Package ‘CohortSymmetry’

January 20, 2025

Title Sequence Symmetry Analysis Using the Observational Medical Outcomes Partnership Common Data Model

Version 0.2.0

Maintainer Xihang Chen <`xihang.chen@ndorms.ox.ac.uk`>

Description Calculating crude sequence ratio, adjusted sequence ratio and confidence intervals using data mapped to the Observational Medical Outcomes Partnership Common Data Model.

License Apache License (>= 2)

Encoding UTF-8

RoxygenNote 7.3.2

Suggests testthat (>= 3.1.5), DBI (>= 1.0.0), cli, checkmate, odbc, RPostgres, tidyselect, knitr, dbplyr (>= 2.5.0), flextable, gt, ggplot2, rmarkdown

Imports CDMConnector (>= 1.3.0), dplyr, PatientProfiles, rlang, tibble, visOmopResults (>= 0.4.0), tidyr, omock (>= 0.3.1), stats, duckdb, here, omopgenerics (>= 0.4.0), DrugUtilisation (>= 0.7.0), CodelistGenerator (>= 3.1.0)

Config/testthat/edition 3

URL https://ohdsi.github.io/CohortSymmetry/

BugReports https://github.com/OHDSI/CohortSymmetry/issues

VignetteBuilder knitr

NeedsCompilation no

Author Xihang Chen [aut, cre] (<https://orcid.org/0009-0001-8112-8959>),
Tyman Stanford [aut] (<https://orcid.org/0000-0002-8570-5493>),
Berta Raventós [aut] (<https://orcid.org/0000-0002-4668-2970>),
Nicole Pratt [aut] (<https://orcid.org/0000-0001-8730-8910>),
Ed Burn [aut] (<https://orcid.org/0000-0002-9286-1128>),
Marti Català [aut] (<https://orcid.org/0000-0003-3308-9905>),
Danielle Newby [aut] (<https://orcid.org/0000-0002-3001-1478>),
Núria Mercadé-Besora [aut] (<https://orcid.org/0009-0006-7948-3747>),
Mike Du [aut] (<https://orcid.org/0000-0002-9517-8834>),

Yuchen Guo [aut] (<https://orcid.org/0000-0002-0847-4855>),
Kim Lopez [aut] (<https://orcid.org/0000-0002-8462-8668>),
Marta Alcalde-Herraiz [aut] (<https://orcid.org/0009-0002-4405-1814>)

Repository CRAN

Date/Publication 2024-11-29 12:50:02 UTC

Contents

---

generateSequenceCohortSet

Intersecting the index and marker cohorts prior to calculating Sequence Symmetry Ratios

---

Description

Join two tables in the CDM (one for index and the other for marker cohorts) into a new table in the cdm taking into account the maximum time interval between events. Index and marker cohorts should be instantiated in advance by the user.

Usage

```
generateSequenceCohortSet(
  cdm,
  indexTable,
  markerTable,
  name,
  indexId = NULL,
  markerId = NULL,
  cohortDateRange = as.Date(c(NA, NA)),
  daysPriorObservation = 0,
  washoutWindow = 0,
  indexMarkerGap = Inf,
  combinationWindow = c(0, 365),
  movingAverageRestriction = 548
)
```

Arguments

`cdm`	A CDM reference.
`indexTable`	A table in the CDM that the index cohorts should come from.
`markerTable`	A table in the CDM that the marker cohorts should come from.
`name`	The name within the cdm that the output is called. Default is joined_cohorts.
`indexId`	Cohort definition IDs in indexTable to be considered for the analysis. Change to NULL if all indices are wished to be included.
`markerId`	Cohort definition IDs in markerTable to be considered for the analysis. Change to NULL if all markers are wished to be included.
`cohortDateRange`	Two dates indicating study period and the sequences that the user wants to restrict to.
`daysPriorObservation`	The minimum amount of prior observation required on both the index and marker cohorts per person.
`washoutWindow`	A washout window to be applied on both the index cohort event and marker cohort.
`indexMarkerGap`	The maximum allowable gap between the end of the first episode and the start of the second episode in a sequence/combination.
`combinationWindow`	A constrain to be placed on the gap between two initiations. Default c(0,365), meaning the gap should be larger than 0 but less than or equal to 365.
`movingAverageRestriction`	The moving window when calculating nSR, default is 548.

Value

A table within the cdm reference.

