Package 'sffdr'

December 2, 2024

Type Package

Title Surrogate Functional False Discovery Rates for Genome-Wide Association Studies

Version 1.0.0

Maintainer Andrew Bass <ab3105@cam.ac.uk>

Description Pleiotropy-informed significance analysis of genome-wide association studies (GWAS) with surrogate functional false discovery rates (sfFDR). The sfFDR framework adapts the fFDR to leverage informative data from multiple sets of GWAS summary statistics to increase power in study while accommodating for linkage disequilibrium. sfFDR provides estimates of key FDR quantities in a significance analysis such as the functional local FDR and q-value, and uses these estimates to derive a functional p-value for type I error rate control and a functional local Bayes' factor for post-GWAS analyses (e.g., fine mapping and colocalization). The sfFDR framework is described in Bass and Wallace (2024) <doi:10.1101/2024.09.24.24314276>.

URL https://github.com/ajbass/sffdr

License LGPL

Encoding UTF-8

VignetteBuilder knitr

LazyData true

Depends R (>= 3.5.0)

Imports locfit, splines, dplyr, ggplot2 (>= 3.5.1), patchwork (>= 1.3.0), gam, qvalue, tibble, tidyr, Rcpp

Suggests testthat (>= 3.0.0), knitr, rmarkdown

LinkingTo Rcpp

RoxygenNote 7.3.1

NeedsCompilation yes

Author Andrew Bass [aut, cre], Chris Wallace [aut]

Repository CRAN

Date/Publication 2024-12-02 12:30:08 UTC

Contents

bmi 2
ffinemap 3
fpi0est 4
fpvalues 6
kernelEstimator 6
pi0_model 7
plot.sffdr 9
sffdr 10

Index **13**

`bmi` *Subset of p-values from the UK Biobank analysis*

Description

The summary level data is a subset of independent SNPs from the UK Biobank where we performed a GWAS of body mass index (BMI), body fat percentage (BFP), cholesterol, and triglycerides. Note that BFP, cholesterol and triglycerides are conditioning traits and were calculated using a separate set of individuals than BMI. See manuscript for details.

Usage

```
data(bmi)
```

Value

A list called `sumstats` containing:

`bmi`	Vector of 10,000 p-values for BMI.
`bfp`	Vector of 10,000 p-values for BFP.
`cho`	Vector of 10,000 p-values for cholesterol.
`tri`	Vector of 10,000 p-values for triglycerides.

See Also

`sffdr`

Examples

```
# import data
data(bmi)

# separate main p-values and conditioning p-values
p <- sumstats$bmi
z <- as.matrix(sumstats[, -1])
```

```
# apply pi0_model to create model
knots <- c(0.005, 0.01, 0.025, 0.05, 0.1)
fmod <- pi0_model(z, knots = knots)

# import data
data(bmi)

# separate main p-values and conditioning p-values
p <- sumstats$bmi
z <- as.matrix(sumstats[, -1])

# apply pi0_model to create model
knots <- c(0.005, 0.01, 0.025, 0.05, 0.1)
fmod <- pi0_model(z, knots = knots)

# estimate functional pi0
fpi0_out <- fpi0est(p, z = fmod$zt, pi0_model = fmod$fmod)
fpi0 <- fpi0_out$fpi0

# apply sffdr
# Note all tests are independent see 'indep_snps' argument
# The very small p-values, set epsilon to min of p
sffdr_out <- sffdr(p, fpi0, epsilon = min(p))

# Plot significance results
plot(sffdr_out, rng = c(0, 5e-4))

# Functional P-values, Q-values, and local FDR
fp <- sffdr_out$fpvalues
fq <- sffdr_out$fqvalues
flfdr <- sffdr_out$flfdr
```

`ffinemap` *Functional fine mapping*

Description

Perform functional fine mapping with a set of functional local FDRs in a region of interest (assuming a single causal variant).

Usage

```
ffinemap(flfdr, fpi0)
```

Arguments

`flfdr`	A vector of functional local FDRs of a region of interest
`fpi0`	An estimate of the function proportion of null tests

Value

A list of object type "sffdr" containing:

BF	The functional local Bayes' factors.
PP	Posterior probability of a SNP being causal.

`fpi0est` *Estimate the functional proportion of null tests*

Description

he function `fpi0est` estimates the functional proportion of null tests given a set of informative variables.

Usage

```
fpi0est(
  p,
  z,
  pi0_model,
  indep_snps = NULL,
  lambda = seq(0.05, 0.9, 0.05),
  method = "gam",
  maxit = 1000,
  pi0.method.control = NULL,
  ...
)
```

Arguments

p	A vector of p-values.
z	A vector of informative variables
pi0_model	Model formula corresponding to z for the functional proportion of truly null tests.
indep_snps	A boolean vector (same size as p) specifying the set of independent tests. Default is NULL and all tests are treated independently.
lambda	A vector of values between [0,1] to estimate the functional proportion of truly null tests.
method	Either the "gam" (generalized additive model) or "glm" (generalized linear models) approach. Default is "gam".
maxit	The maximum number of iterations for "glm" approach. Default is 1000.
pi0.method.control	A user specified set of parameters for convergence for either "gam" or "glm". Default is NULL. See `gam.control` or `glm.control`.
...	Additional arguments passed to `gam` or `glm`.

