Package ‘MSCA’

June 13, 2025

Title Unsupervised Clustering of Multiple Censored Time-to-Event Endpoints

Version 1.2.1

Description Provides basic tools and wrapper functions for computing clusters of instances described by multiple time-to-event censored endpoints. From long-format datasets, where one instance is described by one or more dated records, the main function, `` `make_state_matrices()` ``, creates state matrices. Based on these matrices, optimised procedures using the Jaccard distance between instances enable the construction of longitudinal typologies. The package is under active development, with additional tools for graphical representation of typologies planned. For methodological details, see our accompanying paper: `Delord M, Douiri A (2025) <doi:10.1186/s12874-025-02476-7>`.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.2

LinkingTo Rcpp, RcppArmadillo, RcppParallel,

SystemRequirements GNU make

Imports Rcpp, fastkmedoids, RcppParallel (>= 5.1.10), data.table, dplyr, Matrix, rlang

Depends R (>= 3.5)

LazyData true

Suggests knitr, rmarkdown, cluster, fastcluster

VignetteBuilder knitr

NeedsCompilation yes

Author Marc Delord [aut, cre] (ORCID: <https://orcid.org/0000-0002-0455-6749>)

Maintainer Marc Delord <mdelord@gmail.com>

Repository CRAN

Date/Publication 2025-06-13 21:30:02 UTC

Contents

EHR *Description of the EHR dataset*

Description

This is a toy dataset to illustrate the use of the the MSCA library

Format

A data frame with `3000` records and `3` variables:

link_id Unique patient identifyer

reg Registered long-term condition

aos Age at onset of the registered long-term condition

Source

Toy dataset

Examples

```
# Load the dataset
data(EHR)

# Display the first few rows
head(EHR)
```

fast_clara_jaccard *Fast CLARA-like clustering using Jaccard dissimilarity*

Description

Implements a CLARA (Clustering Large Applications) strategy using Jaccard dissimilarity computed on individual patients state matrices. The algorithm repeatedly samples subsets of the data, performs PAM clustering on each subset, and selects the medoids that minimise the total dissimilarity across the full dataset. Final assignments are made by mapping all data points to the nearest selected medoid.

Usage

```
fast_clara_jaccard(
  data,
  k,
  samples = 20,
  samplesize = NULL,
  seed = 123,
  frac = 1
)
```

Arguments

`data`	A state matrix of censored time-to-event indicators as computed by the `make_state_matrix` function.
`k`	Number of returned clusters.
`samples`	Number of random samples drawn from the analysed population.
`samplesize`	Number of patients per sample (default: min(50 + 5k, ncol(data))).
`seed`	Random seed for reproducibility (default: 123).
`frac`	Fraction of the population to use for cost computation (default: 1).

Details

This implementation adapts the original CLARA method described by Kaufman and Rousseeuw (1990) in "Finding Groups in Data: An Introduction to Cluster Analysis".

Value

A list with index of patients from the sample a, medoid indices, cluster assignment, and cost.

clustering An integer vector of cluster assignments for each patient.

medoids Indices of medoids associated witht the lower cost.

sample Indices of the sampled columns used in clustering.

cost Total cost (sum of dissimilarities to assigned medoids).

Note

To improve efficiency, the function used fastpam procedure from the fastkmedoids library and uses optimized Jaccard index computation. For simulation purpose, the `frac` parameter can be used to reduce time when computing the cost for each sample. The final cost is given using medoids associated with lower cost computed on fractionned data. A final analysis using the proper CLARA method should be conducted setting `frac` to 1.

References

Kaufman, L. & Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley.

`fast_jaccard_dist` *Compute upper triangle Jaccard distance*

Description

Compute upper triangle Jaccard distance

Usage

```
fast_jaccard_dist(mat, as.dist = FALSE)
```

Arguments

`mat`	A numeric binary matrix (0/1/NA)
`as.dist`	Logical. If TRUE, return a dist object; otherwise an upper triangular matrix.

