Package ‘DRpower’

July 13, 2023

Type Package

Title Study design and analysis for pfhrp2/3 deletion prevalence studies

Version 0.1.0

Description This package can be used in the design and/or analysis stages of Plasmodium falciparum pfhrp2/3 deletion prevalence studies. We assume that the study takes the form of a clustered prevalence survey, meaning the data consists of a numerator (number tested) and denominator (number of deletions found) over multiple clusters. We are interested in estimating the study-level prevalence, i.e. over all clusters, while accounting for the possibility of high intra-cluster correlation. The analysis approach uses a Bayesian random effects model to estimate prevalence and intra-cluster correlation. The approach to power analysis is simulation-based, running the analysis many times on simulated data and estimating empirical power. This method can be used to establish a minimum sample size required to achieve a given target power.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

LazyDataCompression xz

RoxygenNote 7.2.3

BugReports https://github.com/mrc-ide/DRpower/issues

Imports dplyr, extraDistr, ggplot2, kableExtra, magrittr, pandoc, Rcpp, rlang, statmod, tidyr

Suggests knitr, rmarkdown, testthat (>= 3.0.0), ggridges, tidyverse

Config/testthat/edition 3

LinkingTo Rcpp

VignetteBuilder knitr

NeedsCompilation yes

Author Bob Verity [aut, cre]

Maintainer Bob Verity <r.verity@imperial.ac.uk>

Depends R (>= 3.5.0)

R topics documented:

`check_DRpower_loaded` *Check that DRpower package has loaded successfully*

Description

Simple function to check that DRpower package has loaded successfully. Prints "DRpower loaded successfully!" if so.

Usage

```
check_DRpower_loaded()
```

`df_sim` *Summary of simulations from the threshold analysis*

Description

TODO

Usage

```
data(df_sim)
```

Format

A data frame of 547200 rows and 32 columns. The first 13 columns give parameter combinations that were used in simulating and analysing data. The "reps" column gives the number of times simulation was repeated, and "seed" gives the value of the seed that was used at the start of this loop (to ensure reproducibility). "prev_thresh" gives the prevalence threshold used in hypothesis testing. The remaining 16 columns give summary results over simulations. The MAP, posterior mean, posterior median, lower and upper credible intervals, and the probability of being above the target threshold are all summarised in terms of their mean and variance. "n_reject" gives the number of times the hypothesis of being below the threshold was rejected. This is used to estimate empirical power as `n_reject / reps`, and lower and upper CIs are calculated around this using the method of Clopper and Pearson.

Examples

```
data(df_sim)
```

`df_ss` *Minimum sample sizes for the threshold analysis*

Description

TODO

Usage

```
data(df_ss)
```

Format

A data frame of 6840 rows and 15 columns. The first 14 columns give parameter combinations that were used in simulating and analysing data. The final "N_opt" column gives the optimal sample size to achieve a power of 80

Examples

```
data(df_ss)
```

`DRpower` *The DRpower app for design and analysis of Plasmodium falciparum pfhrp2/3 data*

Description

This package can be used in the design and/or analysis stages of Plasmodium falciparum pfhrp2/3 deletion prevalence studies. We assume that the study takes the form of a clustered prevalence survey, meaning the data consists of a numerator (number tested) and denominator (number of deletions found) over multiple clusters. We are interested in estimating the study-level prevalence, i.e. over all clusters, while accounting for the possibility of high intra-cluster correlation. The analysis approach uses a Bayesian random effects model to estimate prevalence and intra-cluster correlation. The approach to power analysis is simulation-based, running the analysis many times on simulated data and estimating empirical power. This method can be used to establish a minimum sample size required to achieve a given target power.

get_joint_grid — *Get posterior distribution of both prevalence and the ICC on a grid*

Description

Get posterior distribution of both prevalence and the ICC on a grid. This can be useful for producing e.g. a contour plot of the posterior distribution of both parameters.

