Package 'barmixR'

November 10, 2026

Type Package

Title Bayesian Modeling of Barcoded Tumor Mixtures for Quantitative Treatment Resistance Analysis

Version 0.99.3

Description Implements the Bayesian modeling framework underlying the barmixR (BARcode MIXture analysis) platform for high-throughput quantitative analysis of genotype-specific treatment responses in pooled cancer cell populations. The package integrates barcode sequencing count data with volumetric measurements such as tumor volume (in vivo) or cellular confluency (in vitro) using hierarchical probabilistic models. Barcode counts are modeled with a Dirichlet–multinomial distribution to account for compositional sequencing data, while volumetric measurements are modeled using log-normal (tumor volume) or beta (confluency) likelihoods. Posterior inference is performed using Hamiltonian Monte Carlo through 'rstan'. The resulting posterior distributions enable estimation of clone-specific quantitative treatment resistance (QTR) together with uncertainty propagation from both sequencing and volumetric data. Additional functions provide posterior predictive checks, estimation of resistance ratios, treatment ranking, and visualization of resistance landscapes using violin plots and bubble heatmaps. The methods are designed for multiplexed lineage-tracing experiments in cancer research and were developed to analyze treatment resistance in gastrointestinal stromal tumors (GIST), but are broadly applicable to barcoding-based studies of treatment response and clonal dynamics across diverse cancer types.

License GPL-3 + file LICENSE

URL https://github.com/MohammadDarbalaei/barmixR

BugReports https://github.com/MohammadDarbalaei/barmixR/issues

Encoding UTF-8

LazyData false

Depends R (>= 4.3.0)

Imports utils, BiocParallel, dplyr, forcats, ggplot2, methods, patchwork, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), rstantools (>= 2.4.0), stats

Suggests BiocStyle, knitr, MGLM, rmarkdown, testthat, tidyverse

Config/testthat/edition 3

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

biocViews Software, Bayesian, Sequencing, Visualization

Config/roxygen2/version 8.0.0

Biarch true

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

SystemRequirements GNU make, C++17, StanHeaders

git_url https://git.bioconductor.org/packages/barmixR

git_branch devel

git_last_commit 1d3b628

git_last_commit_date 2026-09-12

Repository Bioconductor 3.24

Date/Publication 2026-11-09

Author Mohammad Darbalaei [aut, cre] (ORCID: <https://orcid.org/0009-0003-4561-0048>), Daniel Hoffmann [aut] (ORCID: <https://orcid.org/0000-0003-2973-7869>), Barbara M. Grüner [ctb], Thomas Mühlenberg [ctb], Julia Zummack [ctb]

Maintainer Mohammad Darbalaei <mohammad.darbalaei@uni-due.de>

Contents

barmixRQTR 2
barmixR_fit 6
fractionRatio 7
populationRatio 9
ppcBarcodes 11
ppcPopulation 13
QTRDecision 15
QTRheatmap 18
QTRresistance 21

Index **25**

`barmixRQTR` *Fit the barmixR model for quantitative treatment resistance (QTR)*

Description

Fits the Bayesian barmixR framework for joint modeling of barcode composition and population size measurements in pooled cell populations. The model integrates barcode sequencing counts with population size data to estimate quantitative treatment resistance (QTR) for individual cell lines.

Usage

```
barmixRQTR(
  data,
  control_group = NULL,
 dispersion = list(psi_mean = 0, psi_sd = 0.5, varphi_mean = 0, varphi_sd = 0.01),
 control = list(chains = 3, iter_count = 1000, iter_V = 10000, cores = 3, warmup = NULL),
  time_d,
  q_up = 0.975,
  q_lo = 0.025,
  n_sam = NULL,
  in_vivo = TRUE,
  seed = NULL,
  verbose = FALSE,
  BPPARAM = NULL,
  ...
)
```

Arguments

data	A list containing the following components: • data_count: A matrix or data frame of barcode read counts. Rows correspond to experimental replicates (e.g., DMSO_1) and columns correspond to genotypes or mutations (e.g., PTEN, TSC2). \item \code{condition_count}: A factor or character vector specifying treatment groups corresponding to rows of \code{data_count}. \item \code{V}: A numeric vector of population size measurements, representing tumor volume in \emph{in vivo} experiments or confluency in \emph{in vitro} assays. \item \code{condition_v}: A factor or character vector specifying treatment groups corresponding to entries of \code{V}. \item \code{cell_line}: A factor or character vector of cell line identifiers corresponding to the columns of \code{data_count}. \item \code{VT0}: Optional numeric vector of baseline population size values used for normalization (tumor volume in \emph{in vivo} experiments or confluency in \emph{in vitro} assays). \item \code{pre_composition}: Optional matrix or data frame used to adjust barcode compositions prior to model fitting.
control_group	Optional label specifying the control treatment group used for comparative analysis.
dispersion	A list controlling dispersion priors in the Dirichlet-multinomial model: • psi_mean: Mean for the log-normal prior on the precision parameter psi (default: 0). \item \code{psi_sd}: Standard deviation for the log-normal prior on \code{psi} (default: 0.5).

