Package 'faers'

November 12, 2026

Title R interface for FDA Adverse Event Reporting System

Version 1.8.0

BugReports https://github.com/WangLabCSU/faers

Description
The FDA Adverse Event Reporting System (FAERS) is a database used for the spontaneous reporting of adverse events and medication errors related to human drugs and therapeutic biological products. faers pacakge serves as the interface between the FAERS database and R. Furthermore, faers pacakge offers a standardized approach for performing pharmacovigilance analysis.

License MIT + file LICENSE

Depends R (>= 3.5.0)

Imports BiocParallel, brio, cli, curl (>= 6.0.0), data.table, httr2 (>= 1.0.0), MCMCpack, methods, openEBGM, rlang, rvest, tools, utils, vroom, xml2

Suggests BiocStyle, countrycode, knitr, rmarkdown, testthat (>= 3.0.0)

biocViews Software, DataImport, BiomedicalInformatics, Pharmacogenomics, Pharmacogenomics

Encoding UTF-8

ByteCompile true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Config/testthat/edition 3

Config/testthat/parallel true

VignetteBuilder knitr

Collate 'athena.R' 'available.R' 'meddra.R' 'class-FAERS.R' 'combine.R' 'counts.R' 'dedup.R' 'download.R' 'faers-package.R' 'faers.R' 'fda_drugs.R' 'import-standalone-assert.R' 'import-standalone-cli.R' 'import-standalone-obj-type.R' 'import-standalone-pkg.R' 'load.R' 'merge.R' 'meta.R' 'methods-FAERS.R' 'parse.R' 'period.R' 'phv_.R' 'phv_ebgm.R' 'rxnorm.R' 'sample.R' 'signal.R' 'standardize.R' 'unify.R' 'utils-assert.R' 'utils-file.R' 'utils-str.R' 'utils.R'

git_url https://git.bioconductor.org/packages/faers

git_branch RELEASE_3_23

git_last_commit 0a111c6

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Yun Peng [aut, cre] (ORCID: <https://orcid.org/0000-0003-2801-3332>),
YuXuan Song [aut],
Caipeng Qin [aut],
JiaXing Lin [aut]

Maintainer Yun Peng <yunyunp96@163.com>

Contents

faers-package . . . 3
athena . . . 3
faers . . . 4
FAERS-class . . . 5
faers_available . . . 7
faers_before_period . . . 8
faers_clearcache . . . 8
faers_combine . . . 9
faers_counts . . . 10
faers_dedup . . . 11
faers_download . . . 12
faers_get . . . 13
faers_load . . . 15
faers_merge . . . 16
faers_meta . . . 18
faers_parse . . . 18
faers_phv_signal . . . 20
faers_standardize . . . 21
fda_drugs . . . 22
MedDRA-class . . . 23
phv_signal . . . 24

Index **29**

faers-package *faers: R interface for FDA Adverse Event Reporting System*

Description

The FDA Adverse Event Reporting System (FAERS) is a database used for the spontaneous reporting of adverse events and medication errors related to human drugs and therapeutic biological products. faers pacakge serves as the interface between the FAERS database and R. Furthermore, faers pacakge offers a standardized approach for performing pharmacovigilance analysis.

Author(s)

Maintainer: Yun Peng <yunyunp96@163.com> (ORCID)

Authors:

- YuXuan Song <yuxuan_song2021@163.com>
- Caipeng Qin <qincaipeng@pkuph.edu.cn>
- JiaXing Lin <1570851599@qq.com>

See Also

Useful links:

- Report bugs at https://github.com/WangLabCSU/faers

athena *Read and Parse ATHENA VOCABULARIES data*

Description

Read and Parse ATHENA VOCABULARIES data

Usage

```
athena(use = NULL, list = FALSE, force = FALSE, url = NULL)
```

Arguments

use	An atomic character specifying the files to use with values in "concept", "domain", "concept_class", "concept_relationship", "concept_ancestor", "concept_synonym", "drug_strength", "relationship", and "vocabulary".
list	A boolean value, should it only list files in the ATHENA VOCABULARIES data?
force	A boolean value. If set to TRUE, it indicates the retrieval of VOCABULARIES data in the url directly, bypassing the cache.
url	A string of url for ATHENA VOCABULARIES data. You must provide it to cache the file when you firstly run this function.

Value

- if `list` = `TRUE`, an atomic character.
- if `list` = `FALSE`, a data.table if `use` is a string or otherwise a list of data.table.

`faers` *Download and parse FAERS Quarterly Data files*

Description

Download and parse FAERS Quarterly Data files

Usage

```
faers(
  years,
  quarters,
  format = NULL,
  dir = getwd(),
  compress_dir = dir,
  handle_opts = list()
)
```

Arguments

`years`	An atomic integer indicates years for which data are required.
`quarters`	An atomic character, only "q1", "q2", "q3", and "q4" are allowed.
`format`	File format to used, only "ascii" and "xml" are availabe. Default: "ascii".
`dir`	The destination directory for any downloads. Defaults to current working dir.
`compress_dir`	A string specifies the directory to extract files to. It will be created if necessary.
`handle_opts`	Extra handle options passed to each request new_handle.

Value

A FAERSxml or FAERSascii object.

