

# Package: messi (via r-universe)

September 6, 2024

**Type** Package

**Title** Mediation Analysis with External Summary-Level Information on Total Effect

**Version** 0.1.1

**Description** Fits the hard constraint, soft constraint, and unconstrained models in Boss et al. (2023) for mediation analyses with external summary-level information on the total effect.

**Maintainer** Michael Kleinsasser <mkleinsa@umich.edu>

**License** GPL-2

**Encoding** UTF-8

**LazyData** true

**Biarch** true

**Depends** R (>= 3.6.2)

**Imports** MASS, ggplot2, patchwork, progress

**URL** <https://github.com/umich-cphds/messi>

**BugReports** <https://github.com/umich-cphds/messi/issues>

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.3

**Repository** <https://umich-cphds.r-universe.dev>

**RemoteUrl** <https://github.com/umich-cphds/messi>

**RemoteRef** HEAD

**RemoteSha** 4af85d7ca0d5c3376c5c4971373b1a27dca8947a

## Contents

constrained.unpenalized . . . . .	2
Med . . . . .	3
messi . . . . .	3

nullMed . . . . .	5
plot_messi . . . . .	5
rand.eff.coord.desc.unpenalized . . . . .	6
rand.eff.unpenalized . . . . .	7
unconstrained.unpenalized . . . . .	8

## Index 9

---

constrained.unpenalized

*Estimate hard constraint model parameters using cyclical coordinate descent.*

---

### Description

Estimate hard constraint model parameters using cyclical coordinate descent.

### Usage

```
constrained.unpenalized(
  Y,
  M,
  A,
  C = NULL,
  T.hat.external,
  err.tol.out = 1e-08,
  err.tol.med = 1e-08,
  max.itr = 10000
)
```

### Arguments

Y	A (n x 1) continuous outcome vector.
M	A (n x p <sub>m</sub> ) matrix of mediators.
A	A (n x 1) vector of exposures.
C	A (n x p <sub>c</sub> ) matrix of confounders and adjustment covariates. If there are no confounders or adjustment covariates set C = NULL.
T.hat.external	External estimate of the total effect.
err.tol.out	Termination condition for cyclical coordinate descent algorithm with respect to the outcome model parameters.
err.tol.med	Termination condition for cyclical coordinate descent algorithm with respect to the mediator model parameters.
max.itr	Maximum number of iterations for cyclical coordinate descent algorithm.

### Value

A list containing point estimates of the hard constraint model parameters and an indicator of whether the algorithm converges.

---

Med	<i>Simulated data with real mediation effect</i>
-----	--

---

**Description**

Simulated data with real mediation effect

**Usage**

```
Med
```

**Format**

An object of class `list` of length 8.

**Examples**

```
data(Med)
```

---

messi	<i>Implementation of Mediation with External Summary Statistics Information (MESSI) from Boss et al. (2023).</i>
-------	--

---

**Description**

Implementation of Mediation with External Summary Statistics Information (MESSI) from Boss et al. (2023).

**Usage**

```
messi(  
  Y,  
  M,  
  A,  
  C = NULL,  
  method = "Soft EB",  
  T.hat.external,  
  var.T.hat.external,  
  n.boot = 200,  
  s2.fixed = NULL  
)
```

**Arguments**

Y	A (n x 1) continuous outcome vector.
M	A (n x p <sub>m</sub> ) matrix of mediators.
A	A (n x 1) vector of exposures.
C	A (n x p <sub>c</sub> ) matrix of confounders and adjustment covariates. If there are no confounders or adjustment covariates set C = NULL.
method	A string specifying which method to use. Options include 'Unconstrained', 'Hard', 'Soft EB', and 'Soft Fixed'. Default is 'Soft EB'.
T.hat.external	External estimate of the total effect. Set to NULL if method = 'Unconstrained'.
var.T.hat.external	Estimated variance of the external estimator of the total effect. Set to NULL if method = 'Unconstrained' or method = 'Hard'.
n.boot	Number of parametric bootstrap draws for obtaining quantile-based confidence intervals for the TE and NDE. Relevant for method = 'Soft EB' and method = 'Soft Fixed'. Can set to NULL for method = 'Unconstrained' and method = 'Hard'.
s2.fixed	Option to specify the tuning parameter $s^2$ in the soft constraint model. Only use if method = 'Soft Fixed'.

**Details**

The Soft EB method should be the default method if the user is not sure which method to use.

**Value**

A list containing the (1) point estimates and confidence intervals for the natural direct effect, the natural indirect effect, and the total effect (2) point estimates for all mediation model parameters (3) the asymptotic variance covariance matrix corresponding to  $\alpha_a$  and  $\beta_m$ .

**Examples**

```
data(Med)

Y = Med$Y
M = Med$M
A = Med$A
C = Med$C
T.hat.external = Med$T.hat.external
var.T.hat.external = Med$var.T.hat.external

test <- messi(Y = Y, M = M, A = A, C = C, method = 'Unconstrained', T.hat.external = T.hat.external,
             var.T.hat.external = var.T.hat.external, s2.fixed = NULL)

n = Med$n
p = Med$p

plot_messi(n = n, alpha.a.hat = test$alpha.a.hat, beta.m.hat = test$beta.m.hat,
           labels = paste0("M", 1:p), asym.var.mat = test$asym.var.mat)
```

```
test <- messi(Y = Y, M = M, A = A, C = C, method = 'Hard', T.hat.external = T.hat.external,
             var.T.hat.external = var.T.hat.external, s2.fixed = NULL)
```

---

nullMed	<i>Simulated data with null mediation effect</i>
---------	--

---

### Description

Simulated data with null mediation effect

### Usage

```
nullMed
```

### Format

An object of class `list` of length 4.