\item \code{varphi_mean}: Mean for the log-normal prior on the variance parameter \code{varphi} (default: 0).

\item \code{varphi_sd}: Standard deviation for the log-normal prior on \code{varphi} (default: 0.01).

`control` A list of control parameters:

- `chains`: Number of MCMC chains (default: 3).

\item \code{iter_count}: Iterations used to fit the barcode composition model (default: 1000).

\item \code{iter_V}: Iterations used to fit the population size model (default: 10000).

\item \code{cores}: Number of CPU cores used for Stan sampling (default: 3).

\item \code{warmup}: Optional number of warm-up iterations passed to Stan. If \code{NULL}, Stan default behavior is used.

`time_d`	Duration of the experiment used for estimating treatment resistance.
`q_up`	Upper quantile threshold used in downstream filtering (default: 0.975).
`q_lo`	Lower quantile threshold used in downstream filtering (default: 0.025).
`n_sam`	Number of posterior samples retained for downstream analysis (defaults to half of `min(iter_count, iter_V)`).
`in_vivo`	Logical indicating the experimental setting. If `TRUE`, population size corresponds to tumor volume in *in vivo* experiments and is modeled using a log-normal likelihood. If `FALSE`, population size corresponds to confluency in *in vitro* assays and is modeled using a beta likelihood.
`seed`	Optional integer seed used for reproducible Stan sampling.
`verbose`	Logical; if `TRUE`, informational messages are printed during model fitting.
`BPPARAM`	Optional `BiocParallelParam` object used to determine parallel workers for Stan sampling.
`...`	Reserved for future extensions.

Details

Population size measurements correspond to tumor volume in *in vivo* experiments or cellular confluency in *in vitro* assays. Barcode counts are modeled using a Dirichlet-multinomial distribution to account for compositional sequencing data, while population size measurements are modeled using a log-normal likelihood for tumor volume (*in vivo*) or a beta likelihood for confluency (*in vitro*).

Posterior inference is performed using Hamiltonian Monte Carlo implemented in **rstan**.

Value

An object of class `barmixR_fit`. This is a list containing:

- `fit_dir_mult`: Fitted Dirichlet-multinomial model for barcode counts.

\item \code{fit_V}: Fitted model for population size
  (tumor volume in \emph{in vivo} experiments or confluency in
  \emph{in vitro} assays).

\item \code{L}: Number of unique cell lines.

\item \code{iter_count}, \code{iter_V}, \code{chains}: MCMC
  parameters used for model fitting.

\item \code{data_count}, \code{V}, \code{VT0}: Input data used
  after preprocessing and filtering.

\item \code{group}, \code{group_V}: Numeric identifiers for
  treatment groups.

\item \code{condition_count}, \code{condition_v}: Treatment
  labels for barcode counts and population size data.

\item \code{control_group}: Identifier for the control group.

\item \code{n_sam}: Number of posterior samples retained for
  downstream analysis.

\item \code{in_vivo}: Logical indicating whether the model was
  fitted using the \emph{in vivo} likelihood.

Examples

```
set.seed(1)

## ----------------------------
## Simulate small example data
## ----------------------------

# barcode count matrix (4 samples x 5 clones)
data_count <- matrix(
  c(
    20, 22, 24, 23, 21,   # DMSO_rep1
    21, 23, 25, 22, 20,   # DMSO_rep2
    26, 25, 22, 20, 18,   # TreatA_rep1
    25, 24, 23, 21, 19    # TreatA_rep2
  ),
  nrow = 4,
  byrow = TRUE
)

colnames(data_count) <- paste0("clone", 1:5)
rownames(data_count) <- c("DMSO_rep1", "DMSO_rep2",
                          "TreatA_rep1", "TreatA_rep2")

# treatment labels
condition_count <- factor(c("DMSO", "DMSO", "TreatA", "TreatA"))

# population size (e.g. tumor volume)
V <- c(300, 280, 100, 120)
```

```
# assemble input
data <- list(
  data_count = data_count,
  condition_count = condition_count,
  V = V,
  condition_v = condition_count,
  cell_line = factor(colnames(data_count))
)

## Fit model (fast settings for example only)
## Note: iteration numbers are intentionally small to ensure the example runs quickly.
## For real analysis, increase iterations (e.g., 10000) and use multiple chains.

fit <- barmixRQTR(
  data = data,
  time_d = 10,
  control = list(
    chains = 1,
    iter_count = 50,
    iter_V = 50,
    cores = 1
  ),
  seed = 123
)

## inspect result
names(fit)
```