Examples

```
# you must change `dir`, as the file included in the package is sampled
data <- faers(2004, "q1",
    dir = system.file("extdata", package = "faers"),
    compress_dir = tempdir()
)
```

FAERS-class *FAERS class*

Description

Provide a container for FAERS Quarterly Data file

Usage

```
## S4 method for signature 'FAERS'
show(object)

## S4 method for signature 'FAERSascii'
show(object)

faers_data(object, ...)

## S4 method for signature 'FAERS'
faers_data(object)

faers_year(object)

## S4 method for signature 'FAERS'
faers_year(object)

faers_quarter(object)

## S4 method for signature 'FAERS'
faers_quarter(object)

faers_period(object)

## S4 method for signature 'FAERS'
faers_period(object)

faers_meddra(object, ...)

## S4 method for signature 'FAERS'
faers_meddra(object, use = NULL)

faers_deleted_cases(object, ...)

## S4 method for signature 'FAERSascii'
faers_deleted_cases(object)

faers_header(object)
```

```
## S4 method for signature 'FAERSxml'
faers_header(object)
```

Arguments

object	A FAERS object.
...	Other arguments passed to specific methods.
use	A string, what meddra data to use, "hierarchy" or "smq". If NULL, a MedDRA will be returned. Only used when object has been standardized

Details

- faers_data: Extract the data slot.
- faers_year: Extract the year slot.
- faers_quarter: Extract the quarter slot.
- faers_period: A data.table combine the year and quarter slot.
- faers_meddra: Extract the meddra slot. If object have never been standardized, always return NULL.
- faers_deleted_cases: Extract the deletedCases slot.

Value

See details.

Slots

year An integer specifies the year information.

quarter A string specifies the quarter information.

data For FAERSxml, a data.table. For FAERSascii, a list of data.table.

meddra A MedDRA or NULL representing the meddra data used for standardization.

format A string of "ascii" or "xml" indicates the file format used.

deletedCases An atomic character, as of 2019 Quarter one there are new files that lists deleted cases. faers_dedup will remove cases in this slot.

standardization A bool, indicates whether standardization has been performed.

deduplication A bool, indicates whether deduplication has been performed.

Examples

```
# ususaly we use faers() function to create a `FAERS` object
# you must change `dir`, as the file included in the package is sampled
data <- faers(2004, "q1",
    dir = system.file("extdata", package = "faers"),
    compress_dir = tempdir()
)
faers_data(data)
faers_year(data)
```

```
faers_quarter(data)
faers_period(data)
faers_meddra(data)
faers_deleted_cases(data)
```

`faers_available` *Check if FAERS year*

Description

This function check if data for the years and quarters selected are available at FAERS to be downloaded.

Usage

```
faers_available(years, quarters, force = FALSE, internal = FALSE)
```

Arguments

`years`	An atomic integer indicates years for which data are required.
`quarters`	An atomic character, only "q1", "q2", "q3", and "q4" are allowed.
`force`	A boolean value. If set to `TRUE`, it indicates the retrieval of information about all records' metadata in the FAERS Quarterly Data Extract Files Site, bypassing the cache.
`internal`	A boolean value. It determines whether to use the internal data associated with the package.

Value

A logical indicates FAERS can have data for the `years` and `quarters` required?

Examples

```
faers_available(c(2011, 2023), c("q1", "q2"))
```

faers_before_period *Test whether years and quarters are before specified period*

Description

Test whether years and quarters are before specified period

Usage

```
faers_before_period(years, quarters, y, q, inclusive = TRUE)
```

Arguments

years	An atomic integer indicates years to test.
quarters	An atomic character indicates quarters to test, only "q1", "q2", "q3", and "q4" are allowed.
y	An integer, specifying the period year.
q	A string, specifying the period quarter.
inclusive	A bool, whether to include the period specifid.

Value

An atomic logical with the same length of the max length of `years` and `quarters`.

Examples

```
faers_before_period(c(2011, 2012), c("q1", "q3"), 2011, "q2")
```

faers_clearcache *Remove caches*

Description

Remove caches

Usage

```
faers_clearcache(caches = NULL, force = FALSE)
```

Arguments

caches	An atomic character, indicates what caches to remove? Only "metadata", "fdadrugs", and "athena" can be used. If `NULL`, all caches will be removed.
force	logical. Should permissions be changed (if possible) to allow the file or directory to be removed?

Value

Path of the deleted directory invisiblely

Examples

```
faers_clearcache()
```

`faers_combine` *Combine FAERS objects from different Quarterly files.*

Description

Packed all `FAERSascii` or `FAERSxml` objects into a single `FAERSascii` or `FAERSxml` object. It is important to note that all data passed to these functions via the `...` argument must belong to the different FAERS objects, indicating that they have the different period data (as defined by faers_period).

Usage

```
faers_combine(...)
```

Arguments

`...`	Multiple `FAERSxml` or `FAERSascii` objects or a list containing `FAERSxml` or `FAERSascii` objects. Objects can be standardized by faers_standardize but cannot be de-duplicated by faers_dedup. If we combine deduplicated objects from different quarterly data files, duplicate reports will be introduced again.

Value

A FAERSxml or FAERSascii object.

