

Package ‘stapler’

July 23, 2025

Version 0.8.0

Title Simultaneous Truth and Performance Level Estimation

Description An implementation of Simultaneous Truth and Performance Level Estimation (STAPLE) <[doi:10.1109/TMI.2004.828354](https://doi.org/10.1109/TMI.2004.828354)>. This method is used when there are multiple raters for an object, typically an image, and this method fuses these ratings into one rating. It uses an expectation-maximization method to estimate this rating and the individual specificity/sensitivity for each rater.

License GPL-3

Imports matrixStats, RNifti

Suggests knitr, rmarkdown, covr, testthat, spelling

Encoding UTF-8

ByteCompile true

Type Package

Maintainer John Muschelli <muschellij2@gmail.com>

VignetteBuilder knitr

URL <https://github.com/muschellij2/stapler>

BugReports <https://github.com/muschellij2/stapler/issues>

RoxygenNote 7.3.2

Language en-US

NeedsCompilation no

Author John Muschelli [aut, cre]

Repository CRAN

Date/Publication 2025-04-01 16:40:02 UTC

Contents

staple	2
staple_bin_img	3

staple_bin_mat	5
staple_example_data	6
staple_multi_mat	7

Index	9
--------------	----------

staple	<i>Generic STAPLE Algorithm</i>
--------	---------------------------------

Description

Tries to do the correct STAPLE algorithm (binary/multi-class) for the type of input (array/matrix/list of images/filenames of images)

Usage

```
staple(x, ..., set_orient = FALSE)

## Default S3 method:
staple(x, ..., set_orient = FALSE)

## S3 method for class 'list'
staple(x, ..., set_orient = FALSE)

## S3 method for class 'character'
staple(x, ..., set_orient = FALSE)

## S3 method for class 'array'
staple(x, ..., set_orient = FALSE)
```

Arguments

x	a nxr matrix where there are n raters and r elements rated, a list of images, or a character vector. Note, readNifti is used for image filenames
...	Options for STAPLE, see staple_bin_mat and staple_multi_mat
set_orient	Should the orientation be set to the same if x is a set of images, including niftiImages.

Examples

```
n = 5
r = 1000
sens = c(0.8, 0.9, 0.8, 0.5, 0.8)
spec = c(0.9, 0.75, 0.99, 0.98, 0.92)
suppressWarnings(RNGversion("3.5.0"))
set.seed(20171120)
n_1 = 200
n_0 = r - n_1
```

```

truth = c(rep(0, n_0), rep(1, n_1))
pred_1 = rbinom(n = n, size = n_1, prob = sens)
pred_0 = rbinom(n = n, size = n_0, prob = spec)
pred_0 = sapply(pred_0, function(n) {
  sample(c(rep(0, n), rep(1, n_0 - n)))
})
pred_1 = sapply(pred_1, function(n) {
  sample(c(rep(1, n), rep(0, n_1 - n)))
})
pred = rbind(pred_0, pred_1)
true_sens = colMeans(pred[ truth == 1, ])
true_spec = colMeans(1-pred[ truth == 0, ])
x = t(pred)
staple_out = staple(x)
print(staple_out$sensitivity)
if (is.matrix(staple_out$sensitivity)) {
  staple_out$sensitivity = staple_out$sensitivity[, "1"]
}
testthat::expect_equal(staple_out$sensitivity,
c(0.781593858553476, 0.895868301462594,
0.760514086161722, 0.464483444340873,
0.765239314719065))
staple_out_prior = staple(x, prior = rep(0.5, r))

if (is.matrix(staple_out_prior$sensitivity)) {
  staple_out_prior$sensitivity = staple_out_prior$sensitivity[, "1"]
}
testthat::expect_equal(staple_out_prior$sensitivity,
c(0.683572080864211, 0.821556768891859,
0.619166852992802, 0.389409921992467, 0.67042085955546))

res_bin = staple_bin_mat(x, prior = rep(0.5, 1000))
testthat::expect_equal(staple_out_prior$sensitivity,
res_bin$sensitivity)
n = 5
r = 1000
x = lapply(seq(n), function(i) {
  x = rbinom(n = r, size = 1, prob = 0.5)
  array(x, dim = c(10,10, 10))
})
mat = sapply(x, c)
staple_out = staple_bin_img(x, set_orient = FALSE)
res_mat = staple(t(mat))
if (is.matrix(res_mat$sensitivity)) {
  res_mat$sensitivity = res_mat$sensitivity[, "1"]
}
testthat::expect_equal(staple_out$sensitivity, res_mat$sensitivity)

```

Description

Run STAPLE on a set of nifti images

Usage

```
staple_bin_img(x, set_orient = FALSE, verbose = TRUE, ...)
```

```
staple_multi_img(x, set_orient = FALSE, verbose = TRUE, ...)
```

Arguments

<code>x</code>	Character vector of filenames or list of arrays/images
<code>set_orient</code>	Should the orientation be set to the same if the images are niftiImages
<code>verbose</code>	print diagnostic messages
<code>...</code>	Additional arguments to staple_bin_mat