### Examples

```
data(nullMed)
```

---

plot_messi	<i>Forestplot to Summarize Estimation and Inference on alpha_a and beta_m.</i>
------------	--

---

### Description

Forestplot to Summarize Estimation and Inference on alpha\_a and beta\_m.

### Usage

```
plot_messi(n, alpha.a.hat, beta.m.hat, labels, asym.var.mat)
```

### Arguments

n	Sample size of the analysis
alpha.a.hat	Estimate of alpha_a, a (p_m x 1) vector.
beta.m.hat	Estimate of beta_m, a (p_m x 1) vector.
labels	A (p_m x 1) vector of mediator names. Make sure that the labels are in the same order as the mediators appear in the design matrix.
asym.var.mat	Joint asymptotic variance-covariance matrix of alpha_a and beta_m, a (2p_m x 2p_m) matrix.

**Value**

Data frames and forestplots summarizing alpha\_a and beta\_m estimation.

**Examples**

```
data(Med)

Y = Med$Y
M = Med$M
A = Med$A
C = Med$C
T.hat.external = Med$T.hat.external
var.T.hat.external = Med$var.T.hat.external

test <- messi(Y = Y, M = M, A = A, C = C, method = 'Unconstrained', T.hat.external = T.hat.external,
              var.T.hat.external = var.T.hat.external, s2.fixed = NULL)

n = Med$n
p = Med$p

plot_messi(n = n, alpha.a.hat = test$alpha.a.hat, beta.m.hat = test$beta.m.hat,
           labels = paste0("M",1:p), asym.var.mat = test$asym.var.mat)
```

---

rand.eff.coord.desc.unpenalized

*Cyclical coordinate descent algorithm for the M-step in the EM Algorithm for the maximizing the soft constraint model likelihood.*

---

**Description**

Cyclical coordinate descent algorithm for the M-step in the EM Algorithm for the maximizing the soft constraint model likelihood.

**Usage**

```
rand.eff.coord.desc.unpenalized(
  Y,
  M,
  A,
  C = NULL,
  first.moment,
  second.moment,
  err.tol.out = 1e-08,
  err.tol.med = 1e-08,
  max.itr = 10000
)
```

**Arguments**

Y	A (n x 1) continuous outcome vector.
M	A (n x p_m) matrix of mediators.
A	A (n x 1) vector of exposures.
C	A (n x p_c) matrix of confounders and adjustment covariates. If there are no confounders or adjustment covariates set C = NULL.
first.moment	Posterior expectation of the total effect parameter.
second.moment	Posterior expectation of the squared total effect parameter.
err.tol.out	Termination condition for cyclical coordinate descent algorithm with respect to the outcome model parameters.
err.tol.med	Termination condition for cyclical coordinate descent algorithm with respect to the mediator model parameters.
max.itr	Maximum number of iterations for cyclical coordinate descent algorithm.

**Value**

A list containing point estimates of the soft constraint model parameters and an indicator of whether the algorithm converges.

---

rand.eff.unpenalized *Estimate soft constraint model parameters using the EM algorithm.*

---

**Description**

Estimate soft constraint model parameters using the EM algorithm.

**Usage**

```
rand.eff.unpenalized(
  Y,
  M,
  A,
  C = NULL,
  rand.eff.mean,
  rand.eff.var,
  T.hat.external = T.hat.external,
  var.T.hat.external = var.T.hat.external,
  err.tol.out = 1e-08,
  err.tol.med = 1e-08,
  max.itr = 10000
)
```

**Arguments**

Y	A (n x 1) continuous outcome vector.
M	A (n x p <sub>m</sub> ) matrix of mediators.
A	A (n x 1) vector of exposures.
C	A (n x p <sub>c</sub> ) matrix of confounders and adjustment covariates. If there are no confounders or adjustment covariates set C = NULL.
rand.eff.mean	Mean of the random effects distribution for the internal total effect parameter.
rand.eff.var	Variance of the random effects distribution for the internal total effect parameter.
T.hat.external	External estimate of the total effect.
var.T.hat.external	Estimated variance of the external total effect estimator.
err.tol.out	Termination condition for cyclical coordinate descent algorithm with respect to the outcome model parameters.
err.tol.med	Termination condition for cyclical coordinate descent algorithm with respect to the mediator model parameters.
max.itr	Maximum number of iterations for cyclical coordinate descent algorithm.

**Value**

A list containing point estimates of the soft constraint model parameters and an indicator of whether the algorithm converges.

---

unconstrained.unpenalized

*Estimate unconstrained model parameters.*

---

**Description**

Estimate unconstrained model parameters.

**Usage**

```
unconstrained.unpenalized(Y, M, A, C = NULL)
```

**Arguments**

Y	A (n x 1) continuous outcome vector.
M	A (n x p <sub>m</sub> ) matrix of mediators.
A	A (n x 1) vector of exposures.
C	A (n x p <sub>c</sub> ) matrix of confounders and adjustment covariates. If there are no confounders or adjustment covariates set C = NULL.

**Value**

A list containing point estimates of the unconstrained model parameters.



# Index

## \* datasets

Med, [3](#)

nullMed, [5](#)

constrained.unpenalized, [2](#)

Med, [3](#)

messi, [3](#)

nullMed, [5](#)

plot\_messi, [5](#)

rand.eff.coord.desc.unpenalized, [6](#)

rand.eff.unpenalized, [7](#)

unconstrained.unpenalized, [8](#)