Examples

```
# the files included in the package are sampled
data1 <- faers_parse(
    system.file("extdata", "aers_ascii_2004q1.zip", package = "faers"),
    compress_dir = tempdir()
)
data2 <- faers_parse(
    system.file("extdata", "faers_ascii_2017q2.zip", package = "faers"),
    compress_dir = tempdir()
)
faers_combine(data1, data2)
```

faers_counts *Counting the number of unique case for each event*

Description

Counting the number of unique case for each event

Usage

```
faers_counts(.object, ...)

## S4 method for signature 'FAERSascii'
faers_counts(
  .object,
  .events = "soc_name",
  .fn = NULL,
  ...,
  .field = "reac",
  .na.rm = FALSE
)
```

Arguments

.object	A FAERSascii object.
...	Other arguments passed to specific methods, for FAERSascii method, other arguments passed to .fn().
.events	A character specify the events column(s) in the .field data to count the unique primaryid. If multiple columns were selected, the combination for all columns will define the interested events.
.fn	A function or formula defined the preprocessing function before creating contingency table, with the .field data as the input and return a data.table. Note: When using the set* or := function from data.table with the "demo", "drug", "ther", "rpsr", and "outc" data, exercise caution as these functions directly modify the internal data. In such cases, it is advisable to use the copy function first. If a **function**, it is used as is. If a **formula**, e.g. ~ .x + 2, it is converted to a function with up to two arguments: .x (single argument) or .x and .y (two arguments). The . placeholder can be used instead of .x. This allows you to create very compact anonymous functions (lambdas) with up to two inputs. If a **string**, the function is looked up in globalenv().
.field	A string indicates the interested FAERS fields to use. Only values "demo", "drug", "indi", "ther", "reac", "rpsr", and "outc" can be used.
.na.rm	A bool, whether NA value in .events column(s) should be removed.

Value

A data.table object.

Examples

```
# you must change `dir`, as the files included in the package are sampled
data <- faers(c(2004, 2017), c("q1", "q2"),
    dir = system.file("extdata", package = "faers"),
    compress_dir = tempdir()
)
## Not run:
# you must standardize and deduplication before disproportionality analysis
# you should replace `meddra_path` with yours
data <- faers_standardize(data, meddra_path)
data <- faers_dedup(data)
faers_counts(data)

## End(Not run)
std_data <- readRDS(system.file("extdata", "standardized_data.rds",
    package = "faers"
))
faers_counts(std_data)
```

---

faers_dedup *Tidy up FAERS Quarterly Data with duplicate records removed*

Description

Tidy up FAERS Quarterly Data with duplicate records removed

Usage

```
faers_dedup(object, ...)

## S4 method for signature 'FAERSascii'
faers_dedup(object, remove_deleted_cases = TRUE)

## S4 method for signature 'FAERSxml'
faers_dedup(object)

## S4 method for signature 'ANY'
faers_dedup(object)
```

Arguments

object	A FAERSascii object.
...	Other arguments passed to specific methods.
remove_deleted_cases	If TRUE, will remove all deletedCases from the final result.

Value

A FAERSascii object.

See Also

faers_standardize

Examples

```
# you must change `dir`, as the files included in the package are sampled
data <- faers(c(2004, 2017), c("q1", "q2"),
    dir = system.file("extdata", package = "faers"),
    compress_dir = tempdir()
)
## Not run:
# we must standardize firstly
# you should replace `meddra_path` with yours
data <- faers_standardize(data, meddra_path)
faers_dedup(data)

## End(Not run)
```

`faers_download` *Download FAERS data*

Description

This function downloads the FAERS data for selected years and quarters.

Usage

```
faers_download(years, quarters, format = NULL, dir = getwd(), ...)
```

Arguments

`years`	An atomic integer indicates years for which data are required.
`quarters`	An atomic character, only "q1", "q2", "q3", and "q4" are allowed.
`format`	File format to used, only "ascii" and "xml" are availabe. Default: "ascii".
`dir`	The destination directory for any downloads. Defaults to current working dir.
`...`	Extra handle options passed to each request multi_download.

Value

An atomic character for the path of downloaded files.

Examples

```
# you must change `dir`, as the file included in the package is sampled
# in this way, the file will downloaded from FAERS
faers_download(
    year = 2004, quarter = "q1",
    dir = system.file("extdata", package = "faers")
)
```

---

faers_get *Methods for FAERS class*

Description

Utils function for FAERSascii class.

Usage

```
faers_get(object, ...)

## S4 method for signature 'FAERSascii'
faers_get(object, field)

faers_mget(object, ...)

## S4 method for signature 'FAERSascii'
faers_mget(object, fields)

faers_primaryid(object, ...)

## S4 method for signature 'FAERSascii'
faers_primaryid(object)

## S4 method for signature 'FAERSascii,ANY,ANY,ANY'
x[i]

## S4 method for signature 'FAERSascii'
x[[i]]

## S4 method for signature 'FAERSascii'
x$name

faers_keep(object, ...)

## S4 method for signature 'FAERSascii'
faers_keep(object, primaryid = NULL, invert = FALSE)

faers_filter(.object, ...)
```

```
## S4 method for signature 'FAERSascii'
faers_filter(.object, .fn, ..., .field = NULL, .invert = FALSE)

faers_modify(.object, ...)

## S4 method for signature 'FAERSascii'
faers_modify(.object, .field, .fn, ...)
```