`barmixR_fit` *Class barmixR_fit*

Description

Object returned by `barmixRQTR()`.

Details

Contains fitted Bayesian models and associated metadata for barcode composition and population size.

Value

An object of class `barmixR_fit`.

See Also

`barmixRQTR`

`fractionRatio` *Compute ratios of relative barcode abundance between treatments*

Description

Computes the ratio of relative barcode abundance for each cell line between treatment groups and a control group using posterior samples of barcode fractions obtained from the barmixR model.

Usage

```
fractionRatio(model, sampled_fraction, q_up = 0.975, q_lo = 0.025)
```

Arguments

`model`	An object of class `barmixR_fit` returned by `barmixRQTR` containing: • `names_cell`: Vector of cell line identifiers. • `control_group`: Name of the control group. • `condition_count`: Vector of treatment conditions. • `n_sam`: Number of posterior samples retained. • `L`: Number of cell lines. • `K`: Number of treatment conditions.
`sampled_fraction`	A list of matrices containing posterior samples of barcode fractions for each treatment condition, typically generated by `ppcBarcodes()`.
`q_up`	Upper quantile threshold used to remove extreme values (default: 0.975).
`q_lo`	Lower quantile threshold used to remove extreme values (default: 0.025).

Details

Posterior samples of barcode composition are derived from the Dirichlet-multinomial component of the model. Relative abundance ratios are calculated for each treatment relative to the control group and summarized using log-transformed values.

Value

An object of class `barmixR_result` containing:

- `plot_ratio_fraction`: A `ggplot2` violin plot showing log-ratios of relative barcode abundance.

```
\item \code{li_sam_ratio_relative}: A list of matrices containing
  computed relative abundance ratios.

\item \code{type}: Result type ("fraction_ratio").
```

Examples

```
set.seed(1)

## ---------------------------
## Simulate small example data
## ---------------------------

# barcode count matrix (4 samples x 5 clones)
data_count <- matrix(
  c(
    20, 22, 24, 23, 21,   # DMSO_rep1
    21, 23, 25, 22, 20,   # DMSO_rep2
    26, 25, 22, 20, 18,   # TreatA_rep1
    25, 24, 23, 21, 19    # TreatA_rep2
  ),
  nrow = 4,
  byrow = TRUE
)

colnames(data_count) <- paste0("clone", 1:5)
rownames(data_count) <- c("DMSO_rep1", "DMSO_rep2",
                          "TreatA_rep1", "TreatA_rep2")

# treatment labels
condition_count <- factor(c("DMSO", "DMSO", "TreatA", "TreatA"))

# population size (e.g. tumor volume)
V <- c(300, 280, 100, 120)

# assemble input
data <- list(
  data_count = data_count,
  condition_count = condition_count,
  V = V,
  condition_v = condition_count,
  cell_line = factor(colnames(data_count))
)

## Fit model (fast settings for example only)
## Note: iteration numbers are intentionally small to ensure the example runs quickly.
## For real analysis, increase iterations (e.g., 10000) and use multiple chains.

fit <- barmixRQTR(
  data = data,
  time_d = 10,
  control = list(
    chains = 1,
    iter_count = 50,
    iter_V = 50,
    cores = 1
  ),
  seed = 123
)

## ---------------------------
## Posterior Predictive Checks for Barcode Counts
```

```
## ---------------------------

ppc <- ppcBarcodes(fit)

## ---------------------------
## Estimate fraction ratios
## ---------------------------

ratio_res <- fractionRatio(
  model = fit,
  sampled_fraction = ppc$sampled_fraction
)

## plot
ratio_res$plot_ratio_fraction
```

`populationRatio` *Compute ratios of population size between treatments*

Description

Computes ratios of population size between treatment groups and a control group using posterior samples obtained from the barmixR model.

