Package 'mvfmr'

May 9, 2026

Type Package

Title Functional Multivariable Mendelian Randomization

Version 0.1.0

Description Implements Multivariable Functional Mendelian Randomization (MV-FMR) to estimate time-varying causal effects of multiple longitudinal exposures on health outcomes. Extends univariable functional Mendelian Randomisation (MR) (Tian et al., 2024 <doi:10.1002/sim.10222>) to the multivariable setting, enabling joint estimation of multiple time-varying exposures with pleiotropy and mediation scenarios. Key features include: (1) data-driven cross-validation for basis component selection, (2) handling of mediation pathways between exposures, (3) support for both continuous and binary outcomes using Generalized Method of Moments (GMM) and control function approaches, (4) one-sample and two-sample MR designs, (5) bootstrap inference and instrument diagnostics including Q-statistics for overidentification testing. Methods are described in Fontana et al. (2025) <doi:10.48550/arXiv.2512.19064>.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 3.5.0)

Imports fdapace, ggplot2 (>= 3.0.0), parallel, doParallel, foreach, pROC, progress, glmnet, gridExtra, stats

Suggests dplyr, tidyr, testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.3.1

NeedsCompilation no

Author Nicole Fontana [aut, cre], Francesca Ieva [aut, ths], Piercesare Secchi [aut, ths]

Maintainer Nicole Fontana <nicole.fontana@polimi.it>

Repository CRAN

Date/Publication 2026-02-09 13:30:09 UTC

Contents

`mvfmr-package` *mvfmr: Multivariable Functional Mendelian Randomization*

Description

Implements Multivariable Functional Mendelian randomization to estimate time-varying causal effects of multiple correlated longitudinal exposures.

Author(s)

Nicole Fontana

`AUTOMATIC_Multi_FMVMR_twosample_simple`
Two-sample joint multivariable FMR (internal)

Description

Two-sample joint multivariable FMR (internal)

Usage

```
AUTOMATIC_Multi_FMVMR_twosample_simple(
  Gmatrix,
  res1,
  res2,
  by_used,
  sy_used,
  ny_used,
  max_nPC1 = NA,
  max_nPC2 = NA,
  X1Ymodel = NA,
  X2Ymodel = NA,
  basis = "eigenfunction"
)
```

Arguments

Gmatrix	Genetic instrument matrix from the exposure sample (N × J)
res1	FPCA result for exposure 1
res2	FPCA result for exposure 2
by_used	Vector of SNP-outcome effect estimates (betas) from the outcome GWAS, length J
sy_used	Vector of standard errors for SNP-outcome effects, length J
ny_used	Sample size of the outcome GWAS
max_nPC1	Maximum number of principal components to retain for exposure 1 (NA = select automatically)
max_nPC2	Maximum number of principal components to retain for exposure 2 (NA = select automatically)
X1Ymodel	True effect model for X1 on Y (for simulation only)
X2Ymodel	True effect model for X2 on Y (for simulation only)
basis	Basis type for functional representation: "eigenfunction" or "polynomial"

Value

List with separate estimation results for both exposures

---

cf_logit *Control function for logit model*

Description

Control function for logit model

Usage

```
cf_logit(
  X,
  Y,
  Z,
  alpha = 1,
  nfolds = 10,
  standardize = TRUE,
  use_lasso = FALSE
)
```

Arguments

X	Matrix of exposure principal components (N x K)
Y	Binary outcome vector (0/1, length N)
Z	Genetic instrument matrix (N x J)
alpha	Elastic net mixing parameter (1=lasso, 0=ridge)
nfolds	Number of cross-validation folds for lambda selection
standardize	Standardize variables before fitting
use_lasso	Use LASSO regularization in first stage. If FALSE, uses OLS.

Value

List with gmm_est, gmm_se, variance_matrix, gmm_pval

---

fmvmr_separate_twosample

Two-Sample Separate Univariable Functional MR

Description

Separate estimation for each exposure using outcome GWAS summary statistics. For single exposure: set G2 = NULL, by2 = NULL, sy2 = NULL.

