Package: SRSim (via r-universe)

September 3, 2024

Type Package

Title Spontaneous Reporting Simulator (SRSim)

Version 0.1

Date 2018-01-18

Author Louis Dijkstra [aut, cre], Marco Garling [ctb]

Maintainer Louis Dijkstra <`dijkstra@leibniz-bips.de`>

Description A package for simulating spontaneous reporting data as used in the field of pharmacovigilance.

Depends R (>= 3.2.3), Rcpp, RcppArmadillo, RcppProgress, dplyr, tidyr, mvtnorm, corpcor, pcalg, Rgraphviz

LinkingTo Rcpp, RcppArmadillo, RcppProgress

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

URL `https://github.com/bips-hb/srsim`

BugReports `https://github.com/bips-hb/srsim/issues`

Repository https://bips-hb.r-universe.dev

RemoteUrl https://github.com/bips-hb/srsim

RemoteRef HEAD

RemoteSha e53edbd2c1a0e393709809ff863d6433c3b6d538

Contents

convert2Tables 2
create2x2TablesDAGRcpp 3
f_beta0 4
generateDAG 4
simulateReport 5

simulateSRS 6
SRSim 7
validReport 8

Index **9**

`convert2Tables` *Create 2 x 2 Tables from DAG based SR data*

Description

Creates a data frame with all 2 x 2 contigency tables given a simulated data set from `simulateSRS`. The tables are organized as follows:

	event j	not event j	*total*
drug i	`a`	`c`	`a + c`
not drug i	`b`	`d`	`b + d`
total	`a + b`	`c + d`	`n_reports`

Usage

```
convert2Tables(sr)
```

Arguments

`sr`	The output generated by the `simulateSRS` function

Value

A data frame where each row represents a 2 x 2 table. The columns represent:

`drug_id`	The ID of the drug
`event_id`	The ID of the event
`prob_drug`	The marginal probability of that drug
`prob_event`	The marginal probability of that event
`or`	The increase of the odds ratio when the drug is on the report
`associated`	`TRUE` is there is a non-zero correlation, `FALSE` otherwise
`a`	Number of times the drug and event appeared together in a report
`b`	Number of times the event appeared without the drug in a report
`c`	Number of times the drug appeared without the event in a report
`d`	Number of times the drug and event both did not appear in a report

See Also

`simulateSRS`

Examples

```
sr <- simulateSRS()
tables <- create2x2Tables(sr)
```

`create2x2TablesDAGRcpp`

Create 2 x 2 Tables

Description

Creates a data frame containing all 2 x 2 contingency tables from the results generated by `simulateSRS`. See the R wrapper function `convert2Tables` for more information.

Usage

```
create2x2TablesDAGRcpp(reports, prob_drugs, prob_events, n_parents, parent_id,
  beta1)
```

Arguments

`reports`	A binary matrix. Each row is a report
`prob_drugs`	A vector with the marginal probabilities of the drugs
`prob_events`	A vector with the marginal probabilities of the events
`n_parents`	A vector with the number of parents for each of the nodes
`parent_id`	A vector with the id of the parent
`beta1`	A vector with the regression coefficients for each node

Value

A dataframe. A description of the columns can be found in the commentary for the function `convert2Tables`

See Also

`convert2Tables`

`f_beta0` *Determining the Intercept*

Description

`f_beta0` is used by `simulateSRS` to find the appropriate intercept for a logistic regression model.

Usage

```
f_beta0(beta0, margprob, beta1, margprob_parent)
```

Arguments

`beta0`	The intercept
`margprob`	The marginal probability
`beta1`	The regression coefficient of the parent variable
`margprob_parent`	The marginal probability of the parent

Value

The absolute difference between the desired marginal probability and the current marginal probability

---

`generateDAG` *Directed Acyclic Graph for SR Data*

Description

`generateDAG` creates a directed acyclic graph used for generating spontaneous reporting data

Usage

```
generateDAG(n_drugs = 10, n_events = 10, n_innocent_bystanders = 5,
  n_correlated_pairs = 2)
```

Arguments

`n_drugs`	Number of drugs (Default: 10)
`n_events`	Number of adverse events (Default: 10)
`n_innocent_bystanders`	Number of innocent bystanders (Default: 5)
`n_correlated_pairs`	Number of drug-event pairs that will be associated (Default: 2)

Value

The DAG as an `igraph` object

See Also

`simulateSRS`

`simulateReport` *Simulate a single Report*

Description

Returns a single report for a spontaneous reporting system. Note that this function does not check whether the report is 'valid', see `validReport`

Usage

```
simulateReport(n_drugs, n_events, n_parents, beta0, beta1, parent_id)
```

Arguments

`n_drugs`	The number of drugs
`n_events`	The number of events
`n_parents`	A vector with the number of parents for each of the nodes
`beta0`	A vector with the intercepts for the logistic regression models
`beta1`	A vector with the regression coefficients for each node
`parent_id`	A vector with the id of the parent

Value

A binary vector

`simulateSRS` *Simulating a Spontaneous Reporting System*

Description

`simulateSRS` simulates a spontaneous reporting (SR) data set. The relationships between the drugs and the adverse events (AEs) are specified by a directed acyclic graph (DAG), see `generateDAG`.

