Package ‘KWCChangepoint’

May 7, 2026

Type Package

Title Robust Changepoint Detection for Functional and Multivariate
Data

Version 0.2.3

Description
Detect and test for changes in covariance structures of functional data, as well as changepoint detection for multivariate data more generally. Method for detecting non-stationarity in resting state functional Magnetic Resonance Imaging (fMRI) scans as seen in Ramsay, K., & Chenouri, S. (2025) <doi:10.1080/10485252.2025.2503891> is implemented in fmri_changepoints(). Also includes depth- and rank-based implementation of the wild binary segmentation algorithm for detecting multiple changepoints in multivariate data.

License MIT + file LICENSE

URL `https://github.com/adeeb99/KWCChangepoint`,
`https://adeeb99.github.io/KWCChangepoint/`

BugReports `https://github.com/adeeb99/KWCChangepoint/issues`

Depends R (>= 4.1)

Encoding UTF-8

Imports ddalpha, fda.usc, tibble, Rcpp (>= 1.0.12)

LinkingTo Rcpp, RcppArmadillo

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

SystemRequirements C++17

RoxygenNote 7.3.2

NeedsCompilation yes

Author Adeeb Rouhani [aut, cre, cph] (ORCID:
<`https://orcid.org/0009-0007-5294-2226`>),
Kelly Ramsay [aut] (ORCID: <`https://orcid.org/0000-0001-5615-2052`>)

Maintainer Adeeb Rouhani <`adeeb.rouhani@gmail.com`>

Repository CRAN

Date/Publication 2025-11-26 20:50:02 UTC

Contents

`KWCChangepoint-package`

KWCChangepoint: Robust Changepoint Detection for Functional and Multivariate Data

---

Description

Detect and test for changes in covariance structures of functional data, as well as changepoint detection for multivariate data more generally. Method for detecting non-stationarity in resting state functional Magnetic Resonance Imaging (fMRI) scans as seen in Ramsay, K., & Chenouri, S. (2025) doi:10.1080/10485252.2025.2503891 is implemented in fmri_changepoints(). Also includes depth- and rank-based implementation of the wild binary segmentation algorithm for detecting multiple changepoints in multivariate data.

Links

- GitHub: `https://github.com/adeeb99/KWCChangepoint`
- Bug reports: `https://github.com/adeeb99/KWCChangepoint/issues`

Author(s)

Maintainer: Adeeb Rouhani `<adeeb.rouhani@gmail.com>` (ORCID) [copyright holder]

Authors:

- Kelly Ramsay (ORCID)

See Also

Useful links:

- `https://github.com/adeeb99/KWCChangepoint`
- `https://adeeb99.github.io/KWCChangepoint/`
- Report bugs at `https://github.com/adeeb99/KWCChangepoint/issues`

`amoc_test` *Conduct an AMOC hypothesis test*

Description

Conduct an at-most one changepoint hypothesis test for changes in the covariance operator of functional data based on the FKWC (functional Kruskal–Wallis covariance changepoint) procedures outlined by Ramsay and Chenouri (2025).

Usage

```
amoc_test(data, ranks = NULL, depth = c("RPD", "FM", "LTR", "FMd", "RPDd"))
```

Arguments

`data`	Data in `matrix` or `data.frame` form, where each row is an observation and each column is a dimension.
`ranks`	Optional if data is already ranked.
`depth`	Depth function of choice.

Value

A list consisting of:

- `$changepoint` : Index of the estimated changepoint.
- `$pvalue` : The p-value based on the null distribution.
- `$method` : A `string "AMOC test (KWCChangepoint)"`

Note

The options for the `depth` argument are as follows:

- `RPD`: Random projection depth
- `FM`: Frainman-Muniz depth
- `LTR`: L^2-root depth, most suitable for detecting changes in the norm
- `FMd`: Frainman-Muniz depth of the data and its first order derivative
- `RPDd`: Random projection depth of the data and its first order derivative

  The depth arguments that incorporate the first order derivative (which is approximated using fda.usc::fdata.deriv) result in a more robust detection of changes in the covariance structure (Ramsay and Chenouri, 2025).