Details

This code extends the function from the fFDR package to handle multiple informative variables and linkage disequilibrium.

Value

A list of object type "fpi0" containing:

`fpi0`	A table containing the functional proportion of truly null tests.
`tableLambda`	Functional proportion of null tests at the lambda values
`MISE`	MISE values.
`lambda.hat`	The chosen lambda value.

Author(s)

Andrew J. Bass, David G. Robinson (author of original function)

See Also

`sffdr`, `plot.sffdr`

Examples

```
# import data
data(bmi)

# separate main p-values and conditioning p-values
p <- sumstats$bmi
z <- as.matrix(sumstats[, -1])

# apply pi0_model to create model
knots <- c(0.005, 0.01, 0.025, 0.05, 0.1)
fmod <- pi0_model(z, knots = knots)

# Estimate functional pi0
fpi0_out <- fpi0est(p, z = fmod$zt, pi0_model = fmod$fmod)
fpi0 <- fpi0_out$fpi0

# See relationship of BFP/cholesterol/triglycerides and fpi0
plot(fmod$zt$bfp, fpi0)
plot(fmod$zt$cholesterol, fpi0)
plot(fmod$zt$triglycerides, fpi0)
```

fpvalues *Functional p-values*

Description

Calculate functional p-values from functional local FDRs. Internal use.

Usage

```
fpvalues(lfdr, p = NULL)
```

Arguments

lfdr	A vector of functional local FDRs of a region
p	A vector of p-values. Default is NULL.

Value

A list of object type "sffdr" containing:

fp	Functional p-values.
fq	Functional q-values.

kernelEstimator *Estimate a density on the unit interval or unit square via local regression*

Description

This function is adapted from the fFDR package.

Usage

```
kernelEstimator(
  x,
  transformation = "probit",
  eval.points = x,
  subsample = 1e+07,
  epsilon = 1e-15,
  epsilon.max = 0.999,
  maxk = 10000,
  trim = 1e-15,
  nn = NULL,
  ...
)
```

Arguments

x	Either a vector or a 2-column matrix
transformation	Either probit (default), complementary log-log, or identity (not recommended)
eval.points	Points at which to evaluate the estimate, default x
subsample	Number of points that are randomly subsampled for computing the fit; useful for computational efficiency and for ensuring the density estimation does not run out of memory. NULL means no the fit is performed on all points
epsilon	How close values are allowed to come to 0
epsilon.max	How close values are allowed to come to 1
maxk	maxk argument passed to locfit
trim	In one-dimensional fitting, the very edges often have high variance. This parameter fixes the estimate on the intervals (0, trim) and (1 - trim, 1).
nn	nearest neighbor parameter
...	additional arguments to be passed to lp in locfit, used only if cv=FALSE

Details

Provide density estimates that are needed by sffdr

Value

A data frame containing:

x1	The first column in eval.points.
x2	The second column in eval.points. Note this is not returned for one dimensional density estimation.
s1	The transformed version of x1.
s2	The transformed version of x2. Note this is not returned for one dimensional density estimation.
fx	Density estimates on the original scale.
fs	Density estimates on the transformed scale.

---

pi0_model *Formulates the model for the proportion of null tests*

Description

pi0_model helps generate the model for the proportion of truly null tests. For more details, refer to the vignette.

Usage

```
pi0_model(z, indep_snps = NULL, basis.df = 3, knots = NULL)
```

Arguments

z	matrix: informative variables that impact the power of the p-values (rows are tests and columns are different informative variables). Currently, there must be no missing values.
indep_snps	vector Boolean indicating the set of independent SNPs
basis.df	integer: the degrees of freedom for the natural cubic spline on each variable. Default is 3 at equally space intervals.
knots	vector: Specify the location of the knots in natural cubic spline. Note that the knots are specified using quantiles by default. Default is NULL and uses basis.df at equally space intervals.

Details

We note that this function is specifically designed for informative p-values and other complex models should be created outside this function.

Value

A list with the following entries:

1. fmod: model formula
2. zt: matrix of rank-transformed informative variables

Author(s)

Andrew Bass

See Also

sffdr

Examples

```
data(bmi)

p <- sumstats$bmi
z <- as.matrix(sumstats[, -1])

# For p-values, you want to specify the lower quantiles
fmod <- pi0_model(z, knots = c(0.005, 0.01, 0.025, 0.05, 0.1))
```

plot.sffdr *Plotting function for sffdr object*

Description

Graphical display of the sffdr object

Usage

```
## S3 method for class 'sffdr'
plot(x, rng = c(0, 5e-08), ...)
```

Arguments

x	A sffdr object.
rng	Significance region to show. Optional.
...	Additional arguments. Currently unused.

