Package: FES (via r-universe)

October 21, 2026

Type Package

Title Fisher Exact Scanning for Dependency

Version 1.0

Description Implements Fisher exact scanning (FES), a multiscale test of dependence for continuous or discrete bivariate data. The method scans nested binary partitions using Fisher's exact tests and combines evidence across windows and resolutions with Sidak, Bonferroni, and meta-analysis corrections. The methodology is described in Ma and Mao (2019) <doi:10.1080/01621459.2017.1397522>.

License GPL (>= 3)

URL https://github.com/mastatlab/fes

BugReports https://github.com/mastatlab/fes/issues

Encoding UTF-8

Depends R (>= 3.6.0)

Imports graphics, grDevices, methods, stats

Suggests BayesFactor, testthat (>= 3.0.0)

Config/testthat/edition 3

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.0.0

Repository https://mastatlab.r-universe.dev

Date/Publication 2026-07-14 12:28:38 UTC

RemoteUrl https://github.com/mastatlab/fes

RemoteRef HEAD

RemoteSha 489cc41a7664e4d3d04b711fa75cf6ab5dca8a0a

Contents

fes 2
fes_details 4

Index **5**

`fes` *Fisher Exact Scanning for Dependency*

Description

Tests dependence between two variables by applying Fisher's exact test to multiscale binary partitions of their joint sample space. Evidence is combined within strata, across resolutions, and globally as described by Ma and Mao (2019).

Usage

```
fes(
  x,
  y,
  K1 = 7,
  K2 = 7,
  fixed.M = FALSE,
  M = NULL,
  rank.transform = TRUE,
  plot.it = FALSE,
  plot.sidak = FALSE,
  plot.scale = FALSE,
  output.dir = NULL,
  mid.p = TRUE,
  compute.bf = FALSE,
  verbose = FALSE,
  return.details = FALSE
)
```

Arguments

`x, y`	Numeric vectors of equal, positive length. Missing and infinite values are not allowed.
`K1, K2`	Positive integers giving the maximum binary partition depths for `x` and `y`, respectively.
`fixed.M`	Logical. If `TRUE`, use the highest active resolution in the requested range for the global correction. If `FALSE`, use the number of active resolutions.
`M`	Optional nonnegative integer giving the maximum resolution to scan. The default scans all resolutions from zero through `K1 + K2 - 2`.
`rank.transform`	Logical. If `TRUE`, transform each margin using its empirical distribution function. Otherwise, linearly rescale each margin to `[0, 1)`.
`plot.it`	Logical. Draw the tested windows when `TRUE`. Each panel uses a shared color scale and overlays a representative sample of transformed observations.
`plot.sidak`	Logical. Plot globally Sidak-adjusted rather than raw window p-values.

plot.scale	Logical. Add a color scale beside every window plot.
output.dir	NULL or a nonempty character string giving a directory for PDF plots. When supplied with plot.it = TRUE, the directory is created if needed and plots are written there instead of to the active graphics device.
mid.p	Logical. Use mid-p values for the individual Fisher tests.
compute.bf	Logical. Compute and plot log Bayes factors using the suggested BayesFactor package. When plot.sidak = TRUE, p-values are plotted instead.
verbose	Logical. Print per-resolution and global summaries.
return.details	Logical. If TRUE, return the global p-values together with a table describing every tested window. The table contains window bounds, cell counts, multiplicity-adjusted p-values, and a signed local log-odds ratio that helps characterize the dependency.

Details

A window is tested only when it contains more than 25 observations and both of its row and column margins exceed 10 observations. Consequently, small samples or fine resolutions can contain no testable windows; those resolutions contribute neutral p-values. The returned name Sidak denotes the Sidak multiplicity correction.

Detailed window bounds are on the transformed [0, 1] scale. Counts n11, n12, n21, and n22 follow the low/high splits of X by the low/high splits of Y. The reported local log-odds ratio uses a 0.5 continuity correction; its sign describes local direction, while adjusted_p_value determines window significance across the multiscale scan.

Value

By default, a named numeric vector containing the global Sidak, Bonferroni, meta-analysis, and unadjusted minimum p-values. With return.details = TRUE, an object of class fes_details containing p.value, windows, significant.windows, and the rank-transformed data.

References

Ma, L. and Mao, J. (2019). Fisher exact scanning for dependency. *Journal of the American Statistical Association*, 114(525), 245–258. doi:10.1080/01621459.2017.1397522

Examples

```
set.seed(12345)
x <- runif(200)
y <- x + rnorm(200, sd = 0.15)
fes(x, y, K1 = 4, K2 = 4)

# Learn where a non-monotone dependency changes direction.
y_curve <- (x - 0.5)^2 + rnorm(200, sd = 0.025)
structure <- fes(x, y_curve, K1 = 3, K2 = 3, return.details = TRUE)
head(structure$significant.windows)
plot(structure, max.windows = 8)

# Run demo("quick_start", package = "FES") for a visual walkthrough.
```

`fes_details` *Detailed FES Results*

Description

Methods for inspecting and visualizing the window-level output returned by `fes()` when `return.details = TRUE`. Positive local log-odds ratios indicate concordant movement within a window; negative values indicate discordant movement. The plot overlays the strongest significant windows in orange or blue, making changes in location, scale, and direction visible.

Usage

```
## S3 method for class 'fes_details'
print(x, ...)

## S3 method for class 'fes_details'
plot(
  x,
  y = NULL,
  alpha = 0.05,
  max.windows = 12L,
  stratum = "best",
  main = "FES dependency structure",
  ...
)
```

Arguments

`x`	An object of class `fes_details`.
`...`	Additional arguments passed to `graphics::plot.default()`.
`y`	Ignored.
`alpha`	Multiplicity-adjusted significance level used to select windows.
`max.windows`	Maximum number of the strongest windows to draw.
`stratum`	Which partition stratum to draw. The default, `"best"`, chooses the `(k1, k2)` stratum containing the most significant windows, with ties favoring the finer stratum. Supply a numeric pair `c(k1, k2)` to choose a stratum, or `"all"` to overlay windows across strata.
`main`	Plot title.

Value

`print()` and `plot()` return `x` invisibly.

Index

fes, 2
fes(), *4*
fes_details, 4

graphics::plot.default(), *4*

plot.fes_details *(*fes_details*)*, 4
print.fes_details *(*fes_details*)*, 4