References

Ramsay, K., & Chenouri, S. (2025). Robust changepoint detection in the variability of multivariate functional data. Journal of Nonparametric Statistics. https://doi.org/10.1080/10485252.2025.2503891

Examples

```
set.seed(11)
test_data <- rbind(replicate(3,rnorm(200,1,1)), #before changepoint
                   replicate(3,rnorm(200,1,5))) #after changepoint

amoc_test(test_data)
```

---

dwbs *Find changepoints using depth-based wild binary segmentation*

Description

Detect multiple changepoints in multivariate data using the depth-based wild binary segmentation algorithm (Ramsay and Chenouri, 2023).

Usage

```
dwbs(
  data,
  numInt = 10,
  thresh = 1.3584,
  alpha = 1,
  depth = c("spat", "hs", "mahal", "mahal75")
)
```

Arguments

`data`	Data in `matrix` or `data.frame` form, where each row is an observation and each column is a dimension.
`numInt`	Number of intervals to be generated.
`thresh`	Numeric scalar; detection threshold. Larger values make detection more conservative.
`alpha`	Set as 1 by default, applying a standard SIC penalty. Set to a number larger than 1 for a strengthened SIC.
`depth`	Depth function.

Value

A list consisting of:

- `$changepoints` : Indicies of the change-points detected; will return `integer(0)` if no changepoints are detected.
- `$method` : A `string` `"DWBS"`

Note

The options for the `depth` argument are as follows:

- `spat`: Spatial depth
- `hs`: Halfspace depth
- `mahal`: Mahalanobis depth
- `mahal75`: Mahalanobis depth based on re-weighted Minimum Covariance Determinant with 25% breakdown.

References

Fryzlewicz, Piotr. "Wild Binary Segmentation for Multiple Change-Point Detection." The Annals of Statistics 42, no. 6 (2014). https://doi.org/10.1214/14-AOS1245.

Killick, R., P. Fearnhead, and I. A. Eckley. "Optimal Detection of Changepoints With a Linear Computational Cost." Journal of the American Statistical Association 107, no. 500 (2012): 1590–98. https://doi.org/10.1080/01621459.2012.737745.

Ramsay, K., & Chenouri, S. (2023). Robust nonparametric multiple changepoint detection for multivariate variability. Econometrics and Statistics. https://doi.org/10.1016/j.ecosta.2023.09.001

Examples

```
set.seed(11)
exdata <- rbind(replicate(3,rnorm(200)),
                replicate(3,rnorm(200,10)),
                replicate(3,rnorm(200,0.2)))
dwbs(data = exdata)

# Increasing `numInt` will result in more accurate detection
dwbs(data = exdata, numInt = 100)
```

`epidemic_test` *Test for an epidemic period in data*

Description

Test for a temporary change in the covariance operator of functional data using the FKWC (functional Kruskal–Wallis covariance changepoint) procedures outlined by Ramsay and Chenouri (2025).

Usage

```
epidemic_test(data, ranks = NULL, depth = c("RPD", "FM", "LTR", "FMd", "RPDd"))
```

Arguments

`data`	Data in `matrix` or `data.frame` form, where each row is an observation and each column is a dimension.
`ranks`	Optional if data is already ranked.
`depth`	Depth function of choice.

Value

A list consisting of:

- `$changepoints` : Indices of the estimated start and end points for the epidemic period.
- `$pvalue` : The p-value based on the null distribution.
- `$method` : A `string "Epidemic test (KWCChangepoint)"`

Note

The options for the `depth` argument are as follows:

- `RPD`: Random projection depth
- `FM`: Frainman-Muniz depth
- `LTR`: L^2-root depth, most suitable for detecting changes in the norm
- `FMd`: Frainman-Muniz depth of the data and its first order derivative
- `RPDd`: Random projection depth of the data and its first order derivative

  The depth arguments that incorporate the first order derivative (which is approximated using fda.usc::fdata.deriv) result in a more robust detection of changes in the covariance structure (Ramsay and Chenouri, 2025).

