Package 'MUGS'

May 19, 2025

Type Package

Title Multisource Graph Synthesis with EHR Data

Version 0.1.0

Description We develop Multi-source Graph Synthesis (MUGS), an algorithm designed to create embeddings for pediatric Electronic Health Record (EHR) codes by leveraging graphical information from three distinct sources:
(1) pediatric EHR data, (2) EHR data from the general patient population, and
(3) existing hierarchical medical ontology knowledge shared across different patient populations.
See Li et al. (2024) <doi:10.1038/s41746-024-01320-4> for details.

License GPL-3

Encoding UTF-8

LazyData true

LazyDataCompression xz

RoxygenNote 7.3.2

URL https://github.com/celehs/MUGS, https://celehs.github.io/MUGS/, https://doi.org/10.1038/s41746-024-01320-4

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Imports MASS, Matrix, fastDummies, doSNOW, dplyr, grplasso, foreach, glmnet, grpreg, inline, mvtnorm, pROC, parallel, RcppArmadillo, rsvd, methods

Depends R (>= 3.5.0)

Config/testthat/edition 3

NeedsCompilation no

Author Mengyan Li [cre, aut],
Thomas Charlon [ctb] (ORCID: 0000-0001-7497-0470),
Xiaoou Li [aut],
Tianxi Cai [aut],
PARSE LTD [aut],
CELEHS Team [aut]

Maintainer Mengyan Li <mengyanli@bentley.edu>

Repository CRAN

Date/Publication 2025-05-19 13:40:09 UTC

Contents

`CodeEff_Matrix` *Function Used To Estimate Code Effects*

Description

This function estimates code effects using left and right embeddings from source and target sites.

Usage

```
CodeEff_Matrix(
  S.1,
  S.2,
  n1,
  n2,
  U.1,
  U.2,
  V.1,
  V.2,
  common_codes,
  zeta.int,
  lambda,
  p
)
```

Arguments

S.1	SPPMI from the source site.
S.2	SPPMI from the target site.
n1	The number of codes from the source site.
n2	The number of codes from the target site.
U.1	The left embeddings left singular vectors times the square root of the singular values from the source site.
U.2	The left embeddings left singular vectors times the square root of the singular values from the target site.
V.1	The right embeddings right singular vectors times the square root of the singular values from the source site.
V.2	The right embeddings right singular vectors times the square root of the singular values from the target site.
common_codes	The list of overlapping codes.
zeta.int	The initial estimator for the code effects.
lambda	The tuning parameter controls the intensity of penalization on the code effect.
p	The length of an embedding.

Value

A list with the following elements:

zeta	The estimated code effects.
dif_F	The Frobenius norm difference between the updated and initial estimators.
V.1.new	Updated right embeddings for the source site.
V.2.new	Updated right embeddings for the target site.

CodeSiteEff_l2_par *Function Used To Estimate Code-Site Effects Parallelly*

Description

Function Used To Estimate Code-Site Effects Parallelly

Usage

```
CodeSiteEff_l2_par(
  S.1,
  S.2,
  n1,
  n2,
  U.1,
  U.2,
```

```
  V.1,
  V.2,
  delta.int,
  lambda.delta,
  p,
  common_codes,
  n.common,
  n.core
)
```

Arguments

S.1	SPPMI from the source site
S.2	SPPMI from the target site
n1	the number of codes from the source site
n2	the number of codes from the target site
U.1	the left embeddings (left singular vectors times the square root of the singular values) from the source site
U.2	the left embeddings (left singular vectors times the square root of the singular values) from the target site
V.1	the right embeddings (right singular vectors times the square root of the singular values) from the source site
V.2	the right embeddings (right singular vectors times the square root of the singular values) from the target site
delta.int	the initial estimator for the code-site effect
lambda.delta	the tuning parameter controls the intensity of penalization on the code-site effects
p	the length of an embedding
common_codes	the list of overlapping codes
n.common	the number of overlapping codes
n.core	the number of cored used for parallel computation

Value

The output for the estimation of code-site effects

DataGen_rare_group *Function used to generate input data (used only for Simulations) Generate SPPMIs, dummy matrices based on prior group structures, and code-code pairs for tuning and evaluation*