Examples

```
library(CohortSymmetry)
cdm <- mockCohortSymmetry()
cdm <- generateSequenceCohortSet(
  cdm = cdm,
  name = "joined_cohorts",
  indexTable = "cohort_1",
  markerTable = "cohort_2"
)
 cdm$joined_cohorts
 CDMConnector::cdmDisconnect(cdm = cdm)
```

`mockCohortSymmetry` *Creates mock cdm object for testing*

Description

Creates a mock cdm with two default synthetic cohorts, one is the index cohort and the other one is the marker cohort. However the users could specify them should they wish.

Usage

```
mockCohortSymmetry(
  seed = 1,
  indexCohort = NULL,
  markerCohort = NULL,
  con = DBI::dbConnect(duckdb::duckdb(), ":memory:"),
  schema = "main"
)
```

Arguments

`seed`	The seed to be inputted.
`indexCohort`	The tibble of your index cohort. Default is NULL, which means the default indexCohort is being used.
`markerCohort`	The tibble of your marker cohort. Default is NULL, which means the default markerCohort is being used.
`con`	Connection detail.
`schema`	Name of your write schema.

Value

A mock cdm object contains your index and marker cohort

Examples

```
library(CohortSymmetry)
cdm <- mockCohortSymmetry()
cdm
CDMConnector::cdmDisconnect(cdm = cdm)
```

`plotSequenceRatios` *A plot for the sequence ratios.*

Description

It provides a ggplot of the sequence ratios of index and marker cohorts.

Usage

```
plotSequenceRatios(
  result,
  onlyASR = FALSE,
  plotTitle = NULL,
  labs = c("SR", "Drug Pairs"),
  colours = c("red", "blue"),
  facet = NULL
)
```

Arguments

`result`	Table output from summariseSequenceRatios.
`onlyASR`	If set to be TRUE then only adjusted SR will be plotted. Otherwise if it is set to be FALSE then both adjusted and crude SR will be plotted.
`plotTitle`	Title of the plot, if NULL no title will be included in the plot.
`labs`	Axis labels for the plot.
`colours`	Colours for sequence ratio.
`facet`	The variable to facet by.

Value

A plot for the sequence ratios of index and marker cohorts.

Examples

```
library(CohortSymmetry)
cdm <- mockCohortSymmetry()
cdm <- generateSequenceCohortSet(cdm = cdm,
                                 indexTable = "cohort_1",
                                 markerTable = "cohort_2",
                                 name = "joined_cohort")
sequence_ratio <- summariseSequenceRatios(cohort = cdm$joined_cohort,
                                          minCellCount = 0)
plotSequenceRatios(result = sequence_ratio)
CDMConnector::cdmDisconnect(cdm = cdm)
```

`plotTemporalSymmetry` *A plot for the temporal symmetry of cohorts.*

Description

It provides a ggplot of the temporal symmetry of two or more cohorts.

Usage

```
plotTemporalSymmetry(
  result,
  plotTitle = NULL,
  labs = c("Time (months)", "Individuals (N)"),
  xlim = c(-12, 12),
  colours = c("blue", "red"),
  scales = "free"
)
```

Arguments

`result`	Table output from summariseTemporalSymmetry.
`plotTitle`	Title of the plot, if NULL no title will be plotted.
`labs`	Axis labels for the plot.
`xlim`	Limits for the x axis of the plot.
`colours`	Colours for both parts of the plot, pre- and post- time 0.
`scales`	Whether to set free y scales for the facet wrap when there are multiple plots (i.e. each plot has its own scaled y axis) or set them equal for all. Only accepts "free" for the former and "fixed" for the latter.

Value

A plot for the temporal symmetry of cohorts.

Examples

```
library(CohortSymmetry)
cdm <- mockCohortSymmetry()
cdm <- generateSequenceCohortSet(cdm = cdm,
                                 indexTable = "cohort_1",
                                 markerTable = "cohort_2",
                                 name = "joined_cohort")
temporal_symmetry <- summariseTemporalSymmetry(cohort = cdm$joined_cohort,
                                               minCellCount = 0)
plotTemporalSymmetry(result = temporal_symmetry)
CDMConnector::cdmDisconnect(cdm = cdm)
```

`summariseSequenceRatios` *Sequence ratio calculations*

Description

Using generateSequenceCohortSet to obtain sequence ratios for the desired outcomes.