References

Ramsay, K., & Chenouri, S. (2025). Robust changepoint detection in the variability of multivariate functional data. Journal of Nonparametric Statistics. https://doi.org/10.1080/10485252.2025.2503891

Examples

```
set.seed(11)
epi_test <- rbind(replicate(3,rnorm(200)),
                  replicate(3,rnorm(200,10)),
                  replicate(3,rnorm(200,0.2)))

epidemic_test(epi_test)
```

`fkwc` *Detect changepoints in functional data*

Description

More specifically, `fkwc()` uses the functional Kruskal-Wallis tests for covariance changepoint algorithm (FKWC) to detect changes in the covariance operator.

Usage

```
fkwc(data, depth = c("RPD", "FM", "LTR", "FMd", "RPDd"), k = 0.25)
```

Arguments

`data`	Functional data in `matrix` or `data.frame` form, where each row is an observation/function and the columns are the grid.
`depth`	Depth function of choice.
`k`	Penalty constant passed to pruned exact linear time algorithm.

Value

A list consisting of:

- `$changepoints` : Indices of the changepoints detected; will return `integer(0)` if no changepoints are detected.
- `$method` : A `string` `"FKWC"`

Note

The options for the `depth` argument are as follows:

- `RPD`: Random projection depth, which generally performs best
- `FM`: Frainman-Muniz depth
- `LTR`: L^2-root depth, most suitable for detecting changes in the norm
- `FMd`: Frainman-Muniz depth of the data and its first order derivative
- `RPDd`: Random projection depth of the data and its first order derivative

  The depth arguments that incorporate the first order derivative (which is approximated using fda.usc::fdata.deriv) result in a more robust detection of changes in the covariance structure (Ramsay and Chenouri, 2025).

The penalty is of the form

$$3.74 + k\sqrt{n}$$

where n is the number of observations. In the case that there is potentially correlated observations, the parameter could be set to $k = 1$. More information could be found in the reference.

References

Killick, R., P. Fearnhead, and I. A. Eckley. “Optimal Detection of Changepoints With a Linear Computational Cost.” Journal of the American Statistical Association 107, no. 500 (2012): 1590–98. https://doi.org/10.1080/01621459.2012.737745.

Ramsay, K., & Chenouri, S. (2025). Robust changepoint detection in the variability of multivariate functional data. Journal of Nonparametric Statistics. https://doi.org/10.1080/10485252.2025.2503891

Examples

```
set.seed(2)
# Generating 80 observations, with a changepoint (in our case a change in
# kernel) at observation 40
n  <- 80
k0 <- 40
T  <- 30
t  <- seq(0, 1, length.out = T)


# Both kernels K1 and K2 are Gaussian (or squared exponential) kernels but
# with different lengthscale values, and thus we hope to detect it.
K_se <- function(s, t, ell) exp(- ( (s - t)^2 ) / (2 * ell^2))
K1   <- outer(t, t, function(a,b) K_se(a,b, ell = 0.20))
K2   <- outer(t, t, function(a,b) K_se(a,b, ell = 0.07))

L1 <- chol(K1 + 1e-8 * diag(T))
L2 <- chol(K2 + 1e-8 * diag(T))

Z1 <- matrix(rnorm(k0 * T),      k0,      T)
Z2 <- matrix(rnorm((n-k0) * T),  n - k0,  T)

# We finally have an 80 x 30 matrix where the rows are the observations and
# the columns are the grid points.
X  <- rbind(Z1 %*% t(L1), Z2 %*% t(L2))

fkwc(X)
```

`fkwc_multisample` *Multisample hypothesis test for difference in covariance operators*

Description

Executes a multisample hypothesis test for differences in covariance operators using functional Kruskal–Wallis tests for covariance (FKWC) as outlined by Ramsay and Chenouri (2024). The function requires the first order derivative of the functional data in order to better detect changes.

