format

`listerdata` is a `data.frame` with 3,525 rows and the following two columns:

Col0_Counts Degree of methylation of the i-th cytosine in reference genome.

Met13_Counts Degree of methylation of the i-th cytosine in mutated genome.

Thus, each row describes a 2x2 table:

AT1G01070.1	This cytosine	All other cytosines
Col0 counts	$X_{i,1}$	34,244 - $X_{i,1}$
Met13 counts	$X_{i,2}$	39,342 - $X_{i,2}$

`listerdata_four_columns` is a `data.frame` with 3,525 rows and the following four columns:

Col0_Counts.ThisCyto Degree of methylation of the i-th cytosine in reference genome.

Met13_Counts.ThisCyto Degree of methylation of the i-th cytosine in mutated genome.

Col0_Counts.AllOtherCytos Degree of methylation of all other cytosines in reference genome.

Met13_Counts.AllOtherCytos Degree of methylation of all other cytosines in mutated genome.

AT1G01070.1	This cytosine	All other cytosines
Col0 counts	$X_{i,1}$	$X_{i,3}$
Met13 counts	$X_{i,2}$	$X_{i,4}$

Note

The original `listerdata` dataset has been taken from the `fdrDiscreteNull` package.

References

Lister, R., O'Malley, R., Tonti-Filippini, J., Gregory, B. D., Berry, C. C., Millar, A. H. & Ecker, J. R. (2008). Highly integrated single-base resolution maps of the epigenome in arabidopsis, *Cell* **133**(3), pp. 523-536. doi:10.1016/j.cell.2008.03.029

`reconstruct_four` *Reconstruct a set of reformatted four-fold tables*

Description

Sometimes, fourfold tables are reformatted by replacing rows or columns by marginal totals. This makes it impossible to use them straight away for statistical tests like Fisher's exact test. But with that knowledge, the missing values can easily be restored. The `reconstruct_four` function uses a set of such reduced tables, stored row-wise in a matrix or a data frame, and rebuilds the two reformatted cells when they were replaced by marginal totals.

Usage

```
reconstruct_four(dat, idx_marginals = NULL, colnames_add = NULL)
```

Arguments

`dat` integer matrix or data frame with exactly two columns; each row represents the first column of a 2x2 matrix for which the other two values are to be computed and appended to `dat` as two new columns; real numbers will be coerced to integer.

`idx_marginals` integer vector of exactly two values or `NULL` (the default) indicating the columns of `dat` that contain the marginal totals; if `NULL`, the last two columns are used.

`colnames_add` character vector of exactly two unique character strings or `NULL` (the default), which contains the desired headers of the new (reconstructed) columns of the input; if `NULL`, the headers of the marginal totals are used.

Value

An integer data frame with four columns.

Examples

```
X1 <- c(4, 2, 2, 14, 6, 9, 4, 0, 1)
X2 <- c(0, 0, 1, 3, 2, 1, 2, 2, 2)
N1 <- rep(148, 9)
N2 <- rep(132, 9)

df1 <- data.frame(X1, X2, N1, N2)
reconstruct_four(df1, colnames_add = c("Y1", "Y2"))
# same as reconstruct_four(df1, c(3, 4), c("Y1", "Y2"))

df2 <- data.frame(X1, N1, X2, N2)
reconstruct_four(df2, c(2, 4), c("Y1", "Y2"))
```

`reconstruct_two` *Reconstruct a set of four-fold tables from rows or columns*

Description

In some situations, fourfold tables are reduced to two elements, which makes it impossible to use them straight away for statistical tests like Fisher's exact test. But sometimes, when all tables had the same known marginal sums, the missing values can be restored using that additional information. The `reconstruct_two` function uses a set of such reduced tables, stored row-wise in a matrix or a data frame, and rebuilds the two missing columns from automatically computed or given marginal totals.

Usage

```
reconstruct_two(
  dat,
  totals = NULL,
  insert_at = NULL,
  colnames_add = NULL,
  colnames_prepend = NULL,
  colnames_append = NULL,
  colnames_sep = "_"
)
```

Arguments

`dat` integer matrix or data frame with exactly two columns; each row represents the first column of a 2x2 matrix for which the other two values are to be computed and appended to `dat` as two new columns; real numbers will be coerced to integer.

`totals` integer vector of exactly one or two values or `NULL` (the default); the new columns will be derived by subtracting the existing column values from `totals`; if `NULL`, the sums of the two existing columns of `dat` are used.

`insert_at` integer vector of exactly two values between 1 and 4 or `NULL` (the default) indicating the indices at which the values are to be inserted; if `NULL`, the new values are appended at the end, i.e. at positions 3 and 4.

`colnames_add` character vector of exactly two unique character strings or `NULL` (the default), which contains the desired headers of the new (reconstructed) columns of the input; if `NULL`, the headers of `dat` are used (with appended strings; see below).

`colnames_prepend` character vector of exactly two unique character strings (`NA`s are allowed) or `NULL` (the default); the first string will be prepended to the headers of the original headers of `dat`, while the second is used in the same manner for the reconstructed columns.

`colnames_append` character vector of exactly two unique character strings (`NA`s are allowed) or `NULL` (the default); the first string will be appended to the headers of the original headers of `dat`, while the second is used in the same manner for the reconstructed columns; if `colnames_add = NULL` and `colnames_append = NULL`, `c("A", "B")` will be used.

`colnames_sep` a single character or `NULL` (the default) giving the separator for combining `colnames_prepend` and `colnames_append` with the column names.

Value

An integer data frame with four columns.

Examples

```
data(amnesia)
amnesia_four_columns <- reconstruct_two(
  amnesia,
  NULL,
  NULL,
  NULL,
  NULL,
  c("ThisDrug", "AllOtherDrugs"),
  "."
)
head(amnesia_four_columns)

data(hiv)
hiv_four_columns <- reconstruct_two(
  hiv,
  73,
  NULL,
  NULL,
  NULL,
  c("Mutation", "NoMutation"),
  "."
)
head(hiv_four_columns)
```

```
data(listerdata)
listerdata_four_columns <- reconstruct_two(
  listerdata,
  c(34244, 39342),
  NULL,
  NULL,
  NULL,
  c("This_Cyto", "All_Other_Cytos"),
  "_"
)
head(listerdata_four_columns)
```

Index