Arguments

object, .object	A FAERSascii object.
...	Other arguments passed to specific methods. For faers_filter: other arguments passed to .fn.
field	A string indicates the FAERS fields to use. Only values "demo", "drug", "indi", "reac", "ther", "rpsr", and "outc" can be used.
fields	A character vector specifying the fields to use. Only values "demo", "drug", "indi", "ther", "reac", "rpsr", and "outc" can be used.
x	A FAERSascii object.
i, name	Indices specifying elements to extract. For i, it will be okay to use integer indices.
primaryid	An atomic character or integer specifies the reports to keep. If NULL, will do nothing.
invert	A bool. If TRUE, will keep reports not in primaryid.
.fn	A function or formula, accept the field data as the input and return an atomic integer or character of primaryid you want to keep or remove based on argument .invert. If a **function**, it is used as is. If a **formula**, e.g. ~ .x + 2, it is converted to a function with up to two arguments: .x (single argument) or .x and .y (two arguments). The . placeholder can be used instead of .x. This allows you to create very compact anonymous functions (lambdas) with up to two inputs. If a **string**, the function is looked up in globalenv().
.field	A string indicating the FAERS data to be used as input for the .fn function to extract the primaryid or modify data. Only the following values can be used: "demo", "drug", "indi", "ther", "reac", "rpsr", and "outc". • faers_filter: Use .fn to extract primaryid. If NULL, .object will be passed directly to .fn. .fn should return an atomic integer or character of primaryid that you want to keep or remove based on the .invert argument. • faers_modify: Use .fn to modify the specified field data. You cannot use NULL here. .fn should always return a data.table.
.invert	A bool. If TRUE, will keep reports not returned by .fn.

Details

- `faers_get`: Extract a specific field data.table. For `reac` and `indi` field, meddra data will be automatically added if avaliable.
- `faers_mget`: Extract a list of field data.table. For `reac` and `indi` field, meddra data will be automatically added if avaliable.
- `[[`, `$`, and `[`: Extract a specific field data.table or a list of field data.table from FAERS object. Note: this just extract field data from `@data` slot directly. For usual usage, just use `faers_get` or `faers_mget`.
- `faers_primaryid`: Extract the `primaryid` from demo field.
- `faers_keep`: only keep data from specified `primaryid`. Note: `year`, `quarter`, `deletedCases` will be kept as the original. So make sure you didn't filter a whole period FAERS quarterly data, in this way, it's much better to run faers.
- `faers_filter`: apply a function to extract the wanted `primaryid`, then use `faers_keep` to keep data from these primaryids.

Value

See details.

Examples

```
# you must change `dir`, as the file included in the package is sampled
data <- faers(2004, "q1",
    dir = system.file("extdata", package = "faers"),
    compress_dir = tempdir()
)
faers_get(data, "indi")
data[["indi"]]
data$indi
faers_get(data, "demo")
data[["demo"]]
data$demo
faers_mget(data, c("indi", "drug"))
faers_mget(data, c("indi", "demo"))
faers_primaryid(data)
faers_keep(data, primaryid = sample(faers_primaryid(data), 20L))
faers_filter(data, .fn = function(x) {
    sample(x$primaryid, 100L)
}, .field = "demo")
```

---

faers_load *Load data attached in faers package*

Description

Load data attached in faers package

Usage

```
faers_load(nm)
```

Arguments

nm	A string of the data name. Available name: "irAEs".

Details

- irAEs: Immune-related adverse events examined in ICI-associated adverse events

Value

- irAEs: A data.table

References

- Chen Chen, Bin Wu, ChenYu Zhang, Ting Xu, Immune-related adverse events associated with immune checkpoint inhibitors: An updated comprehensive disproportionality analysis of the FDA adverse event reporting system, International Immunopharmacology

Examples

```
faers_load("irAEs")
```

---

faers_merge *Merge all FAERS field data into one*

Description

Merge all FAERS field data into one

Usage

```
faers_merge(object, ...)

## S4 method for signature 'FAERSascii'
faers_merge(object, fields = NULL, all = TRUE, all.x = all, all.y = all)

## S4 method for signature 'FAERSxml'
faers_merge(object)

## S4 method for signature 'ANY'
faers_merge(object)
```

Arguments

`object`	A FAERSascii or FAERSxml object.
`...`	Other arguments passed to specific methods.
`fields`	A character vector specifying the fields to use. Only values "demo", "drug", "indi", "ther", "reac", "rpsr", and "outc" can be used.
`all`	logical; `all = TRUE` is shorthand to save setting both `all.x = TRUE` and `all.y = TRUE`.
`all.x`	logical; if `TRUE`, rows from `x` which have no matching row in `y` are included. These rows will have 'NA's in the columns that are usually filled with values from `y`. The default is `FALSE` so that only rows with data from both `x` and `y` are included in the output.
`all.y`	logical; analogous to `all.x` above.

Details

Each pair of field data are merged based on "year", "quarter" and "primaryid". In cases where any pair of data contains information related to "drug_seq" elements, such as "drug_seq", "indi_drug_seq", or "dsg_drug_seq", "drug_seq" will be aligned as well. `fields` shall be organized in the subsequent sequence: 'demo', 'drug', 'indi', 'reac', 'ther', 'rpsr', and 'outc' and the merging sequence will correspondingly adhere to this order. Only the initial instance, of the "caseid" column will be preserved.

Value

A data.table object.

Note

You'd better only merge necessary data, otherwise a lot of memory will be consumed to merge all fields data.

Examples

```
# you must change `dir`, as the file included in the package is sampled
data <- faers(2004, "q1",
    dir = system.file("extdata", package = "faers"),
    compress_dir = tempdir()
)
faers_merge(data, "indi") # only one field is just like faers_get()
faers_merge(data, c("demo", "indi"))
```

faers_meta *List of FAERS data*

Description

The function lists the metadata for the FAERS databases currently available to download.

Usage

```
faers_meta(force = FALSE, internal = !curl::has_internet())
```

Arguments

force	A boolean value. If set to TRUE, it indicates the retrieval of information about all records' metadata in the FAERS Quarterly Data Extract Files Site, bypassing the cache.
internal	A boolean value. It determines whether to use the internal data associated with the package.

