Package 'BBMBC'

November 22, 2026

Title Beta-Binomial Models for Evaluating Overdispersed B-Cell Phenotyping Data

Version 1.0.3

Description Provides tools for statistical power calculations to plan early-phase vaccine and immunotherapy trials comparing rare, antigen-specific immune cell frequencies. Originally designed for memory B-cell phenotyping studies and generalizable to high-parameter flow cytometry assays across vaccinology and immuno-oncology, the models overdispersed count distributions using Beta-Binomial generalized linear models. Unlike rank-based tests, this approach retains response magnitude information while accounting for extra-binomial variation, small sample sizes, unequal between-participant variability, and uncertainty in pilot estimates. Includes method-of-moments dispersion estimation, likelihood ratio testing across experimental arms, and Monte Carlo simulation to evaluate statistical power with well-controlled empirical Type I error rates. This workflow is directly motivated by the analysis of immunology data in ``Vaccination with mRNA-encoded membrane-anchored HIV envelope trimers elicited tier 2 neutralizing antibodies in a phase 1 clinical trial" (Parks et al. 2025) <doi:10.1126/scitranslmed.ady6831>.

Encoding UTF-8

Depends R (>= 4.1.0)

Imports doParallel, foreach, glmmTMB, parallel, stats, utils

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Config/roxygen2/version 8.1.1

License MIT + file LICENSE

NeedsCompilation no

Author Andrew Shin [aut, cre]

Maintainer Andrew Shin <ashin2@uw.edu>

Repository CRAN

Date/Publication 2026-11-22 00:10:02 UTC

Contents

calc_power_bb 2
calc_rho_mom 4
calc_t1_bb 5
test_diff_bb 6

Index **8**

`calc_power_bb` *Calculate Statistical Power for Beta-Binomial Models (Parallelized)*

Description

Simulates overdispersed cell frequency data under the alternative hypothesis to estimate the statistical power of a Beta-Binomial generalized linear model.

Usage

```
calc_power_bb(
  n_0 = 20,
  n_1 = 20,
  p_base = 0.005,
  fold_change = 2,
  rho_0 = 0.006,
  rho_1 = 0.006,
  alpha = 0.05,
  min_cells = 1e+05,
  max_cells = 1e+06,
  seed = 2025,
  n_sims = 1000,
  n_cores = min(2, max(1, parallel::detectCores() - 1))
)
```

Arguments

`n_0`	Integer. Biological replicates in control arm (default: 20).
`n_1`	Integer. Biological replicates in treatment arm (default: 20).
`p_base`	Numeric. Baseline proportion in control group (default: 0.005).
`fold_change`	Numeric. Ratio of treatment mean to control mean (default: 2.0).
`rho_0`	Numeric. Overdispersion for control group (default: 0.006).
`rho_1`	Numeric. Overdispersion for treatment group (default: 0.006).
`alpha`	Numeric. Significance threshold for likelihood ratio test (default: 0.05).
`min_cells`	Integer. Minimum total cells per sample (default: 1e5).
`max_cells`	Integer. Maximum total cells per sample (default: 1e6).

seed	Integer. Random seed for reproducibility (default: 2025).
n_sims	Integer. Number of simulations to run (default: 1000).
n_cores	Integer. Number of cores for parallel processing. Defaults to 2 (CRAN safe default).

Value

A list of class power_result containing the estimated statistical power, convergence metrics, and the input parameters used.

Examples

```
# 1. Sample size planning for an early-phase vaccine trial
# Estimate power to detect a 2-fold increase in rare antigen-specific
# memory B cells (0.5% baseline vs. 1.0% treatment) across candidate arm
# sizes. Increase n_sims (e.g., 1000+) for formal study planning.
sample_sizes <- c(10, 15, 20, 25)

power_table <- data.frame(
  n_per_arm = sample_sizes,
  total_n = sample_sizes * 2,
  power = NA_real_
)

for (i in seq_along(sample_sizes)) {
  n <- sample_sizes[i]
  pwr_res <- calc_power_bb(
    n_0 = n,
    n_1 = n,
    p_base = 0.005,
    fold_change = 2.0,
    rho_0 = 0.006,
    rho_1 = 0.006,
    n_sims = 10,
    n_cores = 1
  )
  power_table$power[i] <- pwr_res$power
}

# Inspect the power curve and find arm sizes meeting an 80% power target
print(power_table)
power_table[power_table$power >= 0.80, ]

# 2. Unbalanced randomization (1:2) with unequal between-arm dispersion
pwr_unbalanced <- calc_power_bb(
  n_0 = 10,
  n_1 = 20,
  p_base = 0.005,
  fold_change = 2.0,
  rho_0 = 0.004,
  rho_1 = 0.012,
  n_sims = 10,
```

```
  n_cores = 1
)
pwr_unbalanced$power
```

`calc_rho_mom` *Estimate Overdispersion (Rho) via Method of Moments*

Description

Calculates a rapid empirical estimate of the intra-class correlation (overdispersion parameter, rho) for proportion data. This method of moments estimator uses the ratio of the sample variance of the proportions to the theoretical maximum variance of a true binomial distribution with the same mean.

Usage

```
calc_rho_mom(props, na.rm = TRUE)
```

Arguments

`props`	Numeric vector of observed proportions (must be between 0 and 1).
`na.rm`	Logical. Should missing values (including NaN) be removed before computation? (default: TRUE).

