Package ‘LSAMBA’

January 19, 2026

Type Package

Title Lasso-SAMBA Algorithm

Version 2024.1.1

Maintainer Auriane Gabaut <aurianegabaut@gmail.com>

Description Provides a lasso-based method for building mechanistic models using the SAMBA algorithm (Stochastic Approximation for Model Building Algorithm) (M Prague, M Lavielle (2022) <doi:10.1002/psp4.12742>). The package extends the 'Rsmlx' package (version 2024.1.0) to better handle high-dimensional data. It relies on the 'Monolix' software (version 2024R1; see (<https://monolixsuite.slp-software.com/monolix/2024R1/>), which must be installed beforehand.

SystemRequirements 'Monolix' (<https://monolixsuite.slp-software.com/monolix/2024R1/>)

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

Imports doParallel, dplyr, ggplot2, grDevices, latex2exp, MASS, parallel, sharp, stringr

Depends R (>= 2.10), Rsmlx (>= 2024.1.0), foreach

NeedsCompilation no

Author Auriane Gabaut [aut, cre],
Mélanie Prague [aut],
Marc Lavielle [aut] (for former code of Rsmlx and connectors)

Repository CRAN

Date/Publication 2026-01-19 15:30:09 UTC

Contents

buildmlx *buildmlx: Automatic statistical model building*

Description

buildmlx uses SAMBA (Stochastic Approximation for Model Building Algorithm), an iterative procedure to accelerate and optimize the process of model building by identifying at each step how best to improve some of the model components. This method allows to find the optimal statistical model which minimizes some information criterion in very few steps.

Usage

```
buildmlx(
  project = NULL,
  final.project = NULL,
  model = "all",
  prior = NULL,
  weight = NULL,
  coef.w1 = 0.5,
  paramToUse = "all",
  covToTest = "all",
  covToTransform = "none",
  center.covariate = FALSE,
  criterion = "BICc",
  linearization = FALSE,
  ll = TRUE,
  test = FALSE,
  direction = NULL,
  steps = 1000,
  n.full = 10,
  max.iter = 20,
  explor.iter = 2,
  fError.min = 0.001,
  seq.cov = FALSE,
  seq.cov.iter = 0,
  seq.corr = TRUE,
  p.max = if (buildMethod == "stepAIC") {
     0.1
 } else {
     1
 },
  p.min = c(0.075, 0.05, 0.1),
  print = TRUE,
  nb.model = 1,
  nfolds = 5,
  alpha = 1,
  nSS = 1000,
  buildMethod = "lasso",
  FDR_thr = 0.1
)
```

Arguments

`project` a string: the initial Monolix project

`final.project` a string: the final Monolix project (default adds "_built" to the original project)

`model` components of the model to optimize c("residualError", "covariate", "correlation"), (default="all")

`prior` list of prior probabilities for each component of the model (default=NULL)

`weight` list of penalty weights for each component of the model (default=NULL)

`coef.w1` multiplicative weight coefficient used for the first iteration only (default=0.5)

`paramToUse` list of parameters possibly function of covariates (default="all")

`covToTest` components of the covariate model that can be modified (default="all")

`covToTransform` list of (continuous) covariates to be log-transformed (default="none")

`center.covariate` TRUE/FALSE center the covariates of the final model (default=FALSE)

`criterion` penalization criterion to optimize c("AIC", "BIC", "BICc", gamma) (default=BICc)

`linearization` TRUE/FALSE whether the computation of the likelihood is based on a linearization of the model (default=FALSE)

`ll` TRUE/FALSE compute the observe likelihood and the criterion to optimize at each iteration

`test` TRUE/FALSE perform additional statistical tests for building the model (default=FALSE)

`direction` for stepAIC method, method for covariate search c("full", "both", "backward", "forward"), (default="full" or "both")

`steps` for stepAIC method, maximum number of iteration for stepAIC (default=1000)

`n.full` for stepAIC method, maximum number of covariates for an exhaustive comparison of all possible covariate models (default=10)

`max.iter` maximum number of iterations (default=20)

`explor.iter` number of iterations during the exploratory phase (default=2)

`fError.min` minimum fraction of residual variance for combined error model (default = 1e-3)

`seq.cov` TRUE/FALSE whether the covariate model is built before the correlation model

`seq.cov.iter` number of iterations before building the correlation model (only when seq.cov=F, default=0)

`seq.corr` TRUE/FALSE whether the correlation model is built iteratively (default=TRUE)

`p.max` maximum p-value used for removing non significant relationships between covariates and individual parameters (default=0.1 for stepAIC and 1 for lasso)

`p.min` vector of 3 minimum p-values used for testing the components of a new model (default=c(0.075, 0.05, 0.1)) `print` TRUE/FALSE display the results (default=TRUE)

`nb.model` number of models to display at each iteration (default=1)

`nfolds` for lasso method, number of folds (default=10)

`alpha` for lasso method, the elasticnet mixing parameter, between 0 and 1. `alpha=1` is the lasso penalty, `alpha=0` is the ridge penalty.

`nSS` for lasso method, number of resampling iterations for stability selection.

`buildMethod` the method used to build the covariate model (default="lasso")

`FDR_thr` for lasso method, upper-bounds in FDP of calibrated stability selection (default=0.1)

Details

For covariates model building, covariate selection can be achieved by stepAIC, as the original SAMBA algorithm was implemented in Rsmlx package (Prague and Lavielle, 2020 ; Mihaljevic, 2023) and by a lasso approach enhanced by stability selection (Bodinier et al., 2023).

Value

a new Monolix project with a new statistical model.

References

Prague M, Lavielle M. SAMBA: A novel method for fast automatic model building in nonlinear mixed-effects models. CPT Pharmacometrics Syst Pharmacol. 2022; 11: 161-172. doi:10.1002/psp4.12742

Bodinier B, Filippi S, Haugdahl Nøst T, Chiquet J, Chadeau-Hyam M. Automated calibration for stability selection in penalised regression and graphical models. Journal of the Royal Statistical Society Series C: Applied Statistics. 2023 ; 72: 1375–1393. doi:10.1093/jrsssc/qlad058

Mihaljevic F (2023). Rsmlx: R Speaks 'Monolix'. R package version2023.1.5, https://CRAN.R-project.org/package=Rsmlx.

Examples

```
## Not run:
project <- getMLXdir()

res = buildmlx(project = project,
               buildMethod = "lasso",
               model='covariate',
               test=FALSE)

getIndividualParameterModel()

## End(Not run)
```

`getMLXdir` *Get monolix demo project path.*

Description

Get monolix demo project path.

Usage

```
getMLXdir()
```

Value

path to the monolix demo from 'Monolix' software

Examples

```
## Not run:
print(getMLXdir())

## End(Not run)
```

Index