Package ‘DiscreteDatasets’

March 4, 2026

Type Package

Title Example Data Sets for Use with Discrete Statistical Tests

Version 0.2.0

Date 2026-03-03

Description Provides several data sets for use with discrete statistical tests
and discrete multiple testing procedures.

License GPL-3

Language en-GB

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 7.3.3

NeedsCompilation no

Author Christina Kihn [aut],
Sebastian Döhler [aut] (ORCID: <https://orcid.org/0000-0002-0321-6355>),
Florian Junge [aut, cre] (ORCID:
<https://orcid.org/0009-0001-6856-6938>),
Lukas Klein [ctb] (ORCID: <https://orcid.org/0000-0002-1413-497X>),
Iqraa Meah [ctb] (ORCID: <https://orcid.org/0000-0003-0519-8039>)

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

Repository CRAN

Date/Publication 2026-03-04 06:31:07 UTC

Contents

`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`> (ORCID)

Authors:

- Christina Kihn
- Sebastian Döhler <`sebastian.doehler@h-da.de`> (ORCID)

Other contributors:

- Lukas Klein (ORCID) [contributor]
- Iqraa Meah (ORCID) [contributor]

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_four_columns")
```

Format

`airway` is a `data.frame` with 63,677 rows and 2 columns. Each row corresponds to a specific gene and each column to treatment and control groups:

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

`Control` 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 - $X_{i,1}$
Control	$X_{i,2}$	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

`Control.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

`Control.AllOtherGenes` Number of reads for all other genes in all untreated samples

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

ENSG00000000003	This gene	All other genes
Treatment	$X_{i,1}$	$X_{i,3}$
Control	$X_{i,2}$	$X_{i,4}$

Details

The cell lines of the treatment samples were treated with dexamethasone, whereas the cell lines of the control 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 of eight samples. These were then merged into treatment and non-treatment groups.

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 Medicines and Healthcare products Regulatory Agency (MHRA) database, the number of cases with amnesia as an adverse event, and the number of cases with other adverse event for this drug are recorded. 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, e.g.:

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 by Heller & Gur from the Drug Analysis Prints, published by the MHRA. See References section 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

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 C_b 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` Observed coverage of each base

`expected` 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.

`federalist` *The Federalist Papers word count data*

Description

Author assignments and counts of the 1,500 most common words from "The Federalist" articles.

"The Federalist Papers" are a set of 85 articles written under the pseudonym "Publius" to promote the ratification of the US constitution by Alexander Hamilton, James Madison and John Jay in 1787 and 1788. There are multiple sources which attribute the articles to their real authors. We use the attributions by the Project Gutenberg and the correction by the authors of the `syllogi` package. This task has been a popular problem in natural language processing. One of the most prominent examples is the work by Mosteller and Wallace (1964) who used the word frequencies to attribute the disputed articles to their authors.

The data provided in this package was prepared with the following steps by employing the `tm` package:

1. Load the texts from the `syllogi` package,
2. Lowercase,
3. Remove punctuation,
4. Strip whitespace,
5. Remove the texts by Jay, one text co-authored by Madison and Hamilton together, and the redundant version of article 70,
6. Find the 1,500 most common words for each author,
7. Count the occurrences of these words in the texts.

Usage

```
data("federalist")
```

Format

`federalist` is a `data.frame` with 77 rows and 1,984 columns:

`doc_no` Article number

`doc_author` Author of the article, according to Project Gutenberg

`...` The remaining 1,982 columns are the word counts

References

Watson, G. S. (1966). Review: Frederick Mosteller, David L. Wallace, Inference and Disputed Authorship: The Federalist. *The Annals of Mathematical Statistics*, **37**(1), 308-312. doi:10.1214/aoms/1177699628

Donoho, D. L., & Kipnis, A. (2022). Higher criticism to compare two large frequency tables, with sensitivity to possible rare and weak differences. *Annals of Statistics*, **50**(3), 1447-1472. doi:10.1214/21AOS2158

Feinerer, I., & Hornik, K. (2024). tm: Text Mining Package. R package version 0.7-15. *CRAN*. `https://CRAN.R-project.org/package=tm`

Studyvin, J. (2024). syllogi: Collection of Data Sets for Teaching Purposes. R package version 1.0.3. *CRAN*. `https://CRAN.R-project.org/package=syllogi`

`hiv` *HIV data*

Description

This data set has been analysed 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:

Amino Acid 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

`impc2015_excerpt` *Excerpt of aggregated data of the International Mouse Phenotyping Consortium*

Description

The **International Mouse Phenotyping Consortium** (IMPC) aims to identify the function of every gene in the mouse genome. It spearheads the whole process from lab studies to data distribution. For each protein-encoding gene of interest, an *in vivo* study takes place to obtain mutant and non-mutant mice, the mutant mice being the one where a specific gene is knocked out at the embryo stage. A wide range of phenotypic traits covering multiple physiological systems, such as visual and neurological phenotypes, are then measured in these mice to better understand the biological function of the knocked-out gene. The collected data is standardised and freely accessible via the consortium's website (see *Source* section).

This dataset contains an excerpt of the results of 5,000 aggregated experiments on male and female mice about 1,692 knockout genes, conducted by 10 different phenotyping centres up to December 2015.

Usage

```
data("impc2015_excerpt")
```

Format

`impc2015_excerpt` is a `data.frame` with 5,000 rows representing experiments with the following ten columns:

`Organisation.Name` Name of the phenotyping organisation that conducted the experiment

`Gene.Symbol` Name of the relevant knockout gene

`Female.Mutant.Atypical` Number of female mice in the experimental group **exhibiting abnormalities**

`Female.Mutant.Typical` Number of **typical** female mice in the experimental group

`Female.Control.Atypical` Number of female mice in the control group **exhibiting abnormalities**

`Female.Control.Typical` Number of **typical** female mice in the control group

`Male.Mutant.Atypical` Number of female mice in the experimental group **exhibiting abnormalities**

`Male.Mutant.Typical` Number of **typical** male mice in the experimental group

`Male.Control.Atypical` Number of male mice in the control group **exhibiting abnormalities**

`Male.Control.Typical` Number of **typical** male mice in the control group

Thus, each line describes two 2x2 tables (one for each sex), e.g.:

Female mice	With abnormalities	Typical
Experimental group	$X_{i,3}$	$X_{i,4}$
Control group	$X_{i,5}$	$X_{i,6}$

and

Male mice	With abnormalities	Typical
Experimental group	$X_{i,7}$	$X_{i,8}$
Control group	$X_{i,9}$	$X_{i,10}$

Details

This data is derived from a dataset used in Karp et al. (2017) (see *References* section). The original data was collected by the IMPC.

Source

Dataset of Karp et al. (2017)
IMPC data

References

N. A. Karp et al. (2017). Prevalence of sexual dimorphism in mammalian phenotypic traits. *Nature Communications*, **8**, 15475. doi:10.1038/ncomms15475

`listerdata` *Lister data*

Description

This dataset has been analysed 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 modelled by binomial distribution, and Fisher's exact test can be applied to assess if a cytosine is differentially methylated. The filtered data `listerdata` 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 original headers of `dat`, while the second is used in the same manner for the new (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 original headers of `dat`, while the second is used in the same manner for the new (reconstructed) columns; if both `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