Package 'opGMMassessment'

April 17, 2026

Type Package

Title Optimized Automated Gaussian Mixture Assessment

Version 0.4.1

Maintainer Jorn Lotsch <j.lotsch@em.uni-frankfurt.de>

Description
Necessary functions for optimized automated evaluation of the number and parameters of Gaussian mixtures in one-dimensional data. Various methods are available for parameter estimation and for determining the number of modes in the mixture. A detailed description of the methods ca ben found in Lotsch, J., Malkusch, S. and A. Ultsch. (2022) <doi:10.1016/j.imu.2022.101113>.

URL https://github.com/JornLotsch/opGMMassessment

Depends R (>= 3.5.0)

License GPL-3

Encoding UTF-8

LazyData true

Imports AdaptGauss, DataVisualizations, DistributionOptimization, cluster, mixtools, grDevices, methods, foreach, stats, utils, rlang, ggplot2, parallel, caTools, dplyr, mclust, mixAK, multimode, NbClust, ClusterR, doParallel

NeedsCompilation no

Author Jorn Lotsch [aut, cre] (ORCID: <https://orcid.org/0000-0002-5818-6958>), Sebastian Malkusch [aut] (ORCID: <https://orcid.org/0000-0001-6766-140X>)

Repository CRAN

Date/Publication 2026-04-17 05:10:57 UTC

Contents

`Chromatogram` *Example data of lysophosphatidic acids, LPA.*

Description

Data set containing times of detector hits after chromatographic separation of five different lysophosphatidic acids (Classes CLs = LPA 16:0, 18:0, 18:3, 20:0, and 20:4).

Usage

```
data("Chromatogram")
```

Details

Size 1166 x 3 , stored in `Chromatogram$[Cls, Time, Lipids]`

Examples

```
data(Chromatogram)
str(Chromatogram)
```

`GMMplotGG` *Plot of Gaussian mixtures*

Description

The function plots the components of a Gaussian mixture and superimposes them on a histogram of the data.

Usage

```
GMMplotGG(Data, Means, SDs, Weights, BayesBoundaries,
SingleGausses = TRUE, Hist = FALSE, Bounds = TRUE, SumModes = TRUE, PDE = TRUE)
```

Arguments

`Data`	the data as a vector.
`Means`	a list of mean values for a Gaussian mixture.
`SDs`	a list of standard deviations for a Gaussian mixture.
`Weights`	a list of weights for a Gaussian mixture.
`BayesBoundaries`	a list of Bayesian boundaries for a Gaussian mixture.
`SingleGausses`	whether to plot the single Gaussian components as separate lines.

`Hist`	whether to plot a histgram of the original data.
`Bounds`	whether to plot the Bayesian boundaries for a Gaussian mixture as vertical lines.
`SumModes`	whether to plot the summed-up mixes.
`PDE`	whether to use the Pareto density estimation instead of the standard R density function.

Value

Returns a ggplot2 object.

`p1`	the plot of Gaussian mixtures.

Author(s)

Jorn Lotsch and Sebastian Malkusch

References

Lotsch, J., Malkusch S. (2021): opGMMassessment – an R Package for automated Guassian mixture modeling.

Examples

```
## example 1
data(iris)
Means0 <- tapply(X = as.vector(iris[,3]), INDEX =  as.integer(iris$Species), FUN = mean)
SDs0 <- tapply(X = as.vector(iris[,3]), INDEX =  as.integer(iris$Species), FUN = sd)
Weights0 <- c(1/3, 1/3, 1/3)
GMM.Sepal.Length <- GMMplotGG(Data = as.vector(iris[3]),
Means = Means0,
SDs = SDs0,
Weights = Weights0,
Hist = TRUE)
```

`Mixture3` *Example Gaussian mixture data.*

Description

Data set containing 1000 instances distributed according to a Gaussian mixture with m = [-10, 0, 10], s = [1, 2, 3], w = [0.07, 0.05, 0.88].

Usage

```
data("Mixture3")
```

Details

Size 1000 x 1

Examples

```
data(Mixture3)
str(Mixture3)
```

`opGMMassessment` *Gaussian mixture assessment*

Description

The package provides the necessary functions for optimized automated evaluation of the number and parameters of Gaussian mixtures in one-dimensional data. It provides various methods for parameter estimation and for determining the number of modes in the mixture.

Usage

```
opGMMassessment(Data, FitAlg = "MCMC", Criterion = "LR",
MaxModes = 8, MaxCores = getOption("mc.cores", 2L), PlotIt = FALSE, KS = TRUE, Seed)
```

Arguments

`Data`	the data as a numerical vector (1D).
`FitAlg`	which fit algorithm to use: "ClusterRGMM" = GMM from ClusterR, "densityMclust" from mclust, "DO" from DistributionOptimization (slow), "MCMC" = NMixMCMC from mixAK, or "normalmixEM" from mixtools.
`Criterion`	which criterion should be used to establish the number of modes from the best GMM fit: "AIC", "BIC", "FM", "GAP", "LR" (likelihood ratio test), "NbClust" (from NbClust), "SI" (Silverman).
`MaxModes`	the maximum number of modes to be tried.
`MaxCores`	the maximum number of processor cores used under Unix.
`PlotIt`	whether to plot the fit directly (plot will be stored nevertheless).
`KS`	perform a Kolmogorow-Smirnow test of the fit versus original distribution.
`Seed`	optional seed parameter set internally.

Value

Returns a list of Gaussian modes.

`Cls`	the classes to which the cases are assigned according to the Gaussian mode membership.
`Means`	means of the Gaussian modes.
`SDs`	standard deviations of the Gaussian modes.
`Weights`	weights of the Gaussian modes.
`Boundaries`	Bayesian boundaries between the Gaussian modes.
`Plot`	Plot of the obtained mixture.
`KS`	Results of the Kolmogorov-Smirnov test.

Author(s)

Jorn Lotsch and Sebastian Malkusch

References

Lotsch J, Malkusch S, Ultsch A. Comparative assessment of automated algorithms for the separation of one-dimensional Gaussian mixtures. Informatics in Medicine Unlocked, Volume 34, 2022, https://doi.org/10.1016/j.imu.2022.101113. (https://www.sciencedirect.com/science/article/pii/S2352914822002507)

Examples

```
## example 1
data(iris)
opGMMassessment(Data = iris$Petal.Length,
  FitAlg = "normalmixEM",
  Criterion = "BIC",
  PlotIt = TRUE,
  MaxModes = 5,
  MaxCores = 1,
  Seed = 42)
```

Index