Usage

```
fmvmr_separate_twosample(
  G1_exposure,
  G2_exposure = NULL,
  fpca_results,
  by_outcome1,
  by_outcome2 = NULL,
  sy_outcome1,
  sy_outcome2 = NULL,
  ny_outcome,
```

```
  max_nPC1 = NA,
  max_nPC2 = NA,
  true_effects = NULL,
  verbose = TRUE
)
```

Arguments

G1_exposure	Genetic instrument matrix from exposure 1 (N × J1)
G2_exposure	Genetic instrument matrix from exposure 2 (N × J2) or NULL for single exposure
fpca_results	List of 2 FPCA objects
by_outcome1	SNP-outcome betas for exposure 1 instruments
by_outcome2	SNP-outcome betas for exposure 2 instruments or NULL
sy_outcome1	Standard errors for exposure 1
sy_outcome2	Standard errors for exposure 2 or NULL
ny_outcome	Outcome GWAS sample size
max_nPC1	Maximum number of principal components to retain for exposure 1 (NA = automatically determined)
max_nPC2	Maximum number of principal components to retain for exposure 2 (NA = automatically determined)
true_effects	List containing true effects for exposure 1 and exposure 2 (simulation only)
verbose	Print progress messages and diagnostics during computation

Value

fmvmr_separate_twosample object

---

fmvmr_twosample *Two-Sample Joint Multivariable Functional MR*

Description

Joint estimation using outcome GWAS summary statistics. Simplified approach: only needs by, sy, ny (not individual outcome data).

Usage

```
fmvmr_twosample(
  G_exposure,
  fpca_results,
  by_outcome,
  sy_outcome,
  ny_outcome,
```

```
  max_nPC1 = NA,
  max_nPC2 = NA,
  true_effects = NULL,
  verbose = TRUE
)
```

Arguments

G_exposure	Genetic instrument matrix from the exposure sample (N × J)
fpca_results	List of 2 FPCA objects
by_outcome	Vector of SNP-outcome effect estimates (betas) from the outcome GWAS, length J
sy_outcome	VVector of standard errors for SNP-outcome effects, length J
ny_outcome	Sample size of the outcome GWAS
max_nPC1	Maximum number of principal components to retain for exposure 1 (NA = automatically determined)
max_nPC2	Maximum number of principal components to retain for exposure 2 (NA = automatically determined)
true_effects	List containing true effects for exposure 1 and exposure 2 (simulation only)
verbose	Print progress messages and diagnostics during computation

Value

fmvmr_twosample object

---

getX_multi_exposure *Generate multi-exposure data with genetic instruments*

Description

Generate multi-exposure data with genetic instruments

Usage

```
getX_multi_exposure(
  N = 10000,
  J = 30,
  ZXmodel = "A",
  nSparse = 10,
  NT = 1000,
  TT = 50,
  shared_effect = TRUE,
  separate_G = FALSE,
  shared_G_proportion = 0.15
)
```

Arguments

N	Sample size
J	Number of genetic instruments
ZXmodel	Model type (currently not used)
nSparse	Number of sparse observations per subject
NT	Number of points
TT	Max observation period
shared_effect	Whether X1 and X2 share confounding
separate_G	Whether to use separate instruments for each exposure
shared_G_proportion	Proportion of shared instruments (0-1)

Value

List with X1, X2 sparse data and genetic instruments

---

getX_multi_exposure_mediation *Generate multi-exposure mediation data with genetic instruments*

Description

Generate multi-exposure mediation data with genetic instruments

Usage

```
getX_multi_exposure_mediation(
  N = 10000,
  J = 30,
  ZXmodel = "A",
  nSparse = 10,
  mediation_strength = 0.3,
  separate_G = FALSE,
  shared_G_proportion = 0,
  mediation_type = "linear"
)
```

Arguments

N	Sample size
J	Number of genetic instruments per exposure
ZXmodel	Model type (currently not used, kept for compatibility)
nSparse	Number of sparse observations per subject

mediation_strength	Strength of mediation X1 -> X2 (default 0.3)
separate_G	Whether to use separate instruments for each exposure
shared_G_proportion	Proportion of shared instruments (0–1)
mediation_type	Character. Type of mediation effect: "linear" (default), "nonlinear", or "time_varying".

