

Package ‘rocTools’

May 7, 2026

Type Package

Title Tools for Distribution-Based ROC Smoothing Using 'pROC'

Version 0.1.3

Description Extends the functionality of the 'pROC' package for conducting smoothed receiver operating characteristic (ROC) curve analysis. Enables automated selection of the distribution families to be fit when smoothing ROC curves via the population probability density function estimation strategy described by Leeftang et al. (2008) <[doi:10.1373/clinchem.2007.096032](https://doi.org/10.1373/clinchem.2007.096032)>, as well as generation of diagnostic performance and cutoff estimates from the resultant smoothed curves.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Imports MASS, pROC (>= 1.19.0)

Suggests testthat (>= 3.0.0)

Author Grant C. O'Connell [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-2642-6670>>)

Maintainer Grant C. O'Connell <goconnell.phd@gmail.com>

Repository CRAN

Depends R (>= 3.5)

Config/testthat/edition 3

NeedsCompilation no

Date/Publication 2026-05-07 16:10:13 UTC

Contents

rocTools-package	2
blood_data	4
find.distr	5
get.fits	7
sm.coords	9

Description

Receiver operating characteristic (ROC) analysis is widely used in studies of diagnostic performance to generate key metrics such as sensitivity and specificity estimates, as well as establish candidate diagnostic cutoffs. However, population sample measures have a finite number of possible diagnostic cutoffs, which can result in raw ROC curves that appear jagged or stepped. This stepping phenomenon can inflate diagnostic performance estimates and reduce the accuracy of diagnostic cutoff estimates, particularly in small to modestly sized samples (Linnet and Brandt, 1986). One strategy to reduce this bias and increase generalizability is to carry out ROC smoothing using population probability density function estimation. While this solution is robust when properly implemented, smoothing using poorly fit distributions can unintentionally increase bias (Leeftang et al., 2008). To that end, the ‘rocTools’ package provides a set of functions that extend the functionality of the ‘pROC’ package for carrying out population probability density function estimation-based ROC smoothing and associated analyses. These functions enable automated selection of the case and control distribution families to be fit for smoothing, as well as generation of population-level diagnostic performance and diagnostic cutoff estimates from the resultant smoothed curves.

Details

The ‘rocTools’ package contains three user facing functions that can be used in combination with the core functions of the ‘pROC’ package to build and analyze smoothed ROC curves via population probability density function estimation methods.

The `find.distr()` function performs automated selection of the distribution families to be fit when generating smoothed ROC curves using the “fitdistr” method of the `smooth()` function of the ‘pROC’ package. It takes a raw roc object generated by the `roc()` function of the ‘pROC’ package as input, fits up to four different distribution families to both the case and control data, and then uses Akaike information criterion (AIC) to select and return the best fitting distribution types along with any associated start values. These returned parameters can then be directly passed to `smooth()` to be used for ROC smoothing.

The `get.fits()` function retrieves and returns the final case and control distribution fits from `smooth.roc` objects generated using the “fitdistr” method of the `smooth()` function. It can be used downstream of `find.distr()` and `smooth()` to evaluate the final distribution fits used for ROC smoothing. The returned values include the distribution types, the distribution parameters, and the associated AIC values. The extracted values can either be printed in console as a human readable summary or stored as a new object.

The `sm.coords()` function calculates population-level sensitivity, specificity, and diagnostic cutoff estimates from smoothed ROC curves generated using the “fitdistr” method of the `smooth()` function. A `smooth.roc` object is provided as input, and these metrics can be calculated for the estimated diagnostic cutoff that maximizes the Youden index, or for the estimated diagnostic cutoff that achieves a user specified sensitivity or specificity value. This function works analogously to the `coords()` function of the ‘pROC’ package, but produces more precise sensitivity and specificity

estimates for smoothed curves through continuous optimization as opposed to grid-based optimization, and has the added functionality of producing diagnostic cutoff estimates.

From an implementation perspective, this collection of functions can be called in the following order to create a seamless workflow to generate population-level diagnostic performance and cutoff estimates from case-control population sample measures: `roc()` to build a raw ROC curve for the population sample data, `find.distr()` to select the best case and control distribution families to fit for roc smoothing, `smooth()` to build a smooth ROC curve approximating population-level diagnostic performance using said distribution families, and `sm.coords()` to calculate population-level sensitivity, specificity, and diagnostic cutoff estimates associated with said smooth curve. `get.fits()` can be used as needed to retrieve the final distribution fits generated as part of this workflow.