Description

Function used to generate input data (used only for Simulations) Generate SPPMIs, dummy matrices based on prior group structures, and code-code pairs for tuning and evaluation

Usage

```
DataGen_rare_group(
  seed = NULL,
  p,
  n1,
  n2,
  n.common,
  n.group,
  sigma.eps.1,
  sigma.eps.2,
  ratio.delta,
  network.k,
  rho.beta,
  rho.U0,
  rho.delta,
  sigma.rare,
  n.rare,
  group.size
)
```

Arguments

seed	for reproducibility
p	the length of an embedding
n1	the number of codes in site 1
n2	the number of codes in site 2
n.common	common: the number of overlapping codes
n.group	the number of groups
sigma.eps.1	the sd of error in site 1
sigma.eps.2	the sd of error in site 2
ratio.delta	the proportion of codes in each site that have site-specific effects applied to them
network.k	the number of distinct blocks within each site for which unique inter-code correlations are modeled

rho.beta	AR parameter for the group effects covariance matrix
rho.U0	AR parameter for the code effects covariance matrix
rho.delta	AR parameter for the code-site effects covariance matrix
sigma.rare	the sd of error for rare codes (usually larger than sigma.eps.1 and sigma.eps.2)
n.rare	The number of rare codes
group.size	the size of each group

Value

Returns input data, SPPMIs, dummy matrices based on prior group structures and code-code pairs for tuning and evaluation

---

download_example_data *Download and Load Example Data from Zenodo*

Description

Download and Load Example Data from Zenodo

Usage

```
download_example_data(file, destdir = tempdir())
```

Arguments

file	Name of the .Rdata file to download (e.g., "S.1.Rdata").
destdir	Directory to store the downloaded data. Defaults to a temporary directory.

Value

A list containing the loaded dataset.

---

evaluation.sim *Function Used For Tuning And Evaluation*

Description

Function Used For Tuning And Evaluation

Usage

```
evaluation.sim(pairs.rel, U, seed = NULL)
```

Arguments

pairs.rel	the known code-code pairs
U	the code embedding matrix
seed	Optional integer for reproducibility of sampling.

Value

The output of tuning and evaluation

---

get_embed *Function For Getting Embedding From SVD*

Description

Function For Getting Embedding From SVD

Usage

```
get_embed(mysvd, d = 2000, normalize = TRUE)
```

Arguments

mysvd	the (managed) svd result (adding an element with 'names')
d	dim of the final embedding
normalize	if the output embeddings have l2 norm equal to 1

Value

The embedding from SVD

GroupEff_par *Function Used To Estimate Group Effects Parallelly*

Description

Function Used To Estimate Group Effects Parallelly

Usage

```
GroupEff_par(
  S.MGB,
  S.BCH,
  n.MGB,
  n.BCH,
  U.MGB,
  U.BCH,
  V.MGB,
  V.BCH,
  X.MGB.group,
  X.BCH.group,
  n.group,
  name.list,
  beta.int,
  lambda = 0,
  p,
  n.core
)
```

Arguments

S.MGB	SPPMI from the source site
S.BCH	SPPMI from the target site
n.MGB	the number of codes from the source site
n.BCH	the number of codes from the target site
U.MGB	the left embeddings (left singular vectors times the square root of the singular values) from the source site
U.BCH	the left embeddings (left singular vectors times the square root of the singular values) from the target site
V.MGB	the right embeddings (right singular vectors times the square root of the singular values) from the source site
V.BCH	the right embeddings (right singular vectors times the square root of the singular values) from the target site
X.MGB.group	the dummy matrix based on prior group structures at the source site
X.BCH.group	the dummy matrix based on prior group structures at the target site
n.group	the number of groups
name.list	the full list of code names from the source site and the target site with repeated names of overlapping codes
beta.int	the initial estimator for the group effects
lambda	the tuning parameter controls the intensity of penalization on the group effect; by default we set it to 0
p	the length of an embedding
n.core	the number of cored used for parallel computation