Value

A data.table reporting years, period, quarter, and file urls and file sizes.

See Also

https://fis.fda.gov/extensions/FPD-QDE-FAERS/FPD-QDE-FAERS.html

Examples

```
faers_meta()
```

faers_parse *Parse FAERS Quarterly Data*

Description

Parse FAERS Quarterly Data

Usage

```
faers_parse(
  path,
  format = NULL,
  year = NULL,
  quarter = NULL,
  compress_dir = getwd()
)
```

Arguments

path	A string specifies the path of FAERS Quarterly Data. You can pass the FAERS zip file directly, In this way, all files in the zip file will be extracted in compress_dir. Or, you can also uncompressed youself, and passed the directory contained the uncompressed files.
format	File format to used, only "ascii" and "xml" are availabe. Default: "ascii".
year	Year of the FAERS Quarterly Data. Coerced into integer, if NULL, this will be extracted from path.
quarter	String specifies quarter of the FAERS data, if NULL, this will be extracted from path.
compress_dir	A string specifies the directory to extract files to. It will be created if necessary.

Value

A FAERSxml or FAERSascii object.

Unification

For all fields data:

- All names have been converted to lowercase.
- Rename "isr" into "primaryid" for periods before 2012q3.

Field specific operations:

- demo:
  - Rename "gndr_cod" into "gender" for periods before 2014q2.
  - Rename "sex" into "gender" for periods after or equal to 2014q2.
  - Rename "case" and "i_f_cod" into "caseid" and "i_f_code" for legacy aers, before 2012q3.
  - "age_in_years" was added, measured in years.
  - "country_code" was added (encoded according to the iso2c standards), it will be convenient to translate it into other code with countrycode().
  - "sex" was added, which recoded "F" as "Female", "M" as "Male" and other values as NA.
- ther: Rename "drug_seq" into "dsg_drug_seq" for legacy aers, before 2012q3.
- indi: Rename "drug_seq" into "indi_drug_seq" for legacy aers, before 2012q3.
- outc: Rename "outc_code" into "outc_cod" for 2012q4 data

Examples

```
# the files included in the package are sampled
data <- faers_parse(
    system.file("extdata", "aers_ascii_2004q1.zip", package = "faers"),
    compress_dir = tempdir()
)
```

faers_phv_signal *Create contingency table and run disproportionality analysis*

Description

Create contingency table and run disproportionality analysis

Usage

```
faers_phv_table(.object, ..., .full, .object2)

## S4 method for signature 'FAERSascii,FAERSascii,missing'
faers_phv_table(.object, .events = "soc_name", ..., .full, .object2)

## S4 method for signature 'FAERSascii,missing,FAERSascii'
faers_phv_table(.object, .events = "soc_name", ..., .full, .object2)

faers_phv_signal(.object, ...)

## S4 method for signature 'FAERSascii'
faers_phv_signal(
  .object,
  .methods = NULL,
  ...,
  .phv_signal_params = list(),
  BPPARAM = SerialParam()
)
```

Arguments

.object	A FAERSascii object. The unique number of primaryids from .object will be regarded as n1..
...	Other arguments passed to specific methods. • faers_phv_table: other arguments passed to faers_counts. • faers_phv_signal: other arguments passed to faers_phv_table.
.full	A FAERSascii object with data from full data. In this way, .object must be a subset of .full. The unique number of primaryids from .full will be regarded as n.
.object2	A FAERSascii object with data from another interested drug, In this way, .object and .object2 should not be overlapped. The unique number of primaryids from object2 will be regarded as n0..
.events	A character specify the events column(s) in the .field data to count the unique primaryid. If multiple columns were selected, the combination for all columns will define the interested events.
.methods	Just an alias of method in phv_signal.

`.phv_signal_params`
: Other arguments passed to phv_signal.

`BPPARAM`
: An optional `BiocParallelParam` instance defining the parallel back-end to be used during evaluation.

Details

- `faers_phv_table`: build a contingency table for all events in `.events`.
- `faers_phv_signal`: Pharmacovigilance Analysis used contingency table constructed with `faers_phv_table`. You must pass `.full` or `.object2` into `faers_phv_table`.

Value

A data.table object.

See Also

phv_signal

Examples

```
# you must change `dir`, as the files included in the package are sampled
data <- faers(c(2004, 2017), c("q1", "q2"),
    dir = system.file("extdata", package = "faers"),
    compress_dir = tempdir()
)
## Not run:
# you must standardize and deduplication before disproportionality analysis
# you should replace `meddra_path` with yours
data <- faers_standardize(data, meddra_path)
data <- faers_dedup(data)
# we use faers_filter() to extract data we are interested
# here, we just sample 100 reports. You should do it based on your purpose.
faers_phv_signal(
    faers_filter(data, .fn = ~ sample(faers_primaryid(.x), 100L)),
    .full = data
)

## End(Not run)
```

---

faers_standardize *Standardize FAERS Quarterly Data for Preferred Term and drug names*

Description

Standardize FAERS Quarterly Data for Preferred Term and drug names

Usage

```
faers_standardize(object, ...)

## S4 method for signature 'FAERSascii'
faers_standardize(object, meddra_path, add_smq = FALSE)
```

Arguments

object	A FAERSascii object.
...	Other arguments passed to specific methods.
meddra_path	A string, define the path of MedDRA directory.
add_smq	A bool, indicates whether Standardised MedDRA Queries (SMQ) should be added. If TRUE, "smq_content.asc", and "smq_list.asc" must exist.