A real-world dataset containing blood biomarker measures is also included and is used throughout the documentation examples. The `blood_data` dataset contains plasma neurofilament-light chain (NfL) levels measured in 33 neurologically normal controls and 14 patients diagnosed with ischemic stroke as described by O’Connell et al. (2020).

Main functions

`find.distr()` Automatically selects distribution families to be used for ROC smoothing.

`get.fits()` Returns distribution fits for smoothed ROC curves.

`sm.coords()` Calculates diagnostic performance and cutoff estimates from smoothed ROC curves.

Datasets

`blood_data` Example blood biomarker dataset.

Citation

If you use the ‘rocTools’ package in published work, please cite:

O’Connell, GC. (2026). *rocTools*. R package version 0.1.3. Available from <https://CRAN.R-project.org/package=rocTools>.

References

Linnet, K. & Brandt, E. (1986). Assessing diagnostic tests once an optimal cutoff point has been selected. *Clinical Chemistry*, 32, 7. doi:10.1093/clinchem/32.7.1341

Leeflang, MMG., Moons, KGM., Reitsma, JB., & Zwinderman, AH. (2008). Bias in Sensitivity and Specificity Caused by Data-Driven Selection of Optimal Cutoff Values: Mechanisms, Magnitude, and Solutions. *Clinical Chemistry*, 54, 4. doi:10.1373/clinchem.2007.096032

Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, JC., & Müller, M. (2011). pROC: an open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinformatics*, 12, 77. doi:10.1186/14712105127

O’Connell, GC., Alder, ML., Smothers, CG., Still, CH., Webel, AR., & Moore, S.M. (2020). Diagnosis of ischemic stroke using circulating levels of brain-specific proteins measured via high-sensitivity digital ELISA. *Brain Research*, 15, 1739. doi:10.1016/j.brainres.2020.146861

See Also

```
pROC-package  
pROC::roc()  
pROC::smooth()  
pROC::coords()  
find.distr()  
get.fits()  
sm.coords()  
blood_data
```

blood_data

Example blood biomarker dataset.

Description

A real-world dataset containing plasma neurofilament-light chain (NfL) levels measured in 33 neurologically normal controls and 14 patients diagnosed with ischemic stroke as described by O'Connell et al. (2020).

Format

A data frame with one column containing stroke diagnoses and another column containing plasma neurofilament-light chain levels in pg/mL.

References

O'Connell, GC., Alder, ML., Smothers, CG., Still, CH., Webel, AR., & Moore, S.M. (2020). Diagnosis of ischemic stroke using circulating levels of brain-specific proteins measured via high-sensitivity digital ELISA. *Brain Research*, 15, 1739. doi:[10.1016/j.brainres.2020.146861](https://doi.org/10.1016/j.brainres.2020.146861)

Examples

```
#load example dataset  
  
data(blood_data)  
  
#return dataset structure  
  
str(blood_data)
```

find.distr	<i>Automatically select distribution families to be used for ROC smoothing.</i>
------------	---

Description

Automatically selects the best fitting distribution families to be used when smoothing ROC curves via population probability density function estimation. It takes a raw roc object generated by the roc() function of the 'pROC' package as input, fits up to four different candidate distribution families to both the case and control data, and then uses AIC to select and return the best fitting distribution types along with any associated start values. These returned parameters can then be directly passed to the smooth() function of the 'pROC' package for generation of a smoothed ROC curve that approximates population-level diagnostic performance using the "fitdistr" method.

Usage

```
find.distr(  
  roc,  
  candidates = c("normal", "gamma", "lognormal", "weibull"),  
  search.mode = c("all_pairs", "per_group"),  
  quiet = FALSE  
)
```

Arguments

roc	A roc object generated by the roc() function.
candidates	A character vector specifying the candidate distribution families to be tested. Options include "normal", "lognormal", "gamma", and "weibull". If NULL, all four families will be tested by default.
search.mode	Character string specifying what search mode to use to select the best fitting distribution families. One of "all_pairs" or "per_group". If "all_pairs", all possible case and control distribution family combinations are tested jointly, and the pair that minimizes the combined AIC across both groups is selected and returned. If "per_group", distribution families are tested independently for the case and control groups, and the distribution families that respectively minimize AIC within each group are selected and returned. Default is "all_pairs". If "all_pairs" mode fails, "per_group" mode is used as a fallback with warning message.
quiet	Specifies whether to silence messages generated during distribution selection. Default value is TRUE.