Each report to a SRS contains two lists:

1. the drugs to which the patient was (thought to be) exposed to, and
2. the AEs that the patient experienced.

We will represent each report as a binary vector. The first items represent whether the patient was exposed to the drug (`1` if he/she was, and `0` otherwise). The second part represents whether the patient experienced the event or not (`1` if he/she did, and `0` otherwise). For example, if there are 3 drugs and 4 events in total, a typical report could be

$$0101100$$

which represents that the patient was exposed to drug 2 (but not to drug 1 and 3), and experienced event 1 and 2 (but not 3 and 4). The simulation results in a binary matrix where each row is a report.

Valid Reports Not any binary sequence is a valid report. Each report should contain at least one drug and at least one event (otherwise it would never been sent to the spontaneous reporitng sytem). While generating reports, we make sure that this is indeed the case. When one does not want to check the validity and wants to allow any binary sequence, one can set `valid_reports` to `FALSE`.

Usage

```
simulateSRS(n_reports = 100, n_drugs = 10, n_events = 10,
  alpha_drugs = 1, beta_drugs = 20, alpha_events = 1, beta_events = 20,
  n_innocent_bystanders = 5, bystander_prob = 0.9, n_correlated_pairs = 2,
  theta = 2, valid_reports = TRUE, seed = NULL, verbose = TRUE)
```

Arguments

`n_reports`	Number of reports (Default: 100)
`n_drugs`	Number of drugs (Default: 10)
`n_events`	Number of adverse drug events (Default: 10)
`alpha_drugs`	Alpha parameter for the drug marginal probabilities (Default: 1.0)
`beta_drugs`	Beta parameter for the drug marginal probabilities (Default: 20.0)
`alpha_events`	Alpha parameter for the event marginal probabilities (Default: 1.0)
`beta_events`	Beta parameter for the event marginal probabilities (Default: 20.0)
`n_innocent_bystanders`	Number of innocent bystanders (Default: 5)

`bystander_prob`	The conditional probability of the innocent bystander being one when the drug that is actually causing the AE is equal to 1. This parameter corresponds to γ in the paper (Default: .9)
`n_correlated_pairs`	Number of drug-AE pairs that are associated (Default: 2)
`theta`	Increase in odds-ratio when there is an edge going from a drug to an AE (Default: 2.0). In case theta is a vector of length two, the odds ratio is drawn from a truncated Normal distribution with mean `theta[1]` and variance `theta[2]`
`valid_reports`	If `TRUE`, only valid reports (with at least one drug and at least one AE) are accepted. (Default: `TRUE`)
`seed`	The seed used by the RNG (Default: automatically set)
`verbose`	Verbosity (Default: `TRUE`)

Value

`sr`	A binary data frame with the simulated reports. The columns are named `drug1`, `drug2` ..., `event1`, `event2`, ...
`dag`	The directed acycled graph as an `igraph` object
`nodes`	A tibble with all the information on each node/variate: • `label` The label for each node/variate • `in_degree` The number of edges pointing to the node • `id` The ID of each node (simple integer) • `parent_id` The ID of the parent node - if any. Otherwise equal to `-1` • `margprob` The marginal probability of the node/variate • `beta0` The intercept in the logistic regression model for that node • `beta1` The regression coefficient in the logistic regression model for the parent
`prob_drugs`	A vector with marginal probabilities of the drugs
`prob_events`	A vector with marginal probabilities of the events

See Also

`convert2Tables`, `generateDAG`

`SRSim` *SRSim*

Description

A package for simulating data from spontaneous reporting systems (SRSs). Spontaneous reporting systems are used to

Author(s)

Louis Dijkstra & Marco Garling

`validReport` *Valid Report*

Description

Checks whether the given report is 'valid'. A report should contain at least one drug and at least one event (otherwise it would never been sent to the spontaneous reporitng sytem anyway).

Usage

```
validReport(report, n_drugs, n_events)
```

Arguments

`report`	A logical matrix with one row and `n_drugs` + `n_events` columns
`n_drugs`	The total number of drugs
`n_events`	The total number of events

Value

`TRUE` when the report is valid, `FALSE` otherwise

Index

`convert2Tables`, 2, *3*, *7*
`create2x2TablesDAGRcpp`, 3

`f_beta0`, 4

`generateDAG`, 4, *6*, *7*

`simulateReport`, 5
`simulateSRS`, *2–5*, 6
`SRSim`, 7
`SRSim-package` *(*`SRSim`*)*, 7

`validReport`, *5*, 8