Value

A FAERSascii object.

See Also

MedDRA

Examples

```
#' # you must change `dir`, as the files included in the package are sampled
data <- faers(c(2004, 2017), c("q1", "q2"),
    dir = system.file("extdata", package = "faers"),
    compress_dir = tempdir()
)
## Not run:
# you should replace `meddra_path` with yours
data <- faers_standardize(data, meddra_path)

## End(Not run)
```

---

fda_drugs *Read and Parse Drugs@FDA data*

Description

Read and Parse Drugs@FDA data

Usage

```
fda_drugs(pattern = "Products", url = NULL, list = FALSE, force = FALSE)
```

Arguments

pattern	File pattern to use. Must define a file exactly, you can set list = TRUE to see what files can be used.
url	A string of the url for Drugs@FDA file. Try to get the link from site: https://www.fda.gov/drugs/drug-approvals-and-databases/drugsfda-data-files.
list	A boolean value, should it only list files in the Drugs@FDA dataset?
force	A boolean value. If set to TRUE, it indicates the retrieval of Drugs@FDA data in the FDA directly, bypassing the cache.

Value

- if list = TRUE, an atomic character.
- if list = FALSE, a data.table.

Examples

```
fda_drugs(list = TRUE)
fda_drugs()
```

MedDRA-class *MedDRA class*

Description

Provide a container for MedDRA Data file

Usage

```
meddra(path, add_smq = FALSE, primary_soc = FALSE)

## S4 method for signature 'MedDRA'
show(object)

meddra_hierarchy(object, ...)

## S4 method for signature 'MedDRA'
meddra_hierarchy(object)

meddra_smq(object, ...)

## S4 method for signature 'MedDRA'
meddra_smq(object)

meddra_version(object, ...)

## S4 method for signature 'MedDRA'
meddra_version(object)
```

Arguments

path	A string, define the path of MedDRA directory.
add_smq	A bool, indicates whether Standardised MedDRA Queries (SMQ) should be added. If TRUE, "smq_content.asc", and "smq_list.asc" must exist.
primary_soc	A bool, indicates whether keep primary soc only.
object	A MedDRA object.
...	Other arguments passed to specific methods.

Value

- meddra: A MedDRA object.
- meddra_hierarchy: Extract the hierarchy slot.
- meddra_smq: Extract the smq slot.
- meddra_version: Extract the version slot.

Slots

hierarchy A data.table or NULL representing the meddra hierarchy data. There are five levels to the MedDRA hierarchy, arranged from very specific to very general.

smq A data.table or NULL representing the meddra smq data. Standardised MedDRA Queries (SMQs) are used to support signal detection and monitoring. SMQs are validated, standard sets of MedDRA terms. These sets of terms have undergone extensive review, testing, analysis and expert discussion. SMQs represent a variety of safety topics of regulatory interest (e.g., SMQ Severe cutaneous adverse reactions, SMQ Anaphylactic reaction).

version A string, the version of MedDRA.

See Also

- https://www.meddra.org/
- https://www.meddra.org/how-to-use/basics/hierarchy
- https://www.meddra.org/how-to-use/tools/smqs

---

phv_signal *Pharmacovigilance Analysis*

Description

Pharmacovigilance, also known as drug safety. In the context of pharmacovigilance studies, disproportionality analysis primarily served as a tool to evaluate possible association between a specific adverse event and a particular drug which can then be investigated through clinical assessment of individual case reports.

Usage

```
phv_signal(
  a,
  b,
  c,
  d,
  methods = NULL,
  alpha = 0.05,
  correct = TRUE,
  n_mcmc = 100000L,
  alpha1 = 0.5,
  alpha2 = 0.5,
  theta_init = NULL,
  squashing = TRUE,
  BPPARAM = SerialParam()
)

phv_ror(a, b, c, d, alpha = 0.05)

phv_prr(a, b, c, d, alpha = 0.05)

phv_chisq(a, b, c, d, correct = TRUE, BPPARAM = SerialParam())

phv_fisher(a, b, c, d, alpha = 0.05, BPPARAM = SerialParam())

phv_bcpnn_norm(a, b, c, d, alpha = 0.05)

phv_bcpnn_mcmc(
  a,
  b,
  c,
  d,
  alpha = 0.05,
  n_mcmc = 100000L,
  BPPARAM = SerialParam()
)

phv_obsexp_shrink(
  a,
  b,
  c,
  d,
  alpha = 0.05,
  alpha1 = 0.5,
  alpha2 = 0.5,
  n_mcmc = 100000L,
  BPPARAM = SerialParam()
)
```

```
phv_ebgm(a, b, c, d, alpha = 0.05, theta_init = NULL, squashing = TRUE)
```

Arguments

a	also referred to as n11 as this is the count of event of interest under exposure of interest.
b	also referred to as n10 as this is the count of *not* event of interest under exposure of interest.
c	also referred to as n01 as this is the count of event of interest under *not* exposure of interest.
d	also referred to as n00 as this is the count of *not* event of interest under *not* exposure of interest.
methods	An atomic character, specifies the method used to signal mining. Currently, only "ror", "prr", "chisq", "bcpnn_norm", "bcpnn_mcmc", "obsexp_shrink", "fisher", and "ebgm" are supported. If NULL, all supported methods will be used.
alpha	Level of significance, for construction of the confidence intervals.
correct	A bool indicating whether to apply Yates's continuity correction when computing the chi-squared statistic.
n_mcmc	Number of MCMC simulations per (a,b,c,d)-tuple to calculate confidence intervals.
alpha1	Numerator shrinkage parameter >=0, default 0.5.
alpha2	Denominator shrinkage parameter >=0, default 0.5.
theta_init	A data frame of initial hyperparameter guesses with columns ordered as: alpha1, beta1, alpha2, beta See openEBGM::autoHyper
squashing	A bool, whether do automated data squashing. If any zeros found in a, will always be TRUE.
BPPARAM	An optional BiocParallelParam instance defining the parallel back-end to be used during evaluation.