Details

Similar to smooth(), all fits are internally performed using the fitdistr() function of the 'MASS' package. Method of moments is used to establish starting values for distribution types where they are explicitly needed. When using find.distr() in tandem with smooth(), get.fits() can be

used to retrieve the final fits ultimate used by `smooth()` for generation of the smooth ROC curve. Furthermore, `sm.coords()` can be used to calculate population-level diagnostic performance and cutoff estimates from the smoothed curve.

Value

A named list containing the parameters associated with the best fitting case and control distribution families. The returned list can be passed directly to `smooth()` for generation of a smooth ROC curve using the "fitdistr" method, and has the following elements:

<code>density.controls</code>	A character string specifying the distribution family selected for the control group.
<code>density.cases</code>	A character string specifying the distribution family selected for the case group.
<code>start.controls</code>	An optional named list of starting values for fitting the control distribution. Returned only for distribution families that require explicit starting values.
<code>start.cases</code>	An optional named list of starting values for fitting the case distribution. Returned only for distribution families that require explicit starting values.

Diagnostic information associated with model selection process is attached attributes, including all calculated AIC values and the final search mode.

See Also

[pROC-package](#)
[pROC::roc\(\)](#)
[pROC::smooth\(\)](#)
[MASS::fitdistr\(\)](#)
[rocTools-package](#)
[get.fits\(\)](#)
[sm.coords\(\)](#)
[blood_data](#)

Examples

```
# EXAMPLE USING LONG CALLS

# load packages and example blood biomarker data

library(pROC)
library(rocTools)
data(blood_data)

# build raw ROC curve

raw_roc <- roc(
  blood_data,
  response = "Diagnosis",
  predictor = "NFL"
)
```

```
# find best fitting case and control distribution
# families to use for ROC smoothing

best_fits <- find.distr(
  raw_roc,
  candidates = c("normal", "gamma", "lognormal", "weibull"),
  search.mode = "all_pairs",
  quiet = FALSE
)

# use best fitting families to build smooth ROC curve

smooth_roc <- smooth(
  raw_roc,
  method = "fitdistr",
  density.controls = best_fits$density.controls,
  density.cases = best_fits$density.cases,
  start.controls = best_fits$start.controls,
  start.cases = best_fits$start.cases
)

# EXAMPLE USING COMPACT CALLS

# load packages and example blood biomarker data

library(pROC)
library(rocTools)
data(blood_data)

# build raw ROC curve

raw_roc <- roc(blood_data$Diagnosis, blood_data$NFL)

# find best fitting case and controls distribution families
# and build smooth ROC curve in single call

smooth_roc <- do.call(
  smooth,
  c(list(raw_roc, method = "fitdistr"), find.distr(raw_roc))
)
```

get.fits

Return distribution fits for smoothed ROC curves.

Description

Retrieves the distribution fits associated with smoothed ROC curves generated using the `smooth()` function of the ‘pROC’ package via the “fitdistr” method. A `smooth.roc` object is provided as

input, and the fitted case and control distribution families, parameters, and AIC values are returned. Can be used downstream of `find.distr()` and `smooth()` to inspect and evaluate the final distribution fits ultimately used to generate the smoothed ROC curve.

Usage

```
get.fits(smooth.roc)
```

Arguments

`smooth.roc` A `smooth.roc` object generated by `smooth()` using `method = "fitdistr"`.

Value

A nested list with the following elements:

<code>controls</code>	A list containing fit information for the fitted control distribution: <code>distribution</code> The distribution family. <code>parameters</code> A named numeric vector containing the distribution parameters. <code>aic</code> The AIC value.
<code>cases</code>	A list containing fit information for the fitted case distribution: <code>distribution</code> The distribution family. <code>parameters</code> A named numeric vector containing the distribution parameters. <code>aic</code> The AIC value.

Objects returned by `get.fits()` contain a custom print method that displays a human readable summary of the fitted distributions when printed at the console.