Value

A list similar to [staple_bin_mat](#), but has a resulting image

Examples

```
n = 5
r = 1000
x = lapply(seq(n), function(i) {
  x = rbinom(n = r, size = 1, prob = 0.5)
  array(x, dim = c(10,10, 10))
})
staple_out = staple_bin_img(x, set_orient = FALSE)
res = staple(x)
testthat::expect_equal(staple_out$sensitivity,
res$sensitivity)

x = lapply(x, RNifti::asNifti, internal = FALSE)
staple_img_out = staple_bin_img(x, set_orient = FALSE)
testthat::expect_equal(staple_out$sensitivity,
staple_img_out$sensitivity)
n = 5
r = 1000
x = lapply(seq(n), function(i) {
  x = rbinom(n = r, size = 5, prob = 0.5)
  array(x, dim = c(10,10, 10))
})
staple_out = staple_multi_img(x, set_orient = FALSE)
```

staple_bin_mat	<i>STAPLE on binary matrix</i>
----------------	--------------------------------

Description

STAPLE on binary matrix

Usage

```
staple_bin_mat(
  x,
  sens_init = 0.99999,
  spec_init = 0.99999,
  max_iter = 10000,
  tol = .Machine$double.eps,
  prior = "mean",
  verbose = TRUE,
  trace = 10,
  drop_all_same = FALSE
)
```

Arguments

<code>x</code>	a nxr matrix where there are n raters and r elements rated
<code>sens_init</code>	Initialize parameter for sensitivity (p)
<code>spec_init</code>	Initialize parameter for specificity (q)
<code>max_iter</code>	Maximum number of iterations to run
<code>tol</code>	Tolerance for convergence
<code>prior</code>	Either "mean" or a vector of prior probabilities,
<code>verbose</code>	print diagnostic messages
<code>trace</code>	Number for modulus to print out verbose iterations
<code>drop_all_same</code>	drop all records where they are all the same. DO NOT use in practice, only for validation of past results

Value

List of output sensitivities, specificities, and vector of probabilities

Examples

```
n = 5
r = 1000
sens = c(0.8, 0.9, 0.8, 0.5, 0.8)
spec = c(0.9, 0.75, 0.99, 0.98, 0.92)
suppressWarnings(RNGversion("3.5.0"))
```

```

set.seed(20171120)
n_1 = 200
n_0 = r - n_1
truth = c(rep(0, n_0), rep(1, n_1))
pred_1 = rbinom(n = n, size = n_1, prob = sens)
pred_0 = rbinom(n = n, size = n_0, prob = spec)
pred_0 = sapply(pred_0, function(n) {
  sample(c(rep(0, n), rep(1, n_0 - n)))
})
pred_1 = sapply(pred_1, function(n) {
  sample(c(rep(1, n), rep(0, n_1 - n)))
})
pred = rbind(pred_0, pred_1)
true_sens = colMeans(pred[ truth == 1, ])
true_spec = colMeans(1-pred[ truth == 0, ])
x = t(pred)
staple_out = staple_bin_mat(x)
testthat::expect_equal(staple_out$sensitivity,
c(0.781593858553476, 0.895868301462594,
0.760514086161722, 0.464483444340873,
0.765239314719065))
staple_out_prior = staple_bin_mat(x, prior = rep(0.5, r))
testthat::expect_equal(staple_out_prior$sensitivity,
c(0.683572080864211, 0.821556768891859,
0.619166852992802, 0.389409921992467, 0.67042085955546))

```

staple_example_data	<i>STAPLE Example Data</i>
---------------------	----------------------------

Description

STAPLE Example Data

Usage

```
staple_example_data()
```

Value

Character vector of filenames

Examples

```
staple_example_data()
```

staple_multi_mat	<i>STAPLE on Multi-class matrix</i>
------------------	-------------------------------------

Description

STAPLE on Multi-class matrix

Usage

```
staple_multi_mat(
  x,
  sens_init = 0.99999,
  spec_init = 0.99999,
  max_iter = 10000,
  tol = .Machine$double.eps,
  prior = "mean",
  verbose = TRUE,
  trace = 25,
  ties.method = c("first", "random", "last"),
  drop_all_same = FALSE
)
```

Arguments

<code>x</code>	a nxr matrix where there are n raters and r elements rated
<code>sens_init</code>	Initialize matrix for sensitivity (p)
<code>spec_init</code>	Initialize matrix for specificity (q)
<code>max_iter</code>	Maximum number of iterations to run
<code>tol</code>	Tolerance for convergence
<code>prior</code>	Either "mean" or a matrix of prior probabilities,
<code>verbose</code>	print diagnostic messages
<code>trace</code>	Number for modulus to print out verbose iterations
<code>ties.method</code>	Method passed to max.col for hard segmentation
<code>drop_all_same</code>	drop all records where they are all the same. DO NOT use in practice, only for validation of past results

Value

List of matrix output sensitivities, specificities, and matrix of probabilities

Examples

```
rm(list = ls())
x = matrix(rbinom(5000, size = 5, prob = 0.5), ncol = 1000)
sens_init = 0.99999
spec_init = 0.99999
max_iter = 10000
tol = .Machine$double.eps
prior = "mean"
verbose = TRUE
trace = 25
ties.method = "first"

res = staple_multi_mat(x)
xx = rbind(colMeans(x >= 2) > 0.5, colMeans(x >= 2) >= 0.5)
res = staple_multi_mat(xx, prior = rep(0.5, 1000))
res_bin = staple_bin_mat(xx, prior = rep(0.5, 1000))
testthat::expect_equal(res$sensitivity[, "1"], res_bin$sensitivity)
```

Index

`max.col`, [7](#)

`readNifti`, [2](#)

`staple`, [2](#)

`staple_bin_img`, [3](#)

`staple_bin_mat`, [2](#), [4](#), [5](#)

`staple_example_data`, [6](#)

`staple_multi_img(staple_bin_img)`, [3](#)

`staple_multi_mat`, [2](#), [7](#)