Value

List with same structure as getX_multi_exposure()

`getY_multi_exposure` *Generate outcome from exposures*

Description

Generate outcome from exposures

Usage

```
getY_multi_exposure(
  RES,
  X1Ymodel = "1",
  X2Ymodel = "1",
  X1_effect = TRUE,
  X2_effect = TRUE,
  outcome_type = "continuous"
)
```

Arguments

RES	Output from getX_multi_exposure() or getX_multi_exposure_mediation()
X1Ymodel	Effect model for X1 (0-9)
X2Ymodel	Effect model for X2 (0-9)
X1_effect	Include X1 effect?
X2_effect	Include X2 effect?
outcome_type	"continuous" or "binary"

Value

Data frame with outcome Y

gmm_lm_onesample *GMM estimation for continuous outcome*

Description

GMM estimation for continuous outcome

Usage

```
gmm_lm_onesample(X, Y, Z, beta0 = NA)
```

Arguments

X	Matrix of exposure principal components (N x K)
Y	Outcome vector (length N)
Z	Genetic instrument matrix (N x J)
beta0	Initial values for beta (default NA, uses zero initialization)

Value

List with gmm_est, gmm_se, variance_matrix, gmm_pval, Q_stat, Q_pval

gmm_twosample_simple *Two-sample GMM*

Description

Two-sample GMM

Usage

```
gmm_twosample_simple(bx, by, sy, ny)
```

Arguments

bx	Matrix J x K of first-stage coefficients (SNP -> PC associations)
by	Vector length J of outcome GWAS betas
sy	Vector length J of outcome GWAS standard errors
ny	Outcome GWAS sample size

Value

List with gmm_est, gmm_se, variance_matrix, gmm_pval, Q_stat, Q_df, Q_pval

IS *Calculate F-statistics and Q-statistic for instrument strength (internal)*

Description

Calculate F-statistics and Q-statistic for instrument strength (internal)

Usage

```
IS(J, K, PC, datafull, Y = NULL)
```

Arguments

J	Number of genetic instruments
K	Number of exposures
PC	Vector of indices indicating which columns in datafull correspond to the principal components
datafull	Data frame containing instruments (first J columns) and principal components (subsequent columns) [G, X]
Y	Optional outcome vector; if provided, Q-statistic for overidentification is calculated)

Value

Matrix with columns: PC (component index), RR (R-squared), FF (F-statistic), cFF (conditional F-statistic). If Y is provided, additional columns: Qvalue (Hansen's J overidentification test statistic), df (degrees of freedom for Q-test), pvalue (p-value for Q-test from chi-squared distribution).

mvfmr *Joint Multivariable Functional Mendelian Randomization*

Description

Joint Multivariable Functional Mendelian Randomization

Usage

```
mvfmr(
  G,
  fpca_results,
  Y,
  outcome_type = c("continuous", "binary"),
  method = c("gmm", "cf", "cf-lasso"),
  nPC1 = NA,
```

```
  max_nPC1 = NA,
  nPC2 = NA,
  max_nPC2 = NA,
  improvement_threshold = 0.001,
  bootstrap = FALSE,
  n_bootstrap = 100,
  n_cores = parallel::detectCores() - 1,
  true_effects = NULL,
  X_true = NULL,
  verbose = FALSE
)
```

Arguments

G	Genetic instrument matrix (N x J)
fpca_results	List with two FPCA objects from fdapace (res1 and res2)
Y	Outcome vector
outcome_type	Type of outcome: "continuous" for numeric outcomes, "binary" for 0/1 outcomes
method	Estimation method: "gmm" (Generalized Method of Moments), "cf" (control function), or "cf-lasso" (control function with Lasso)
nPC1	Fixed number of principal components to retain for exposure 1 (NA = select automatically)
max_nPC1	Maximum number of principal components to retain for exposure 1 (NA = automatically determined)
nPC2	Fixed number of principal components to retain for exposure 2 (NA = select automatically)
max_nPC2	Maximum number of principal components to retain for exposure 2 (NA = automatically determined)
improvement_threshold	Minimum cross-validation improvement required to add an additional principal component
bootstrap	Whether to compute confidence intervals using bootstrap resampling
n_bootstrap	Number of bootstrap replicates (only used if bootstrap = TRUE)
n_cores	Number of CPU cores to use for parallel computations
true_effects	List with true_effect1 and true_effect2 (simulation only)
X_true	List with X1_true and X2_true curves (simulation only)
verbose	Print progress and diagnostic messages during computation