Usage

```
summariseSequenceRatios(
  cohort,
  cohortId = NULL,
  confidenceInterval = 95,
  minCellCount = 5
)
```

Arguments

`cohort`	A cohort table in the cdm.
`cohortId`	The Ids in the cohort that are to be included in the analyses.
`confidenceInterval`	Default is 95, indicating the central 95% confidence interval.
`minCellCount`	The minimum number of events to reported, below which results will be obscured. If 0, all results will be reported.

Value

A local table with all the analyses.

Examples

```
library(CohortSymmetry)
cdm <- mockCohortSymmetry()
cdm <- generateSequenceCohortSet(cdm = cdm,
                                 name = "joined_cohorts",
                                 indexTable = "cohort_1",
                                 markerTable = "cohort_2")
pssa_result <- summariseSequenceRatios(cohort = cdm$joined_cohorts, minCellCount = 0)
pssa_result
CDMConnector::cdmDisconnect(cdm)
```

---

summariseTemporalSymmetry

Summarise temporal symmetry

Description

Using generateSequenceCohortSet to obtain temporal symmetry (aggregated counts) of two cohorts.

Usage

```
summariseTemporalSymmetry(
  cohort,
  cohortId = NULL,
  timescale = "month",
  minCellCount = 5
)
```

Arguments

`cohort`	A cohort table in the cdm.
`cohortId`	The Ids in the cohort that are to be included in the analyses.
`timescale`	Timescale for the x axis of the plot (month, day, year).
`minCellCount`	The minimum number of events to reported, below which results will be obscured. If 0, all results will be reported.

Value

An aggregated table with difference in time (marker - index) and the relevant counts.

Examples

```
library(CohortSymmetry)
cdm <- mockCohortSymmetry()
cdm <- generateSequenceCohortSet(cdm = cdm,
                                 name = "joined_cohorts",
                                 indexTable = "cohort_1",
                                 markerTable = "cohort_2")
temporal_symmetry <- summariseTemporalSymmetry(cohort = cdm$joined_cohorts, minCellCount = 0)
CDMConnector::cdmDisconnect(cdm)
```

`tableSequenceRatios` *A formatted visualization of sequence_symmetry objects.*

Description

It provides a formatted table with the contents of the summariseSequenceRatios output.

Usage

```
tableSequenceRatios(
  result,
  header = "marker_cohort_name",
  groupColumn = "cdm_name",
  type = "gt",
  hide = "variable_level"
)
```

Arguments

`result`	A sequence_symmetry object.
`header`	A vector specifying the elements to include in the header. See visOmopResults package for more information on how to use this parameter.
`groupColumn`	Columns to use as group labels. See visOmopResults package for more information on how to use this parameter.
`type`	The desired format of the output table.
`hide`	Columns to drop from the output table.

Value

A formatted version of the sequence_symmetry object.

Examples

```
library(CohortSymmetry)
cdm <- mockCohortSymmetry()
cdm <- generateSequenceCohortSet(cdm = cdm,
                                 indexTable = "cohort_1",
                                 markerTable = "cohort_2",
                                 name = "joined_cohort")
res <- summariseSequenceRatios(cohort = cdm$joined_cohort)
gtResult <- tableSequenceRatios(res)
CDMConnector::cdmDisconnect(cdm = cdm)
```

`tableTemporalSymmetry` *A formatted visualization of temporal_symmetry objects.*

Description

It provides a formatted table with the contents of the summariseTemporalSymmetry output.

Usage

```
tableTemporalSymmetry(
  result,
  header = "variable_level",
  groupColumn = c("cdm_name", "index_name"),
  type = "gt",
  hide = "variable_name"
)
```

Arguments

`result`	A temporal_symmetry object.
`header`	A vector specifying the elements to include in the header. See visOmopResults package for more information on how to use this parameter.
`groupColumn`	Columns to use as group labels. See visOmopResults package for more information on how to use this parameter.
`type`	The desired format of the output table.
`hide`	Columns to drop from the output table.

Value

A formatted version of the temporal_symmetry object.

Examples

```
library(CohortSymmetry)
cdm <- mockCohortSymmetry()
cdm <- generateSequenceCohortSet(cdm = cdm,
                                 indexTable = "cohort_1",
                                 markerTable = "cohort_2",
                                 name = "joined_cohort")
res <- summariseTemporalSymmetry(cohort = cdm$joined_cohort, minCellCount = 0)
gtResult <- tableTemporalSymmetry(result = res)
CDMConnector::cdmDisconnect(cdm = cdm)
```

Index