Value

A numeric value representing the estimated overdispersion parameter (rho). Returns `NA` if the sample mean is exactly 0 or 1, as the denominator becomes 0.

Examples

```
# Simulate 100 proportions with a known mean of 0.2 and overdispersion of 0.05
set.seed(123)
p_base <- 0.2
rho_true <- 0.05
shape1 <- p_base * (1 - rho_true) / rho_true
shape2 <- (1 - p_base) * (1 - rho_true) / rho_true

sim_props <- stats::rbeta(100, shape1, shape2)

# Estimate rho empirically
estimated_rho <- calc_rho_mom(sim_props)
print(estimated_rho)
```

`calc_t1_bb` *Calculate Type I Error for Beta-Binomial Models*

Description

Simulates immunology data under the null hypothesis of equal means (no biological difference in cell frequency) using parallel processing. Handles balanced/unbalanced sample sizes and dynamic overdispersion.

Usage

```
calc_t1_bb(
  n_0 = 20,
  n_1 = 20,
  p_base = 0.005,
  rho_0 = 0.006,
  rho_1 = 0.006,
  alpha = 0.05,
  min_cells = 1e+05,
  max_cells = 1e+06,
  seed = 2026,
  n_sims = 1000,
  n_cores = min(2, max(1, parallel::detectCores() - 1))
)
```

Arguments

`n_0`	Integer. The number of biological replicates in the control arm (default: 20).
`n_1`	Integer. The number of biological replicates in the treatment arm (default: 20).
`p_base`	Numeric. Baseline proportion of the target cell population for BOTH groups (default: 0.005).
`rho_0`	Numeric. Intra-class correlation (overdispersion) for the control group (default: 0.006).
`rho_1`	Numeric. Intra-class correlation (overdispersion) for the treatment group (default: 0.006).
`alpha`	Numeric. Significance threshold for the likelihood ratio test (default: 0.05).
`min_cells`	Integer. Minimum number of total cells analyzed per sample (default: 1e5).
`max_cells`	Integer. Maximum number of total cells analyzed per sample (default: 1e6).
`seed`	Integer. Random seed for reproducibility (default: 2026).
`n_sims`	Integer. The number of simulations to run (default: 1000).
`n_cores`	Integer. Number of cores to use for parallel processing. Defaults to 2 (CRAN safe default).

Value

A list of class `t1_error_result` containing the estimated Type I error rate, convergence metrics, and the input parameters.

Examples

```
# 1. Early-phase vaccine trial with balanced arms and equal overdispersion
# Check empirical Type I error for comparing rare antigen-specific memory
# B cells (0.5% baseline frequency, 100k-1M total B cells per sample) across
# 15 participants per arm. Increase n_sims (e.g., 1000+) for study planning.

res_equal <- calc_t1_bb(
  n_0 = 15,
  n_1 = 15,
  p_base = 0.005,
  rho_0 = 0.006,
  rho_1 = 0.006,
  min_cells = 1e5,
  max_cells = 1e6,
  alpha = 0.05,
  n_sims = 10,
  n_cores = 1
)

# Inspect the empirical Type I error rate and model convergence
res_equal$type_1_error_rate
res_equal$convergences
res_equal$failed_fits

# 2. Unbalanced design (1:2 randomization) with unequal arm variability
# Specifying rho_0 != rho_1 automatically fits a group-specific dispersion
# model (dispformula = ~ group).
res_unequal <- calc_t1_bb(
  n_0 = 10,
  n_1 = 20,
  p_base = 0.005,
  rho_0 = 0.004,
  rho_1 = 0.012,
  alpha = 0.05,
  n_sims = 10,
  n_cores = 1
)

res_unequal$type_1_error_rate
res_unequal$parameters$model_dispformula
```

`test_diff_bb` *Beta-Binomial Likelihood Ratio Test for Group Differences*

Description

Performs a Likelihood Ratio Test (LRT) using Beta-Binomial models to evaluate whether the underlying proportion of events differs significantly between two groups.

Usage

```
test_diff_bb(
  data,
  success_col,
  total_col,
  group_col,
  unequal_dispersion = FALSE
)
```

Arguments

data	A data frame containing the experimental data.
success_col	Character string specifying the column name for successes.
total_col	Character string specifying the column name for total trials.
group_col	Character string specifying the column name with group labels (must have 2 levels).
unequal_dispersion	Logical. If TRUE, models group-specific overdispersion. Default: FALSE.

Value

A list of class `bb_test_result` containing p-value, chi-square statistic, degrees of freedom, log-odds ratio, and its standard error.

Examples

```
set.seed(2025)
trial_data <- data.frame(
  arm = rep(c("Control", "Vaccine"), each = 15),
  total_bcells = as.integer(stats::runif(30, 100000, 500000))
)
trial_data$target_cells <- ifelse(
  trial_data$arm == "Control",
  stats::rbinom(15, trial_data$total_bcells, stats::rbeta(15, 5, 995)),
  stats::rbinom(15, trial_data$total_bcells, stats::rbeta(15, 10, 990))
)

result <- test_diff_bb(
  data = trial_data,
  success_col = "target_cells",
  total_col = "total_bcells",
  group_col = "arm",
  unequal_dispersion = FALSE
)
print(result$p_value)
```

Index

calc_power_bb, 2
calc_rho_mom, 4
calc_t1_bb, 5

test_diff_bb, 6