See Also

[pROC-package](#)
[pROC::roc\(\)](#)
[pROC::smooth\(\)](#)
[rocTools-package](#)
[find.distr\(\)](#)
[blood_data](#)

Examples

```
# load packages and example blood biomarker data

library(pROC)
library(rocTools)
data(blood_data)

# build raw ROC curve

raw_roc <- roc(
  blood_data,
  response = "Diagnosis",
```

```
    predictor = "NFL"
  )

  # find best fitting case and control distribution
  # families to use for ROC smoothing

  best_fits <- find.distr(raw_roc)

  # use best fitting families to build smooth ROC curve

  smooth_roc <- smooth(
    raw_roc,
    method = "fitdistr",
    density.controls = best_fits$density.controls,
    density.cases = best_fits$density.cases,
    start.controls = best_fits$start.controls,
    start.cases = best_fits$start.cases
  )

  # print final distribution fits used
  # to build the smoothed curve

  get.fits(smooth_roc)
```

sm.coords

Calculate diagnostic performance and cutoff estimates from smoothed ROC curves.

Description

Calculates population-level sensitivity, specificity, and diagnostic cutoff estimates from smoothed ROC curves generated using the `smooth()` function of the ‘pROC’ package via the “fitdist” method. A `smooth.roc` object is provided as input, and metrics can be calculated for the estimated diagnostic cutoff that maximizes the Youden index, or for the estimated diagnostic cutoff that achieves a user specified sensitivity or specificity value.

Usage

```
sm.coords(
  smooth.roc,
  type = c("best", "specificity", "sensitivity"),
  target.value = NULL,
  method = c("continuous", "grid"),
  quiet = FALSE
)
```

Arguments

smooth.roc	A smooth.roc object generated by smooth() using method = "fitdistr".
type	Character string specifying the criterion used to define the diagnostic cutoff to return estimates for. If "best", estimates for the cutoff that maximizes the Youden index are calculated and returned. If "specificity", estimates for the cutoff that achieves a target specificity value specified by target.value are calculated and returned. If "sensitivity", estimates for the cutoff that achieves a target sensitivity value specified by target.value are calculated and returned.
target.value	Numeric value ranging between 0 and 1 that specifies the target sensitivity or specificity value defining the cutoff to calculate estimates for when type is not "best".
method	Character string specifying the methodology used to generate estimates. If "continuous", values are estimated via continuous optimization of the smoothed ROC curve using the fitted case and control population probability density functions. If "grid", cutoff values are selected via optimization over the grid-discretized operating points of the smoothed ROC curve, matching the internal behavior of the native coords() function of the 'pROC' package. "continuous" provides finer resolution and potentially more accurate estimates; however, sensitivity and specificity values will not perfectly match those returned by the coords() function. "continuous" is default.
quiet	Specifies whether to silence messages generated during estimate calculation. Default value is FALSE.

Value

A named numeric vector containing the estimated diagnostic performance and cutoff values:

threshold	The estimated diagnostic cutoff value.
specificity	The estimated specificity at the cutoff value.
sensitivity	The estimated sensitivity at the cutoff value.

See Also

[pROC-package](#)
[pROC::roc\(\)](#)
[pROC::smooth\(\)](#)
[pROC::coords\(\)](#)
[rocTools-package](#)
[find.distr\(\)](#)
[blood_data](#)

Examples

```
# load packages and example blood biomarker data

library(pROC)
library(rocTools)
```

```
data(blood_data)

# build raw ROC curve

raw_roc <- roc(
  blood_data,
  response = "Diagnosis",
  predictor = "NFL"
)

# find best fitting case and control distribution
# families to use for ROC smoothing

best_fits <- find.distr(raw_roc)

# use best fitting families to build smooth ROC curve

smooth_roc <- smooth(
  raw_roc,
  method = "fitdistr",
  density.controls = best_fits$density.controls,
  density.cases = best_fits$density.cases,
  start.controls = best_fits$start.controls,
  start.cases = best_fits$start.cases
)

# calculate sensitivity, specificity, and cutoff
# estimates for the max Youden cutoff

sm.coords(smooth_roc, type = "best")

# calculate sensitivity, specificity, and cutoff estimates
# for the cutoff that yields 90% sensitivity

sm.coords(
  smooth_roc,
  type = "sensitivity",
  target.value = 0.9
)

# calculate sensitivity, specificity, and cutoff estimates
# for the cutoff that yields 75% specificity

sm.coords(
  smooth_roc,
  type = "specificity",
  target.value = 0.75
)
```

Index

`blood_data`, [3](#), [4](#), [4](#), [6](#), [8](#), [10](#)

`find.distr`, [5](#)

`find.distr()`, [2-5](#), [8](#), [10](#)

`get.fits`, [7](#)

`get.fits()`, [2-6](#), [8](#)

`rocTools-package`, [2](#), [6](#), [8](#), [10](#)

`sm.coords`, [9](#)

`sm.coords()`, [2-4](#), [6](#)