Value

mvfmr object with:

- coefficients - Estimated beta coefficients
- vcov - Variance-covariance matrix

- effects - List with effect1 and effect2 curves
- nPC_used - Components selected (nPC1, nPC2)
- diagnostics - F-statistics, instrument diagnostics
- performance - MISE, coverage (if true effects provided)

`mvfmr_separate` *Separate Univariable Functional Mendelian Randomization*

Description

Separate Univariable Functional Mendelian Randomization

Usage

```
mvfmr_separate(
  G1,
  G2,
  fpca_results,
  Y,
  outcome_type = c("continuous", "binary"),
  method = c("gmm", "cf", "cf-lasso"),
  nPC1 = NA,
  max_nPC1 = NA,
  nPC2 = NA,
  max_nPC2 = NA,
  improvement_threshold = 0.001,
  bootstrap = FALSE,
  n_bootstrap = 100,
  n_cores = parallel::detectCores() - 1,
  true_effects = NULL,
  X_true = NULL,
  verbose = FALSE
)
```

Arguments

`G1`	Genetic instrument matrix for exposure 1
`G2`	Genetic instrument matrix for exposure 2, or NULL if only a single exposure is analyzed
`fpca_results`	List of FPCA objects
`Y`	Outcome vector
`outcome_type`	Type of outcome: "continuous" for numeric outcomes, "binary" for 0/1 outcomes
`method`	Estimation method: "gmm" (Generalized Method of Moments), "cf" (control function), or "cf-lasso" (control function with Lasso)

nPC1	Fixed number of principal components to retain for exposure 1 (NA = select automatically)
max_nPC1	Maximum number of principal components to retain for exposure 1 (NA = automatically determined)
nPC2	Fixed number of principal components to retain for exposure 2 (NA = select automatically)
max_nPC2	Maximum number of principal components to retain for exposure 2 (NA = automatically determined)
improvement_threshold	Minimum cross-validation improvement required to add an additional principal component
bootstrap	Whether to compute confidence intervals using bootstrap resampling
n_bootstrap	Number of bootstrap replicates (only used if bootstrap = TRUE)
n_cores	Number of CPU cores to use for parallel computations
true_effects	List with true_effect1 and true_effect2 (simulation only)
X_true	List with X1_true and X2_true curves (simulation only)
verbose	Print progress and diagnostic messages during computation

Value

fmvmr_separate object

Separate_Multi_FMVMR_twosample_simple

Separate univariable two-sample FMR (internal)

Description

Separate univariable two-sample FMR (internal)

Usage

```
Separate_Multi_FMVMR_twosample_simple(
  Gmatrix1,
  Gmatrix2 = NULL,
  res1,
  res2,
  by_used1,
  by_used2 = NULL,
  sy_used1,
  sy_used2 = NULL,
  ny_used,
  max_nPC1 = NA,
  max_nPC2 = NA,
```

```
  X1Ymodel = NA,
  X2Ymodel = NA,
  basis = "eigenfunction"
)
```

Arguments

Gmatrix1	Genetic instrument matrix from exposure 1 (N × J1)
Gmatrix2	Genetic instrument matrix from exposure 2 (N × J2) or NULL
res1	FPCA result for exposure 1
res2	FPCA result for exposure 2
by_used1	Vector of SNP-outcome effect estimates (betas for X1) from the outcome GWAS, length J
by_used2	Vector of SNP-outcome effect estimates (betas for X2) from the outcome GWAS, length J or NULL
sy_used1	Vector of standard errors for SNP-outcome effects for X1, length J
sy_used2	Vector of standard errors for SNP-outcome effects for X2, length J or NULL
ny_used	Sample size of the outcome GWAS
max_nPC1	Maximum number of principal components to retain for exposure 1 (NA = select automatically)
max_nPC2	Maximum number of principal components to retain for exposure 2 (NA = select automatically)
X1Ymodel	True effect model for X1 on Y (for simulation only)
X2Ymodel	True effect model for X2 on Y (for simulation only)
basis	Basis type for functional representation: "eigenfunction" or "polynomial"

Value

List with separate estimation results for both exposures

Index