Usage

```
get_joint_grid(
  n,
  N,
  prior_prev_shape1 = 1,
  prior_prev_shape2 = 1,
  prior_ICC_shape1 = 1,
  prior_ICC_shape2 = 9,
  prev_cells = 64,
  ICC_cells = 64
)
```

Arguments

`n, N` the numerator (`n`) and denominator (`N`) per cluster (vectors).

`prior_prev_shape1, prior_prev_shape2, prior_ICC_shape1, prior_ICC_shape2`
parameters that dictate the shape of the Beta priors on prevalence and the ICC. Increasing the first shape parameter (e.g. `prior_prev_shape1`) pushes the distribution towards 1, increasing the second shape parameter (e.g. `prior_prev_shape2`) pushes the distribution towards 0. Increasing both shape parameters squeezes the distribution towards the centre and therefore makes it narrower. The default values of these parameters are based on an analysis of historical pfhrp2/3 studies.

`prev_cells, ICC_cells`
the number of cells in the grid in each dimension.

Examples

```
get_joint_grid(n = c(5, 2, 9), N = c(100, 80, 120))
```

get_posterior — *Estimate prevalence and intra-cluster correlation from raw counts*

Description

Takes raw counts of the number of positive samples per cluster (numerator) and the number of tested samples per cluster (denominator) and returns posterior estimates of the prevalence and intra-cluster correlation coefficient (ICC).

Usage

```
get_prevalence(
  n,
  N,
  alpha = 0.05,
  prev_thresh = 0.05,
  prior_prev_shape1 = 1,
  prior_prev_shape2 = 1,
  prior_ICC_shape1 = 1,
  prior_ICC_shape2 = 9,
  MAP_on = TRUE,
  post_mean_on = FALSE,
  post_median_on = FALSE,
  post_CrI_on = TRUE,
  post_thresh_on = TRUE,
  post_full_on = FALSE,
  post_full_breaks = seq(0, 1, l = 1001),
  CrI_type = "HDI",
  n_intervals = 20,
  round_digits = 2,
  debug_on = FALSE,
  use_cpp = TRUE
)

get_ICC(
  n,
  N,
  alpha = 0.05,
  prior_prev_shape1 = 1,
  prior_prev_shape2 = 1,
  prior_ICC_shape1 = 1,
  prior_ICC_shape2 = 9,
  MAP_on = TRUE,
  post_mean_on = FALSE,
  post_median_on = FALSE,
  post_CrI_on = TRUE,
  post_full_on = FALSE,
  post_full_breaks = seq(0, 1, l = 1001),
  CrI_type = "HDI",
  n_intervals = 20,
  round_digits = 4,
  debug_on = FALSE,
  use_cpp = TRUE
)
```

Arguments

`n, N`	the numerator (`n`) and denominator (`N`) per cluster (vectors).
`alpha`	the significance level of the credible interval - for example, use `alpha = 0.05` for a 95% interval. See also `CrI_type` argument for how this is calculated.
`prev_thresh`	return the probability that the prevalence is above this threshold. Can be a vector, in which case the return object contains one value for each input.

`prior_prev_shape1, prior_prev_shape2, prior_ICC_shape1, prior_ICC_shape2`
: parameters that dictate the shape of the Beta priors on prevalence and the ICC. Increasing the first shape parameter (e.g. `prior_prev_shape1`) pushes the distribution towards 1, increasing the second shape parameter (e.g. `prior_prev_shape2`) pushes the distribution towards 0. Increasing both shape parameters squeezes the distribution towards the centre and therefore makes it narrower. The default values of these parameters are based on an analysis of historical pfhrp2/3 studies.

`MAP_on, post_mean_on, post_median_on, post_CrI_on, post_thresh_on, post_full_on`
: a series of boolean (TRUE/FALSE) objects specifying which outputs to produce. The following options are available:

  - `MAP_on`: if `TRUE` then return the maximum a posteriori.
  - `post_mean_on`: if `TRUE` then return the posterior mean.
  - `post_median_on`: if `TRUE` then return the posterior median.
  - `post_CrI_on`: if `TRUE` then return the posterior credible interval at significance level `alpha`. See `CrI_type` argument for how this is calculated.
  - `post_thresh_on`: if `TRUE` then return the posterior probability of being above the threshold specified by `prev_thresh`.
  - `post_full_on`: if `TRUE` then return the full posterior distribution (as approximated using the adaptive quadrature approach) in 0.1% intervals from 0% to 100%.