Usage

```
populationRatio(
  model,
  sampled_elements,
  q_up = 0.975,
  q_lo = 0.025,
  control_group = NULL
)
```

Arguments

`model`	An object of class `barmixR_fit` returned by `barmixRQTR` containing: • `condition_v`: Vector of treatment condition labels. • `K`: Number of treatment conditions. • `n_sam`: Number of posterior samples retained. • `in_vivo`: Logical indicating whether population size corresponds to tumor volume (*in vivo*) or confluency (*in vitro*).
`sampled_elements`	A named list of posterior samples of population size (tumor volume in *in vivo* experiments or confluency in *in vitro* assays), typically generated by `ppcPopulation()`.
`q_up`	Upper quantile threshold used to remove extreme values (default: 0.975).
`q_lo`	Lower quantile threshold used to remove extreme values (default: 0.025).
`control_group`	Optional name of the control treatment group.

Details

Population size corresponds to tumor volume in *in vivo* experiments or cellular confluency in *in vitro* assays. Posterior samples of population size are obtained from the population model component of `barmixRQTR()`, typically using `ppcPopulation()`. Ratios are calculated relative to the control group and summarized using log-transformed values.

Value

An object of class `barmixR_result` containing:

- `plot_ratio_population`: A `ggplot2` violin plot showing log-ratios of population size between treatments.

  \item \code{merged_data}: Data frame used for visualization.

  \item \code{li_sam_ratio_V}: List containing computed population
    size ratios.

  \item \code{type}: Result type ("population_ratio").

Examples

```
set.seed(1)

## ---------------------------
## Simulate small example data
## ---------------------------

# barcode count matrix (4 samples x 5 clones)
data_count <- matrix(
  c(
    20, 22, 24, 23, 21,   # DMSO_rep1
    21, 23, 25, 22, 20,   # DMSO_rep2
    26, 25, 22, 20, 18,   # TreatA_rep1
    25, 24, 23, 21, 19    # TreatA_rep2
  ),
  nrow = 4,
  byrow = TRUE
)

colnames(data_count) <- paste0("clone", 1:5)
rownames(data_count) <- c("DMSO_rep1", "DMSO_rep2",
                          "TreatA_rep1", "TreatA_rep2")

# treatment labels
condition_count <- factor(c("DMSO", "DMSO", "TreatA", "TreatA"))

# population size (e.g. tumor volume)
V <- c(300, 280, 100, 120)

# assemble input
data <- list(
  data_count = data_count,
  condition_count = condition_count,
  V = V,
  condition_v = condition_count,
```

```
  cell_line = factor(colnames(data_count))
)

## Fit model (fast settings for example only)
## Note: iteration numbers are intentionally small to ensure the example runs quickly.
## For real analysis, increase iterations (e.g., 10000) and use multiple chains.

fit <- barmixRQTR(
  data = data,
  time_d = 10,
  control = list(
    chains = 1,
    iter_count = 50,
    iter_V = 50,
    cores = 1
  ),
  seed = 123
)

## ----------------------------
## Posterior Predictive Check for Population Size
## ----------------------------

ppc_pop <- ppcPopulation(fit)

## ----------------------------
## Estimate ratios of population size
## ----------------------------
ratio_pop <- populationRatio(
  model = fit,
  sampled_elements = ppc_pop$sampled_population
)

## plot
ratio_pop $plot_ratio_population
```

---

ppcBarcodes *Posterior predictive checks for barcode composition*

Description

Performs posterior predictive checks (PPC) for barcode composition modeled using the Dirichlet-multinomial component of the barmixR model. Observed barcode counts are compared with predicted counts from the posterior distribution to assess model fit and uncertainty.

Usage

```
ppcBarcodes(model, q_up = 0.975, q_lo = 0.025)
```

Arguments

model	An object of class `barmixR_fit` returned by `barmixRQTR()` containing: • `fit_dir_mult`: Fitted Dirichlet-multinomial model.

- group: Numeric identifiers for treatment groups.
- condition_count: Treatment labels for barcode data.
- chains: Number of MCMC chains used during fitting.
- L: Number of cell lines.
- names_cell: Cell line identifiers.
- data_count: Observed barcode count matrix.
- K: Number of treatment groups.
- n_sam: Number of posterior samples retained.

q_up — Upper quantile threshold used for uncertainty visualization.

q_lo — Lower quantile threshold used for uncertainty visualization.

Details

Posterior predictive distributions are visualized using violin plots, with observed counts overlaid for each cell line and treatment condition.