Value

Plotting function to summarize significance results from sffdr.

Author(s)

Andrew J. Bass

See Also

sffdr

Examples

```
# import data
data(bmi)

# separate main p-values and conditioning p-values
p <- sumstats$bmi
z <- as.matrix(sumstats[, -1])

# apply pi0_model to create model
knots <- c(0.005, 0.01, 0.025, 0.05, 0.1)
fmod <- pi0_model(z, knots = knots)

# estimate functional pi0
fpi0_out <- fpi0est(p, z = fmod$zt, pi0_model = fmod$fmod)
fpi0 <- fpi0_out$fpi0

# apply sffdr
# Note all tests are independent see 'indep_snps' argument
```

```
# The very small p-values, set epsilon to min of p
sffdr_out <- sffdr(p, fpi0, epsilon = min(p))

# Plot significance results
plot(sffdr_out, rng = c(0, 5e-4))
```

`sffdr` *Estimate the functional p-values, q-values, and local false discovery rates given a set of p-values and informative variables*

Description

Estimate the functional p-values, q-values, and local false discovery rates given a set of p-values and informative variables. The functional p-values is mapping from the functional q-value (FDR-based measure) to a p-value for type I error rate control.

Usage

```
sffdr(
  p.value,
  fpi0,
  surrogate = NULL,
  indep_snps = NULL,
  monotone.window = NULL,
  epsilon = 1e-15,
  nn = NULL,
  fp_ties = TRUE,
  ...
)
```

Arguments

`p.value`	A vector of p-values.
`fpi0`	An estimate of the function proportion of null tests using the `fpi0est` function.
`surrogate`	A surrogate variable that compresses more than one informative variables. Default is NULL. If `fpi0` is specified and `surrogate` is NULL then `fpi0` is used as the surrogate variable.
`indep_snps`	A boolean vector (same size as p) specifying the set of independent tests. Default is NULL and all tests are treated independently.
`monotone.window`	Enforce monotonicity at specified step size. Default is NULL.
`epsilon`	A numerical value the truncation for the p-values during density estimation. Default is 1e-15. You may want to consider decreasing this value if there are a substantial number of small p-values.

nn	A numerical value specifying the nearest neighbor parameter in kernelEstimator. Default is NULL.
fp_ties	A boolean specifying whether ties should be broken using the ordering of the p-values when calculating the fp-values. Only impacts the tests when the local FDR is tied. Default is TRUE.
...	Additional arguments passed to kernelEstimator.

Details

The function fpi0est should be called externally to estimate the functional proportion of null tests given the set of informative variables. The surrogate functional FDR methodology builds from the functional FDR methodology and implements some of the functions from the package.

Value

A list of object type "sffdr" containing:

pvalues	A vector of the original p-values.
fpvalues	A vector of the estimated functional p-values.
fqvalues	A vector of the estimated functional q-values.
flfdr	A vector of the estimated functional local FDR values.
pi0	An vector of the original functional proportion of null tests.
density	An object containing the kernel density estimates from kernelEstimator.

Author(s)

Andrew J. Bass

See Also

fpi0est, plot.sffdr

Examples

```
# import data
data(bmi)

# separate main p-values and conditioning p-values
p <- sumstats$bmi
z <- as.matrix(sumstats[, -1])

# apply pi0_model to create model
knots <- c(0.005, 0.01, 0.025, 0.05, 0.1)
fmod <- pi0_model(z, knots = knots)

# estimate functional pi0
fpi0_out <- fpi0est(p, z = fmod$zt, pi0_model = fmod$fmod)
fpi0 <- fpi0_out$fpi0
```

```
# apply sffdr
# Note all tests are independent see 'indep_snps' argument
# The data has very small p-values, set epsilon to min of p
sffdr_out <- sffdr(p, fpi0, epsilon = min(p))

# Plot significance results
plot(sffdr_out, rng = c(0, 5e-4))

# Functional P-values, Q-values, and local FDR
fp <- sffdr_out$fpvalues
fq <- sffdr_out$fqvalues
flfdr <- sffdr_out$flfdr
```

Index

∗ **fpi0est**
    fpi0est, 4
∗ **pi0_model**
    pi0_model, 7
∗ **plot**
    plot.sffdr, 9
∗ **sffdr**
    sffdr, 10

bmi, 2

ffinemap, 3
fpi0est, *4*, 4, *10, 11*
fpvalues, 6

gam, *4*
gam.control, *4*
glm, *4*
glm.control, *4*

kernelEstimator, 6, *11*

pi0_model, 7
plot, *(*plot.sffdr*)*, 9
plot.sffdr, *5*, 9, *11*

sffdr, *2*, *5*, *8*, *9*, 10
sumstats *(*bmi*)*, 2