Value

The output of estimating group effects parallelly

MUGS *Main function for MUGS algorithm*

Description

Main function for MUGS algorithm

Usage

```
MUGS(
  TUNE = FALSE,
  Eva = TRUE,
  Lambda = c(10),
  Lambda.delta = c(1000),
  n.core = 4,
  tol = 1,
  seed = NULL,
  S.1 = NULL,
  S.2 = NULL,
  X.group.source = NULL,
  X.group.target = NULL,
  pairs.rel.CV = NULL,
  pairs.rel.EV = NULL,
  p = 100,
  n.group = 400,
  outdir = NULL
)
```

Arguments

TUNE	Logical value indicating whether the function should tune parameters TRUE or use predefined parameters FALSE.
Eva	Logical value indicating whether to perform evaluation (TRUE) or skip it (FALSE).
Lambda	The candidate values for the tuning parameter controlling the intensity of penalization on the code effects.
Lambda.delta	The candidate values for the tuning parameter controlling the intensity of penalization on the code-site effects.
n.core	Integer specifying the number of cores to use for parallel processing.
tol	Numeric value representing the tolerance level for convergence in the algorithm.
seed	Integer used to set the seed for random number generation, ensuring reproducibility. Set to NULL to disable.

S.1	The SPPMI matrix from site 1.
S.2	The SPPMI matrix from site 2.
X.group.source	The dummy matrix representing the group structure of codes at site 1.
X.group.target	The dummy matrix representing the group structure of codes at site 2.
pairs.rel.CV	Code-code pairs used for tuning via cross-validation.
pairs.rel.EV	Code-code pairs used for evaluation.
p	Integer indicating the length of embeddings.
n.group	The number of groups.
outdir	Optional directory to write output files. Defaults to a temporary directory.

Value

A list or saved files containing the embedding matrices, similarity matrices, and site-heterogeneous code analysis.

---

pairs.rel.CV *pairs.rel.CV Dataset*

Description

A data frame containing cross-validation pairs for relative comparisons.

Usage

```
pairs.rel.CV
```

Format

A data frame with multiple columns:

col Integer representing the column index of a pair.

row Integer representing the row index of a pair.

type Character string indicating the type of data (e.g., "train", "test").

pairs.rel.EV *pairs.rel.EV Dataset*

Description

A data frame containing evaluation pairs for relative comparisons.

Usage

```
pairs.rel.EV
```

Format

A data frame with multiple columns:

col Integer representing the column index of a pair.

row Integer representing the row index of a pair.

type Character string indicating the type of data (e.g., "validation").

S.1 *S.1 Dataset*

Description

A matrix containing SPPMI data from the source site. This dataset is used as input for analysis in the package.

Usage

```
S.1
```

Format

A matrix with 2000 rows and 10 columns:

Row Names Unique identifiers for each row.

Columns Numeric values representing SPPMI data.

S.2 *S.2 Dataset*

Description

A matrix containing SPPMI data from the target site. This dataset is used as input for analysis in the package.

Usage

```
S.2
```

Format

A matrix with 2000 rows and 10 columns:

Row Names Unique identifiers for each row.

Columns Numeric values representing SPPMI data.

U.1 *U.1 Dataset*

Description

A matrix containing left embeddings from the source site. These embeddings are used for embedding-based computations.

Usage

```
U.1
```

Format

A matrix with 2000 rows and 10 columns:

Row Names Unique identifiers for each row.

Columns Numeric values representing embeddings.

U.2 *U.2 Dataset*

Description

A matrix containing left embeddings from the target site. These embeddings are used for embedding-based computations.

Usage

```
U.2
```

Format

A matrix with 2000 rows and 10 columns:

Row Names Unique identifiers for each row.

Columns Numeric values representing embeddings.

X.group.source *X.group.source Dataset*

Description

A matrix containing group structures at the source site. It represents binary group membership of entities at the source.

Usage

```
X.group.source
```

Format

A matrix with 2000 rows and 50 columns:

Rows Entities at the source site.

Columns Binary values (0 or 1) indicating group membership.

`X.group.target` *X.group.target Dataset*

Description

A matrix containing group structures at the target site. It represents binary group membership of entities at the target.

Usage

```
X.group.target
```

Format

A matrix with 2000 rows and 50 columns:

Rows Entities at the target site.

Columns Binary values (0 or 1) indicating group membership.

Index