Usage

```
fkwc_multisample(data, derivs, g, p = 20)
```

Arguments

`data`	Functional data in `matrix` or `data.frame` form, where each row is an observation/function and the columns are the grid.
`derivs`	First order derivative of the functional data in `matrix` or `data.frame` form.
`g`	A `factor` object that indicates which sample each row of data belongs to.
`p`	Number of random projections to be generated in order to compute random projection depths of the data.

Value

A list consisting of:

- `$statistic` : The observed test statistic.
- `$pvalue` : The p-value based on the null distribution.
- `$method` : A `string` `"FKWC"`

References

Ramsay, K., & Chenouri, S. (2024). Robust nonparametric hypothesis tests for differences in the covariance structure of functional data. Canadian Journal of Statistics, 52 (1), 43–78. https://doi.org/10.1002/cjs.11767

See Also

`fda.usc::fdata.deriv()`: for approximating the first order derivative if unavailable.

`fkwc_posthoc()`: for a post-hoc version of this test

Examples

```
set.seed(111)
t <- seq(0, 1, length.out = 200)

### Generating three sets of Brownian curves with different kernels, each
### kernel generating 20 observations
# Brownian process 1
fd1 <- fda.usc::rproc2fdata(n = 20, t = t, sigma = "brownian",
                            par.list = list(scale = 10, theta = 1))
fd1_d <- fda.usc::fdata.deriv(fd1)

# Brownian process 2
fd2 <- fda.usc::rproc2fdata(n = 20, t = t, sigma = "brownian",
                            par.list = list(scale = 1, theta = 1))
fd2_d <- fda.usc::fdata.deriv(fd2)

# Brownian process 3
fd3 <- fda.usc::rproc2fdata(n = 20, t = t, sigma = "brownian",
                            par.list = list(scale = 1, theta = 5))
fd3_d <- fda.usc::fdata.deriv(fd3)

# Functional data in one matrix and first order derivatives in another matrix
funcdata <- rbind(fd1$data, fd2$data, fd3$data)
```

```
funcderivs <- rbind(fd1_d$data, fd2_d$data, fd3_d$data)

fkwc_multisample(data = funcdata,
                 derivs = funcderivs,
                 g = factor(rep(1:3, each = 20)),
                 p = 1000)
```

`fkwc_posthoc` *Post-hoc hypothesis test for difference in covariance operators.*

Description

This function is post-hoc, pairwise test version of `fkwc_multisample()`

Usage

```
fkwc_posthoc(data, derivs, g, p = 20)
```

Arguments

`data`	Functional data in `matrix` or `data.frame` form, where each row is an observation/function and the columns are the grid.
`derivs`	First order derivative of the functional data in `matrix` or `data.frame` form.
`g`	A `factor` object that indicates which sample each row of data belongs to.
`p`	Number of random projections to be generated in order to compute random projection depths of the data.

Value

A matrix of p-values for each pairwise comparison with a Šidák correction applied.

References

Ramsay, K., & Chenouri, S. (2024). Robust nonparametric hypothesis tests for differences in the covariance structure of functional data. Canadian Journal of Statistics, 52 (1), 43–78. https://doi.org/10.1002/cjs.11767

See Also

fda.usc::fdata.deriv: for approximating the first order derivative if unavailable.