Value

An object of class barmixR_diagnostic, containing:

- ppc_plot: A ggplot2 posterior predictive check visualization for barcode composition.

```
\item \code{sampled_fraction}: Posterior samples of barcode
  compositional fractions used for downstream analysis.

\item \code{type}: Result type ("ppc_barcodes").
```

Examples

```
set.seed(1)

## ---------------------------
## Simulate small example data
## ---------------------------

# barcode count matrix (4 samples x 5 clones)
data_count <- matrix(
  c(
    20, 22, 24, 23, 21,   # DMSO_rep1
    21, 23, 25, 22, 20,   # DMSO_rep2
    26, 25, 22, 20, 18,   # TreatA_rep1
    25, 24, 23, 21, 19    # TreatA_rep2
  ),
  nrow = 4,
  byrow = TRUE
)

colnames(data_count) <- paste0("clone", 1:5)
rownames(data_count) <- c("DMSO_rep1", "DMSO_rep2",
                          "TreatA_rep1", "TreatA_rep2")

# treatment labels
condition_count <- factor(c("DMSO", "DMSO", "TreatA", "TreatA"))
```

```
# population size (e.g. tumor volume)
V <- c(300, 280, 100, 120)

# assemble input
data <- list(
  data_count = data_count,
  condition_count = condition_count,
  V = V,
  condition_v = condition_count,
  cell_line = factor(colnames(data_count))
)

## Fit model (fast settings for example only)
## Note: iteration numbers are intentionally small to ensure the example runs quickly.
## For real analysis, increase iterations (e.g., 10000) and use multiple chains.

fit <- barmixRQTR(
  data = data,
  time_d = 10,
  control = list(
    chains = 1,
    iter_count = 50,
    iter_V = 50,
    cores = 1
  ),
  seed = 123
)

## ----------------------------
## Posterior Predictive Checks for Barcode Counts
## ----------------------------

ppc <- ppcBarcodes(fit)

# plot PPC
ppc$ppc_plot

# posterior sampled fractions
names(ppc$sampled_fraction)
```

`ppcPopulation` *Posterior predictive checks for population size*

Description

Performs posterior predictive checks (PPC) for population size data modeled in the barmixR framework. Population size corresponds to tumor volume in *in vivo* experiments or cellular confluency in *in vitro* assays.

Usage

```
ppcPopulation(model, q_up = 0.975, q_lo = 0.025)
```

Arguments

model	An object of class barmixR_fit returned by barmixRQTR containing: • condition_v: Vector of treatment labels. • fit_V: Fitted population size model (tumor volume in *in vivo* experiments or confluency in *in vitro* assays). • V: Observed population size measurements. • n_sam: Number of posterior samples retained. • VT0: Optional baseline population size values used for normalization.
q_up	Upper quantile threshold (unused; kept for API compatibility).
q_lo	Lower quantile threshold (unused; kept for API compatibility).

Details

Posterior predictive samples are generated from the population size component of the barmixR model fitted by barmixRQTR(). Observed values are compared with predicted values to assess model fit and uncertainty. Predictive distributions are visualized using violin plots with observed measurements overlaid.

Value

An object of class barmixR_diagnostic containing:

- ppc_plot: Posterior predictive check plot.

```
\item \code{ppc_plot_normalized}: Posterior predictive check plot
  for normalized population size values (if \code{VT0} is provided).

\item \code{ppc_data}: Data used for visualization.

\item \code{sampled_population}: Posterior predictive samples of
  population size.

\item \code{type}: Result type ("ppc_population").
```

Examples

```
set.seed(1)

## ---------------------------
## Simulate small example data
## ---------------------------

# barcode count matrix (4 samples x 5 clones)
data_count <- matrix(
  c(
    20, 22, 24, 23, 21,   # DMSO_rep1
    21, 23, 25, 22, 20,   # DMSO_rep2
    26, 25, 22, 20, 18,   # TreatA_rep1
    25, 24, 23, 21, 19    # TreatA_rep2
  ),
  nrow = 4,
  byrow = TRUE
)
```

```
colnames(data_count) <- paste0("clone", 1:5)
rownames(data_count) <- c("DMSO_rep1", "DMSO_rep2",
                          "TreatA_rep1", "TreatA_rep2")

# treatment labels
condition_count <- factor(c("DMSO", "DMSO", "TreatA", "TreatA"))

# population size (e.g. tumor volume)
V <- c(300, 280, 100, 120)

# assemble input
data <- list(
  data_count = data_count,
  condition_count = condition_count,
  V = V,
  condition_v = condition_count,
  cell_line = factor(colnames(data_count))
)

## Fit model (fast settings for example only)
## Note: iteration numbers are intentionally small to ensure the example runs quickly.
## For real analysis, increase iterations (e.g., 10000) and use multiple chains.

fit <- barmixRQTR(
  data = data,
  time_d = 10,
  control = list(
    chains = 1,
    iter_count = 50,
    iter_V = 50,
    cores = 1
  ),
  seed = 123
)

## ---------------------------
## Posterior Predictive Check for Population Size
## ---------------------------

ppc_pop <- ppcPopulation(fit)

## visualize PPC
ppc_pop$ppc_plot
```