`post_full_breaks`
: a vector of breaks at which to evaluate the full posterior distribution (if `post_full_on = TRUE`).

`CrI_type`
: which method to use when computing credible intervals, with options "ETI" (equal-tailed interval) and "HDI" (high-density interval). The ETI searches a distance `alpha/2` from either side of the [0,1] interval. The HDI method returns the narrowest interval that subtends a proportion `1-alpha` of the distribution. The HDI method is used by default.

`n_intervals`
: the number of intervals used in the adaptive quadrature method. Increasing this value gives a more accurate representation of the true posterior, but comes at the cost of reduced speed.

`round_digits`
: the number of digits after the decimal point that are used when reporting estimates. This is to simplify results and to avoid giving the false impression of extreme precision.

`debug_on`
: for use in debugging. If `TRUE` and if `use_cpp = FALSE` then produces a plot of the posterior distribution evaluated by brute force (black) overlaid with the adaptive quadrature approximation (blue) for direct comparison. If `use_cpp = TRUE` then only the approximation is plotted. If the approximate distribution does not agree closely with the brute force solution then consider increasing the value of `n_intervals`. Note that producing this plot can be very slow, and so this option should be turned off when not needed.

`use_cpp`
: if `TRUE` then use an Rcpp implementation of the adaptive quadrature approach that is much faster and therefore useful when running the function a large number of times.

Details

There are two unknown quantities in the DRpower model: the prevalence and the ICC. Thes following functions integrate over a prior on one quantity to give the marginal posterior distribution of the other. Possible outputs include the posterior mean, median, credible interval (CrI), probability of being above a threshold, and the full posterior distribution. For speed, distributions are approximated using an adaptive quadrature approach, in which the full distribution is split into intervals of differing width and each intervals is approximated using Simpson's rule. The number of intervals used in quadrature can be increased for more accurate results, at the cost of slower speed.

Examples

```
# basic example of estimating prevalence and ICC from observed counts
df_counts <- data.frame(sample_size = c(80, 110, 120),
                        deletions = c(3, 5, 6))
get_prevalence(n = df_counts$deletions, N = df_counts$sample_size)
get_ICC(n = df_counts$deletions, N = df_counts$sample_size)
```

get_power_presence *Calculate power when testing for presence of deletions*

Description

Calculates power directly for the case of a clustered prevalence survey where the aim is to detect the presence of *any* deletions over all clusters. This design can be useful as a pilot study to identify priority regions where deletions are likely. Note that we need to take account of intra-cluster correlation here, as a high ICC will make it more likely that we see zero deletions even when the prevalence is non-zero.

Usage

```
get_power_presence(N, prevalence = 0.01, ICC = 0.1)
```

Arguments

N	vector giving the number of samples obtained from each cluster.
prevalence	assumed true prevalence of pfhrp2 deletions. Input as proportion between 0 and 1.
ICC	assumed true intra-cluster correlation (ICC), between 0 and 1.

Examples

```
get_power_presence(N = c(120, 90, 150), prevalence = 0.01, ICC = 0.1)
```

get_power_threshold *Estimate power when testing prevalence against a threshold*

Description

Estimates power empirically via repeated simulation for the case of a clustered prevalence survey comparing against a set threshold. Returns an estimate of the power, along with lower and upper bounds of this estimate.

Usage

```
get_power_threshold(
  N,
  prevalence = 0.1,
  ICC = 0.05,
  prev_thresh = 0.05,
  rejection_threshold = 0.95,
  prior_prev_shape1 = 1,
  prior_prev_shape2 = 1,
  prior_ICC_shape1 = 1,
  prior_ICC_shape2 = 9,
  n_intervals = 20,
  round_digits = 2,
  reps = 100
)
```

Arguments

`N` vector giving the number of samples obtained from each cluster.