Examples

```
set.seed(111)
t <- seq(0, 1, length.out = 200)

### Generating three sets of brownian curves with different kernels
# Brownian process 1
fd1 <- fda.usc::rproc2fdata(n = 20, t = t, sigma = "brownian",
                            par.list = list(scale = 10, theta = 1))
fd1_d <- fda.usc::fdata.deriv(fd1)

# Brownian process 2
fd2 <- fda.usc::rproc2fdata(n = 20, t = t, sigma = "brownian",
                            par.list = list(scale = 1, theta = 1))
fd2_d <- fda.usc::fdata.deriv(fd2)

# Brownian process 3
fd3 <- fda.usc::rproc2fdata(n = 20, t = t, sigma = "brownian",
                            par.list = list(scale = 1, theta = 5))
fd3_d <- fda.usc::fdata.deriv(fd3)

# Functional data in one matrix and first order derivatives in another matrix
funcdata <- rbind(fd1$data, fd2$data, fd3$data)
funcderivs <- rbind(fd1_d$data, fd2_d$data, fd3_d$data)

fkwc_posthoc(data = funcdata,
             derivs = funcderivs,
             g = factor(rep(1:3, each = 20)),
             p = 1000)
```

`fmri_changepoints` *Detect changepoints in a resting state fMRI scan*

Description

Functional magnetic resonance imaging scans are expected to be stationary after being pre-processed. This function attempts to find potential changepoints using the findings of Ramsay and Chenouri (2025).

Usage

```
fmri_changepoints(data, p = 100, k = 0.3)
```

Arguments

`data`	A four dimensional array, where the fourth dimension is time.
`p`	Number of random vector projections, set to 100 by default.
`k`	Penalty constant passed to pruned exact linear time algorithm.

Value

A list consisting of:

- `$changepoints` : Indices of the change-points detected; will return `integer(0)` if no change-points are detected.
- `$method` : A string `"fMRI changepoints (KWCChangepoint)"`

Note

The penalty is of the form

$$3.74 + k\sqrt{n}$$

where n is the number of observations. In the case that there is potentially correlated observations, the parameter could be set to $k = 1$. More information could be found in the reference.

The example in this document is a simple "toy example", as good fMRI data simulation requires more dependencies. For generating fMRI data, see `neuRosim::simVOLfmri()`, `neuRosim::simTSrestingstate()`.

References

Ramsay, K., & Chenouri, S. (2025). Robust changepoint detection in the variability of multivariate functional data. Journal of Nonparametric Statistics. https://doi.org/10.1080/10485252.2025.2503891

Examples

```
# In order to replicate how a changepoint would appear in a resting-state
# fMRI scan in a manner that is not computationally expensive, this example
# constructs an image of a 3D ball taken at 12 time stamps. The noise, and
# therefore the covariance function, changes at time stamp 6.
x_dim <- 24
y_dim <- 24
z_dim <- 10
time_dim <- 12
image_array <- array(0, dim = c(x_dim, y_dim, z_dim, time_dim))

center <- c(x_dim / 2, y_dim / 2, z_dim / 2)
radius <- min(x_dim, y_dim, z_dim) / 4

set.seed(42)

for (t in 1:time_dim) {
  for (x in 1:x_dim) {
    for (y in 1:y_dim) {
      for (z in 1:z_dim) {
       dist_from_center <- sqrt((x - center[1])^2 + (y - center[2])^2 + (z - center[3])^2)
        if (dist_from_center <= radius) {
          # Adding noise with increasing variability at timestamp 6
          if (t <= 6) {
            noise <- rnorm(1, mean = 0, sd = 0.1)  # Low variability noise
          } else {
            noise <- rnorm(1, mean = 0, sd = 2)  # High variability noise
          }
```

```
          image_array[x, y, z, t] <- noise
        } else {
          # Add lower intensity noise outside the ball
          image_array[x, y, z, t] <- rnorm(1, mean = 0, sd = 0.005)
        }
      }
    }
  }
}
fmri_changepoints(image_array, k = 0.1, p = 10)
```

`mkwp` *Find changepoints in multivariate data*

Description

The `mkwp()` function detects changepoints in multivariate data using multivariate Kruskal-Wallis PELT (MKWP) algorithm developed by Ramsay and Chenouri (2023).