`QTRDecision` *Rank treatments within each cell line using QTR boxplot summaries*

Description

Visualizes quantitative treatment resistance (QTR) by ranking treatments within each cell line using the summary statistics returned by `QTRresistance()`.

Usage

```
QTRDecision(
  model,
  summary_table,
  cell_order = NULL,
  ncol = 4,
  legend_text = NULL
)
```

Arguments

model	An object of class barmixR_fit returned by barmixRQTR containing names_cell.
summary_table	Data frame returned by QTRresistance(). It must contain the columns: • cell • treat • median • q25 • q75 • lower_whisker • upper_whisker
cell_order	Optional character vector giving the desired order of cell-line panels.
ncol	Number of columns in the combined panel layout.
legend_text	Optional annotation shown beneath the combined figure.

Details

Treatments are ordered within each cell line by the posterior median QTR estimate. The plot displays boxplot-style summaries using the interquartile range, whiskers, and median.

Value

An object of class barmixR_result containing:

- rank_plot: Combined multi-panel rank plot.
- rank_summary: Summary table used for plotting, including within-cell ranks.
- type: Result type ("QTR_decision").

Examples

```
set.seed(1)

## ----------------------------
## Simulate small example data
## ----------------------------

# barcode count matrix (4 samples x 5 clones)
data_count <- matrix(
  c(
    20, 22, 24, 23, 21,   # DMSO_rep1
    21, 23, 25, 22, 20,   # DMSO_rep2
    26, 25, 22, 20, 18,   # TreatA_rep1
```

```
    25, 24, 23, 21, 19    # TreatA_rep2
  ),
  nrow = 4,
  byrow = TRUE
)

colnames(data_count) <- paste0("clone", 1:5)
rownames(data_count) <- c("DMSO_rep1", "DMSO_rep2",
                          "TreatA_rep1", "TreatA_rep2")

# treatment labels
condition_count <- factor(c("DMSO", "DMSO", "TreatA", "TreatA"))

# population size (e.g. tumor volume)
V <- c(300, 280, 100, 120)

# assemble input
data <- list(
  data_count = data_count,
  condition_count = condition_count,
  V = V,
  condition_v = condition_count,
  cell_line = factor(colnames(data_count))
)

## Fit model (fast settings for example only)
## Note: iteration numbers are intentionally small to ensure the example runs quickly.
## For real analysis, increase iterations (e.g., 10000) and use multiple chains.

fit <- barmixRQTR(
  data = data,
  time_d = 10,
  control = list(
    chains = 1,
    iter_count = 50,
    iter_V = 50,
    cores = 1
  ),
  seed = 123
)

## ---------------------------
## Full downstream pipeline
## ---------------------------

# barcode PPC
ppc_bar <- ppcBarcodes(fit)

# fraction ratios
frac_ratio <- fractionRatio(
  model = fit,
  sampled_fraction = ppc_bar$sampled_fraction
)

# population PPC
ppc_pop <- ppcPopulation(fit)
```

```
# population ratios
pop_ratio <- populationRatio(
  model = fit,
  sampled_elements = ppc_pop$sampled_population
)

# resistance estimation
res <- QTRresistance(
  model = fit,
  li_sam_ratio_relative = frac_ratio$li_sam_ratio_relative,
  li_sam_ratio_V = pop_ratio$li_sam_ratio_V
)

## ---------------------------
## Decision plot
## ---------------------------

decision <- QTRDecision(
  model = fit,
  summary_table = res$summary_table
)

## visualize ranking
decision$rank_plot
```

`QTRheatmap` *Visualize quantitative treatment resistance (QTR) using bubble heatmaps*

Description

Visualizes quantitative treatment resistance (QTR) differences between cell lines and treatments using bubble heatmaps. The visualization is derived from posterior summaries of treatment resistance computed with `QTRresistance()`.