`prevalence` assumed true prevalence of pfhrp2 deletions. Input as proportion between 0 and 1.

`ICC` assumed true intra-cluster correlation (ICC), between 0 and 1.

`prev_thresh` the threshold prevalence that we are testing against.

`rejection_threshold`
the posterior probability of being above the prevalence threshold needs to be greater than `rejection_threshold` in order to reject the null hypothesis.

`prior_prev_shape1, prior_prev_shape2, prior_ICC_shape1, prior_ICC_shape2`
parameters that dictate the shape of the Beta priors on prevalence and the ICC. Increasing the first shape parameter (e.g. `prior_prev_shape1`) pushes the distribution towards 1, increasing the second shape parameter (e.g. `prior_prev_shape2`) pushes the distribution towards 0. Increasing both shape parameters squeezes the distribution towards the centre and therefore makes it narrower. The default values of these parameters are based on an analysis of historical pfhrp2/3 studies.

`n_intervals` the number of intervals used in the adaptive quadrature method. Increasing this value gives a more accurate representation of the true posterior, but comes at the cost of reduced speed.

`round_digits` the number of digits after the decimal point that are used when reporting estimates. This is to simplify results and to avoid giving the false impression of extreme precision.

`reps` number of times to repeat simulation per parameter combination.

Details

Estimates power using the following approach:

1. Simulate repeatedly from the function `rbbinom_reparam()` using known values (e.g. a known "true" prevalence and intra-cluster correlation).
2. Analyse data using `get_prevalence()` to determine the probability of being above `prev_thresh`.
3. If this probability is above `rejection_threshold` then reject the null hypothesis, and encode this as a single correct conclusion.
4. Count the number of simulations for which the correct conclusion is reached, relative to the total number of simulations. This gives an estimate of empirical power, along with upper and lower 95% binomial CIs on the power via the method of Clopper and Pearson (1934).

Note that this function can be run even when `prevalence` is less than `prev_thresh`, although in this case what is returned is not the power. Power is defined as the probability of *correctly* rejecting the null hypothesis, whereas here we would be incorrectly rejecting the null. Therefore, what we obtain in this case is an estimate of the false positive rate.

References

Clopper, C.J. and Pearson, E.S., 1934. The use of confidence or fiducial limits illustrated in the case of the binomial. Biometrika, 26, 404–413. doi: 10.2307/2331986.

Examples

```
get_power_threshold(N = c(120, 90, 150), prevalence = 0.15, ICC = 0.1 , reps = 1e2)
```

get_sample_size_presence

Get minimum sample size when testing for presence of deletions

Description

Calculates the minimum sample size required per cluster to achieve a certain power for the case of a clustered prevalence survey where the aim is to detect the presence of *any* deletions over all clusters (see ?`get_power_presence()`). Assumes the same sample size per cluster.

Usage

```
get_sample_size_presence(
  n_clust,
  target_power = 0.8,
  prevalence = 0.01,
  ICC = 0.1,
  N_max = 2000
)
```

Arguments

`n_clust`	the number of clusters.
`target_power`	the power we are aiming to achieve.
`prevalence`	assumed true prevalence of pfhrp2 deletions. Input as proportion between 0 and 1.
`ICC`	assumed true intra-cluster correlation (ICC), between 0 and 1.
`N_max`	the maximum allowed sample size.

Examples

```
get_sample_size_presence(n_clust = 5, prevalence = 0.01, ICC = 0.1)
```

`historical_data` *TODO*

Description

TODO

Usage

```
data(historical_data)
```

Format

TODO

Examples

```
data(historical_data)
```

`studies_inclusion` *TODO*

Description

TODO

Usage

```
data(studies_inclusion)
```

Format

TODO

Examples

```
data(studies_inclusion)
```