Usage

```
mkwp(data, depth = c("spat", "mahal", "mahal75", "hs"), k = 0.2)
```

Arguments

`data`	Data in `matrix` or `data.frame` form, where each row is an observation and each column is a dimension.
`depth`	Depth function.
`k`	Penalty constant passed to pruned exact linear time algorithm.

Value

A list consisting of:

- `$changepoints` : Indices of the changepoints detected; will return `integer(0)` if no changepoints are detected.
- `$method` : A `string` `"Multivariate Kruskal-Wallis PELT (MKWP)"`

Note

The options for the `depth` argument are as follows:

- `spat`: Spatial depth
- `hs`: Halfspace depth
- `mahal`: Mahalanobis depth

- `mahal75`: Mahalanobis depth based on re-weighted Minimum Covariance Determinant with 25% breakdown.

  Spatial depth is the default choice, as it computationally quicker than the other depths for larger data while giving similar result to other depths.

The penalty is of the form

$$3.74 + k\sqrt{n}$$

where n is the number of observations. In the case that there is potentially correlated observations, the parameter could be set to $k = 1$. More information could be found in the reference.

References

Killick, R., P. Fearnhead, and I. A. Eckley. "Optimal Detection of Changepoints With a Linear Computational Cost." Journal of the American Statistical Association 107, no. 500 (2012): 1590–98. https://doi.org/10.1080/01621459.2012.737745.

Ramsay, K., & Chenouri, S. (2023). Robust nonparametric multiple changepoint detection for multivariate variability. Econometrics and Statistics. https://doi.org/10.1016/j.ecosta.2023.09.001

Examples

```
set.seed(111)
multi_data <-rbind(replicate(3,rnorm(200)),
                   replicate(3,rnorm(200,10)),
                   replicate(3,rnorm(200,0.2)))
mkwp(multi_data)
```

`uni_mean` *Find mean changes in a univariate sequence*

Description

The `uni_mean()` function ranks the observations from smallest to largest, then applies the pruned exact linear time algorithm with the penalty parameter `beta` to detect changepoints.

Usage

```
uni_mean(data, beta = 10)
```

Arguments

`data`	A vector or one-dimensional array.
`beta`	Numeric penalty constant passed to pruned exact linear time algorithm.

Value

A list consisting of:

- `$changepoints` : Indices of the changepoints detected; will return `integer(0)` if no changepoints are detected.
- `$method` : A `string "Univariate Changepoint in Mean (FKWC)"`

References

Killick, R., P. Fearnhead, and I. A. Eckley. “Optimal Detection of Changepoints With a Linear Computational Cost.” Journal of the American Statistical Association 107, no. 500 (2012): 1590–98. https://doi.org/10.1080/01621459.2012.737745.

Examples

```
set.seed(11)
mean_test <- c(rnorm(100, mean = 0), # before change in mean
               rnorm(100, mean = 5)) # after change in mean
uni_mean(mean_test)
```

`uni_scale` *Find scale changes in a univariate sequence*

Description

The `uni_scale()` function ranks the observations based on their distance from the mean, then applies the pruned exact linear time algorithm with the penalty parameter `beta` to detect changepoints.

Usage

```
uni_scale(data, beta = 10)
```

Arguments

`data`	A vector or one-dimensional array.
`beta`	Numeric penalty constant passed to pruned exact linear time algorithm, 10 by default.

Value

A list consisting of:

- `$changepoints` : Indices of the changepoints detected; will return `integer(0)` if no changepoints are detected.
- `$method` : A `string "Univariate Changepoint in Scale (KWCChangepoint)"`

References

Killick, R., P. Fearnhead, and I. A. Eckley. “Optimal Detection of Changepoints With a Linear Computational Cost.” Journal of the American Statistical Association 107, no. 500 (2012): 1590–98. https://doi.org/10.1080/01621459.2012.737745.

Examples

```
set.seed(11)
scale_test <- c(rnorm(100, sd=5), # before change in sale
                rnorm(100, sd=1)) # after change in scale
uni_scale(scale_test)
```

Index