Usage

```
QTRheatmap(
  model,
  summary_table,
  q_up = 0.975,
  q_lo = 0.025,
  min_w = 1,
  max_w = 8,
  min_b = 1,
  max_b = 8
)
```

Arguments

`model`	An object of class `barmixR_fit` returned by `barmixRQTR` containing `condition_count`.

summary_table — Data frame returned by `QTRresistance()` containing summary statistics for treatment resistance. The table must include the columns:

- `cell`: Cell line identifier.
- `treat`: Treatment identifier.
- `mean`: Mean treatment resistance estimate.
- `CDF`: Cumulative distribution function value for resistance estimates.

q_up — Upper quantile threshold (unused; retained for API compatibility).

q_lo — Lower quantile threshold (unused; retained for API compatibility).

min_w — Minimum bubble size used in the within-treatment heatmap.

max_w — Maximum bubble size used in the within-treatment heatmap.

min_b — Minimum bubble size used in the treatment-level heatmap.

max_b — Maximum bubble size used in the treatment-level heatmap.

Details

Bubble size represents the magnitude of resistance differences, while color indicates the direction of resistance change.

Value

An object of class `barmixR_result` containing:

- `heatmap_within`: Bubble heatmap showing pairwise resistance differences between cell lines within each treatment.

```
\item \code{heatmap_treatment}: Bubble heatmap summarizing
  resistance differences across treatments.

\item \code{type}: Result type ("QTR_heatmap").
```

Examples

```
set.seed(1)

## ---------------------------
## Simulate small example data
## ---------------------------

# barcode count matrix (4 samples x 5 clones)
data_count <- matrix(
  c(
    20, 22, 24, 23, 21,   # DMSO_rep1
    21, 23, 25, 22, 20,   # DMSO_rep2
    26, 25, 22, 20, 18,   # TreatA_rep1
    25, 24, 23, 21, 19    # TreatA_rep2
  ),
  nrow = 4,
  byrow = TRUE
)

colnames(data_count) <- paste0("clone", 1:5)
rownames(data_count) <- c("DMSO_rep1", "DMSO_rep2",
                          "TreatA_rep1", "TreatA_rep2")
```

```
# treatment labels
condition_count <- factor(c("DMSO", "DMSO", "TreatA", "TreatA"))

# population size (e.g. tumor volume)
V <- c(300, 280, 100, 120)

# assemble input
data <- list(
  data_count = data_count,
  condition_count = condition_count,
  V = V,
  condition_v = condition_count,
  cell_line = factor(colnames(data_count))
)

## Fit model (fast settings for example only)
## Note: iteration numbers are intentionally small to ensure the example runs quickly.
## For real analysis, increase iterations (e.g., 10000) and use multiple chains.

fit <- barmixRQTR(
  data = data,
  time_d = 10,
  control = list(
    chains = 1,
    iter_count = 50,
    iter_V = 50,
    cores = 1
  ),
  seed = 123
)

## ---------------------------
## Full downstream pipeline
## ---------------------------

# barcode PPC
ppc_bar <- ppcBarcodes(fit)

# fraction ratios
frac_ratio <- fractionRatio(
  model = fit,
  sampled_fraction = ppc_bar$sampled_fraction
)

# population PPC
ppc_pop <- ppcPopulation(fit)

# population ratios
pop_ratio <- populationRatio(
  model = fit,
  sampled_elements = ppc_pop$sampled_population
)

# resistance estimation
res <- QTRresistance(
  model = fit,
```

```
  li_sam_ratio_relative = frac_ratio$li_sam_ratio_relative,
  li_sam_ratio_V = pop_ratio$li_sam_ratio_V
)

## ---------------------------
## Heatmap visualization
## ---------------------------

heat <- QTRheatmap(
  model = fit,
  summary_table = res$summary_table
)

## within-treatment heatmap
heat$heatmap_within

## treatment-level heatmap
heat$heatmap_treatment
```

`QTRresistance` *Estimate quantitative treatment resistance (QTR)*

Description

Estimates quantitative treatment resistance (QTR) for each cell line by combining posterior samples of barcode relative abundance and population size ratios derived from the barmixR model.