Value

A matrix or dist object of Jaccard dissimilarities (upper triangle)

`get_cluster_sequences` *Extract sequences of length k within clusters*

Description

For each cluster, extract all sequence of length `k` from the ordered observations grouped by individual IDs. Returns a list of sequences per cluster.

Usage

```
get_cluster_sequences(
  dt,
  cl_col = "cl",
  id_col = "link_id",
  event_col = "reg",
  aos_col = "aos",
  cens = "cens",
  k = 2
)
```

Arguments

`dt` A `data.table` or data.frame containing the data in a long format.

`cl_col` Name of the column containing cluster labels.

`id_col` Name of the column identifying individual trajectories (e.g. patient ID).

`event_col` Name of the column containing ordered events (e.g. diagnoses, prescriptions).

`aos_col` Name of the column containing age at onset.

`cens` Code indicating censoring.

`k` Integer specifying the sequence length (recomended 2).

Value

A named list of data frames, each containing sequences of length `k` observed in a given cluster.

Author(s)

Marc Delord

References

Delord M, Douiri A (2025) `doi:10.1186/s12874-025-02476-7`

See Also

`cspade` in the **arulesSequences** package for sequential pattern mining using the SPADE algorithm.

make_state_matrices *Construct state matrices from longitudinal EHR Data*

Description

Builds a binary matrix (0/1/NA) encoding whether each individual had each long-term condition (LTC) at each time point from 0 to `l`, based on their age of onset. The matrix includes all LTCs, including those used to determine censoring and failure. However, the presence of `fail_code` or `cens_code` still triggers NA values after their onset.

Usage

```
make_state_matrices(
  data,
  id = "link_id",
  ltc = "reg",
  aos = "aos",
  l = 111,
  fail_code = "death",
  cens_code = "cens"
)
```

Arguments

`data`	A data frame containing one row per condition occurrence.
`id`	Name of the column identifying individuals.
`ltc`	Name of the column containing LTC labels.
`aos`	Name of the column giving age of onset (or time of onset) for each LTC.
`l`	The maximum time index (inclusive); matrix has `l + 1` time rows per LTC.
`fail_code`	Label in `ltc` indicating a failure event (e.g., death).
`cens_code`	Label in `ltc` indicating censoring.

Value

A matrix with `(l + 1) * number of LTCs` rows and one column per unique individual. Values are 1 after onset, 0 before, and NA after censor/fail. Rows are named `<ltc>_<time>`, and columns are individual IDs.

Note

For large datasets, computations may be split into multiple jobs to manage memory and performance. Consider reducing the time granularity and/or the number of long-term condition (event of interest) to improve efficiency and stability.

Author(s)

@author Marc Delord

References

Delord M, Douiri A (2025) doi:10.1186/s12874-025-02476-7

`sequence_stats` *Compute sequence statistics*

Description

Computes descriptive statistics for sequences, including sequence frequency for any sequence length, and conditional probability and relative risk for sequences of length 2 (pairwise transitions).

Usage

```
sequence_stats(
  seq_data,
  min_seq_freq = 0.01,
  min_conditional_prob = 0,
  min_relative_risk = 0,
  forward = TRUE
)
```

Arguments

`seq_data` A list of data frames containing sequences, must be the output of `get_cluster_sequences`.

`min_seq_freq` Numeric threshold (default = 0.01). Filters out sequences with relative frequency below this value.

`min_conditional_prob` Numeric threshold (default = 0). Applies only for pairwise sequences (`k = 2`).

`min_relative_risk` Numeric threshold (default = 0). Applies only for pairwise sequences (`k = 2`).

`forward` If `TRUE` only sequences with median age at onset of `from` is lower than median age at onset of `to` are kept

Details

For `k = 2`, the function computes:

- **seq_freq:** Proportion of all sequences that match the pair
- **conditional_prob:** P(to | from)
- **relative_risk:** conditional probability divided by the marginal probability of `to`

For `k > 2`, only `seq_freq` is computed.

Value

A list of data frames, each containing the sequence statistics for one cluster.

See Also

`get_cluster_sequences`

Index