Details

Note that the a, b, c, d inputs can be an atomic vectors of equal length, for which the function will perform the calculations for each individual (a,b,c,d)-tuple moving across the vectors.

It is assumed that the contingency table under consideration has drugs/exposures in the rows and outcomes/events in the columns. See contingency table section.

We use the distinct patient count method to obtain the frequency counts of patients exposed to each interested drug, those reporting interested event. As illustrated in the Contingency table, n equals the total number of patients in the database, n11 is the number of patients with exposure to the interested drug during the model period and reporting interested events, n10 is the number of patients that have used the interested drug but did not experience interested event during any of the model periods associated with the drug, n01 is the number of patients that did not use the interested drug but experienced interested event, and n00 is the number of patients that were not exposed to the interested drug and did not report interested condition.

Value

A data.table with columns of estimated value and it's confidence interval (`ci_low` and `ci_high`). Estimated column are as follows:

- `phv_ror`: reporting odds ratio (`ror`).
- `phv_prr`: proportional reporting ratio (`prr`). Signal defined as a `prr` of at least 2, chi-squared with Yates's correction of at least 4 and `a >= 3`. An equivalent alternative to chi-squared is to calculate a confidence interval around the `prr`.
- `phv_bcpnn_norm`: information component (`ic`).
- `phv_bcpnn_mcmc`: information component (`ic`).
- `phv_obsexp_shrink`: observed to expected ratio (`oe_ratio`).
- `phv_ebgm`: Empirical Bayes Geometric Mean (`ebgm`).

Contingency table

	ADR of interest	Other ADRs	Total
Drug of interest	a=n11	b=n10	a+b=n1.
Other drugs	c=n01	d=n00	c+d=n0.
Total	a+c=n.1	b+d=n.0	a+b+c+d=n

phv_obsexp_shrink

The observed to expected (OE) ratio with approximate confidence intervals are constructed on the log2 scale as outlined in Norén et al. (2013).

Expected value was estimated by `(a + b) / (a + b + c + d) * (a + c)`.

The OE ratio with shrinkage estimates is calculated as `(O + alpha1) / (E + alpha2)`.

If `(O + alpha1) < 1`, then the exact uncertainty limits should be used. That is the confidence intervals as implemented in `phv_bcpnn_mcmc` (Norén et al., 2013).

`log2(OE)` approximates the Bayesian confidence propagation neural network information component (IC) with reasonable accuracy when `alpha1 = alpha2 = 0.5` (Norén et al., 2013).

phv_ebgm

An implementation of the Gamma-Poisson Shrinker (GPS) model for identifying unexpected counts in large contingency tables using an empirical Bayes approach. The Empirical Bayes Geometric Mean (EBGM) and quantile scores are obtained from the GPS model estimates. The GPS was proposed by DuMouchel as a signal detection tool for large frequency tables with both observed (O) and expected (E) counts for each drug-outcome pair. It assumes the observed count of any drug-outcome pair follows the Poisson distribution.

For each drug-outcome pair, the primary parameter of interest was the risk ratio. Rather than using the observed over expected (O/E), GPS uses the empirical Bayesian geometric mean (EBGM) posterior distribution of the risk ratio and the surrounding confidence interval for each drug-outcome

pair to identify statistical signals of excess risk. To prevent spurious false positives due to implausibly high risk ratios, GPS implements a Bayesian framework that "shrinks" O/E estimates towards a value which is close to the average O/E values for all drug-event pairs at each level of granularity.

References

- David Olaleye, SAS Institute Inc. (2019), Real-World Evidence and Population Health Analytics: Intersection and Application, https://support.sas.com/resources/papers/proceedings19/3361-2019.pdf
- Evans, S.J.W., Waller, P.C. and Davis, S. (2001), Use of proportional reporting ratios (PRRs) for signal generation from spontaneous adverse drug reaction reports. Pharmacoepidem. Drug Safe., 10: 483-486. https://doi.org/10.1002/pds.677
- Norén GN, Hopstadius J, Bate A. Shrinkage observed-to-expected ratios for robust and transparent large-scale pattern discovery. Statistical methods in medical research. 2013 Feb;22(1):57-69.
- https://journal.r-project.org/archive/2017/RJ-2017-063/RJ-2017-063.pdf

Examples

```
phv_signal(122, 1320, 381, 31341)
phv_signal(122, 1320, 381, 31341, "ror")
phv_ror(122, 1320, 381, 31341)
phv_signal(122, 1320, 381, 31341, "prr")
phv_prr(122, 1320, 381, 31341)
phv_signal(122, 1320, 381, 31341, "chisq")
phv_chisq(122, 1320, 381, 31341)
phv_signal(122, 1320, 381, 31341, "bcpnn_norm")
phv_bcpnn_norm(122, 1320, 381, 31341)
phv_signal(122, 1320, 381, 31341, "bcpnn_mcmc")
phv_bcpnn_mcmc(122, 1320, 381, 31341)
phv_signal(122, 1320, 381, 31341, "obsexp_shrink")
phv_obsexp_shrink(122, 1320, 381, 31341)
phv_signal(122, 1320, 381, 31341, "fisher")
phv_fisher(122, 1320, 381, 31341)
phv_signal(122, 1320, 381, 31341, "ebgm")
phv_ebgm(122, 1320, 381, 31341)
```