Usage

```
QTRresistance(
  model,
  li_sam_ratio_relative,
  li_sam_ratio_V,
  q_up = 0.975,
  q_lo = 0.025
)
```

Arguments

`model`	An object of class `barmixR_fit` returned by `barmixRQTR` containing: • `names_cell`: Vector of cell line identifiers. • `n_sam`: Number of posterior samples retained. • `L`: Number of cell lines. • `time_d`: Duration of the experiment.
`li_sam_ratio_relative`	A list of matrices containing posterior ratios of relative barcode abundance, typically generated by `fractionRatio()`.
`li_sam_ratio_V`	A list of vectors containing posterior ratios of population size (tumor volume in *in vivo* experiments or confluency in *in vitro* assays), typically generated by `populationRatio()`.
`q_up`	Upper quantile threshold used to remove extreme values (default: 0.975).
`q_lo`	Lower quantile threshold used to remove extreme values (default: 0.025).

Details

Population size corresponds to tumor volume in *in vivo* experiments or cellular confluency in *in vitro* assays. Treatment resistance is computed from posterior ratios of relative barcode abundance and population size, and summarized across posterior samples.

The function computes resistance distributions, cumulative distribution function (CDF) statistics, and summary values for each treatment and cell line.

Value

An object of class `barmixR_result` containing:

- `treatment_resistance`: Violin plot showing posterior distributions of quantitative treatment resistance (QTR).

```
\item \code{summary_table}: Data frame summarizing treatment
  resistance statistics for each cell line and treatment,
  including:
  \itemize{
    \item \code{cell}: Cell line identifier.
    \item \code{treat}: Treatment identifier.
    \item \code{mean}: Mean resistance estimate.
    \item \code{median}: Median resistance estimate.
    \item \code{sd}: Standard deviation of resistance estimates.
    \item \code{q25}: First quartile (25th percentile).
    \item \code{q75}: Third quartile (75th percentile).
   \item \code{lower_whisker}: Lower whisker of the resistance distribution.
   \item \code{upper_whisker}: Upper whisker of the resistance distribution.
    \item \code{CDF}: Cumulative distribution function value
      evaluated at zero resistance.
  }
```

- `type`: Result type ("QTR").

Examples

```
set.seed(1)

## ---------------------------
## Simulate small example data
## ---------------------------

# barcode count matrix (4 samples x 5 clones)
data_count <- matrix(
  c(
    20, 22, 24, 23, 21,   # DMSO_rep1
    21, 23, 25, 22, 20,   # DMSO_rep2
    26, 25, 22, 20, 18,   # TreatA_rep1
    25, 24, 23, 21, 19    # TreatA_rep2
  ),
  nrow = 4,
  byrow = TRUE
)

colnames(data_count) <- paste0("clone", 1:5)
rownames(data_count) <- c("DMSO_rep1", "DMSO_rep2",
```

```
                        "TreatA_rep1", "TreatA_rep2")

# treatment labels
condition_count <- factor(c("DMSO", "DMSO", "TreatA", "TreatA"))

# population size (e.g. tumor volume)
V <- c(300, 280, 100, 120)

# assemble input
data <- list(
  data_count = data_count,
  condition_count = condition_count,
  V = V,
  condition_v = condition_count,
  cell_line = factor(colnames(data_count))
)

## Fit model (fast settings for example only)
## Note: iteration numbers are intentionally small to ensure the example runs quickly.
## For real analysis, increase iterations (e.g., 10000) and use multiple chains.

fit <- barmixRQTR(
  data = data,
  time_d = 10,
  control = list(
    chains = 1,
    iter_count = 50,
    iter_V = 50,
    cores = 1
  ),
  seed = 123
)

## ---------------------------
## Downstream analysis pipeline
## ---------------------------

# barcode PPC
ppc_bar <- ppcBarcodes(fit)

# fraction ratios
frac_ratio <- fractionRatio(
  model = fit,
  sampled_fraction = ppc_bar$sampled_fraction
)

# population PPC
ppc_pop <- ppcPopulation(fit)

# population ratios
pop_ratio <- populationRatio(
  model = fit,
   sampled_elements = ppc_pop$sampled_population
)

## ---------------------------
## Estimate QTR
```

```
## ---------------------------

res <- QTRresistance(
  model = fit,
  li_sam_ratio_relative = frac_ratio$li_sam_ratio_relative,
  li_sam_ratio_V = pop_ratio$li_sam_ratio_V
)

## visualize resistance
res$treatment_resistance
```

Index

barmixR_fit, 6, *7*, *9*, *14*, *16*, *18*, *21*
barmixRQTR, 2, *6, 7*, *9*, *14*, *16*, *18*, *21*
BiocParallelParam, *4*

fractionRatio, 7

populationRatio, 9
ppcBarcodes, 11
ppcPopulation, 13

QTRDecision, 15
QTRheatmap, 18
QTRresistance, 21