Index

∗ **internal**
faers-package, 3
[,FAERSascii,ANY,ANY,ANY-method *(faers_get)*, 13
[,FAERSascii-method *(faers_get)*, 13
[[,FAERSascii-method *(faers_get)*, 13
$,FAERSascii-method *(faers_get)*, 13

athena, 3

BiocParallelParam, *21*, *26*

copy, *10*
countrycode(), *19*

data.table, *4*, *6*, *10, 11*, *14–18*, *21*, *23, 24*, *27*
deletedCases, *11*

FAERS, *9*, *15*
FAERS *(FAERS-class)*, 5
faers, 4, *15*
FAERS-class, 5
faers-package, 3
faers_available, 7
faers_before_period, 8
faers_clearcache, 8
faers_combine, 9
faers_counts, 10, *20*
faers_counts,FAERSascii-method *(faers_counts)*, 10
faers_data *(FAERS-class)*, 5
faers_data,FAERS-method *(FAERS-class)*, 5
faers_dedup, *6*, *9*, 11
faers_dedup,ANY-method *(faers_dedup)*, 11
faers_dedup,FAERSascii-method *(faers_dedup)*, 11
faers_dedup,FAERSxml-method *(faers_dedup)*, 11
faers_deleted_cases *(FAERS-class)*, 5
faers_deleted_cases,FAERSascii-method *(FAERS-class)*, 5
faers_download, 12
faers_field *(FAERS-class)*, 5
faers_filter *(faers_get)*, 13
faers_filter,FAERSascii-method *(faers_get)*, 13
faers_get, 13
faers_get,FAERSascii-method *(faers_get)*, 13
faers_header *(FAERS-class)*, 5
faers_header,FAERSxml-method *(FAERS-class)*, 5
faers_keep *(faers_get)*, 13
faers_keep,FAERSascii-method *(faers_get)*, 13
faers_load, 15
faers_meddra *(FAERS-class)*, 5
faers_meddra,FAERS-method *(FAERS-class)*, 5
faers_merge, 16
faers_merge,ANY-method *(faers_merge)*, 16
faers_merge,FAERSascii-method *(faers_merge)*, 16
faers_merge,FAERSxml-method *(faers_merge)*, 16
faers_meta, 18
faers_mget *(faers_get)*, 13
faers_mget,FAERSascii-method *(faers_get)*, 13
faers_modify *(faers_get)*, 13
faers_modify,FAERSascii-method *(faers_get)*, 13
faers_parse, 18
faers_period, *9*
faers_period *(FAERS-class)*, 5
faers_period,FAERS-method *(FAERS-class)*, 5
faers_phv_signal, 20
faers_phv_signal,FAERSascii-method *(faers_phv_signal)*, 20

faers_phv_table *(faers_phv_signal)*, 20
faers_phv_table,FAERSascii,FAERSascii,missing-method *(faers_phv_signal)*, 20
faers_phv_table,FAERSascii,missing,FAERSascii-method *(faers_phv_signal)*, 20
faers_primaryid *(faers_get)*, 13
faers_primaryid,FAERSascii-method *(faers_get)*, 13
faers_quarter *(FAERS-class)*, 5
faers_quarter,FAERS-method *(FAERS-class)*, 5
faers_standardize, *9*, *12*, 21
faers_standardize,FAERSascii-method *(faers_standardize)*, 21
faers_year *(FAERS-class)*, 5
faers_year,FAERS-method *(FAERS-class)*, 5
FAERSascii, *4*, *9–14*, *17*, *19*, *20*, *22*
FAERSascii *(FAERS-class)*, 5
FAERSascii-class *(FAERS-class)*, 5
FAERSxml, *4*, *9*, *17*, *19*
FAERSxml *(FAERS-class)*, 5
FAERSxml-class *(FAERS-class)*, 5
fda_drugs, 22

MedDRA, *6*, *22*
MedDRA *(MedDRA-class)*, 23
meddra *(MedDRA-class)*, 23
MedDRA-class, 23
meddra_hierarchy *(MedDRA-class)*, 23
meddra_hierarchy,MedDRA-method *(MedDRA-class)*, 23
meddra_smq *(MedDRA-class)*, 23
meddra_smq,MedDRA-method *(MedDRA-class)*, 23
meddra_version *(MedDRA-class)*, 23
meddra_version,MedDRA-method *(MedDRA-class)*, 23
multi_download, *12*

new_handle, *4*

openEBGM::autoHyper, *26*

phv_bcpnn_mcmc *(phv_signal)*, 24
phv_bcpnn_norm *(phv_signal)*, 24
phv_chisq *(phv_signal)*, 24
phv_ebgm *(phv_signal)*, 24
phv_fisher *(phv_signal)*, 24
phv_obsexp_shrink *(phv_signal)*, 24
phv_prr *(phv_signal)*, 24
phv_ror *(phv_signal)*, 24
phv_signal, *20*, *21*, 24
show,FAERS-method *(FAERS-class)*, 5
show,FAERSascii-method *(FAERS-class)*, 5
show,MedDRA-method *(MedDRA-class)*, 23