Package 'BayesDissolution'

April 30, 2024

Title Bayesian Models for Dissolution Testing

Version 0.2.1

Description Fits Bayesian models (amongst others) to dissolution data sets that can be used for dissolution testing. The package was originally constructed to include only the Bayesian models outlined in Pourmohamad et al. (2022) <doi:10.1111/rssc.12535>. However, additional Bayesian and non-Bayesian models (based on bootstrapping and generalized pivotal quanties) have also been added. More models may be added over time.

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Imports coda, geoR, MCMCpack, mnormt, pscl, shiny, stats

Depends R (>= 2.10)

Suggests ggplot2, scales

NeedsCompilation no

Author Tony Pourmohamad [aut, cre],
Steven Novick [aut],
Robert Richardson [aut]

Maintainer Tony Pourmohamad <tpourmohamad@gmail.com>

Repository CRAN

Date/Publication 2024-04-30 19:20:02 UTC

R topics documented:

---

bmn *Bayesian Multivariate Normal Model for Dissolution Profile Modeling*

Description

This function implements the Bayesian multivariate normal model described in Pourmohamad et al (2022).

Usage

```
bmn(dis_data, B = 10000)
```

Arguments

`dis_data`	A data frame containing the dissolution data. The first column of the data frame should denote the group labels identifying whether a given dissolution belongs to the "reference" or "test" formulation group. For a given dissolution run, the remaining columns of the data frame contains the individual run's dissolution measurements sorted in time. Alternatively, the user may provide a data object of class dis_data containing the dissolution data. See the `make_dis_data()` function for the particular structure of the data object.
`B`	A positive integer specifying the number of posterior samples to draw. By default `B` is set to 10000.

Value

The function returns a list of B posterior samples for the following parameters:

- delta: A vector of posterior samples of delta as defined in Novick et. al 2015
- f2: A vector of posterior values of f2
- muR: A matrix of posterior samples for the reference group mean. Each row of the matrix corresponds to an observed time point, and each column of the matrix corresponds to a posterior sample.
- muT: A matrix of posterior samples for the test group mean. Each row of the matrix corresponds to an observed time point, and each column of the matrix corresponds to a posterior sample.

Note

You should always check MCMC diagnostics on the posterior samples before drawing conclusions.

References

Novick, S., Shen, Y., Yang, H., Peterson, J., LeBlond, D., and Altan, S. (2015). Dissolution Curve Comparisons Through the F2 Parameter, a Bayesian Extension of the f2 Statistic. Journal of Biopharmaceutical Statistics, 25(2):351-371.

Pourmohamad, T., Oliva Aviles, C.M., and Richardson, R. Gaussian Process Modeling for Dissolution Curve Comparisons. Journal of the Royal Statistical Society, Series C, 71(2):331-351.

Examples

```
### dis_data comes loaded with the package
### We set B = 1000 to obtain 1000 posterior samples
B <- 1000
post <- bmn(dis_data, B = B)

### We can check how well the posterior samples of the means are mixing by
### plotting the individual chains by time point
burnin <- B * 0.1     # A 10% burn-in
post$mu_R <- post$muR[,-(1:burnin)]
post$mu_T <- post$muT[,-(1:burnin)]

N <- B - burnin      # Number of posterior samples after burn-in
chains <- data.frame(samples = rep(c(1:N, 1:N), each = ncol(dis_data) - 1),
                     group = rep(c("muR", "muT"), each = (ncol(dis_data) - 1) * N),
                    timepoint = paste("Time Point", rep(1:(ncol(dis_data) - 1), 2 * N)),
                     values = c(c(post$mu_R), c(post$mu_T)))

g <- ggplot2::ggplot(chains, ggplot2::aes(samples, values)) +
                     ggplot2::geom_line() +
                     ggplot2::labs(x = "Iterations", y = "Posterior Sample Values") +
                     ggplot2::facet_wrap(group ~ timepoint) +
                     ggplot2::theme(text = ggplot2::element_text(size = 16))

### If we want to calculate the Pr(f2 > 50)
post$f2<- post$f2[-(1:burnin)]
prob <- sum(post$f2 > 50) / (B - burnin)

### Or if we want calculate a 95% credible interval for f2
alpha <- 0.05
f2_cred <- c(quantile(post$f2, alpha / 2),quantile(post$f2, 1 - alpha / 2))
```

`dissplot` *Dissolution Data Plot*

Description

This function plots dissolution data sets.

Usage

```
dissplot(
  dis_data,
  tp = NULL,
  pch = c(19, 17),
  color = c("gray65", "black"),
  groups = c("Reference", "Test"),
  legend_location = "bottomright",
  xlab = "Time Points",
  ylab = "Percentage Dissolved",
  mean = FALSE,
  var = FALSE,
  var_label = TRUE,
  ...
)
```

Arguments

dis_data	A data frame containing the dissolution data. The first column of the data frame should denote the group labels identifying whether a given dissolution belongs to the "reference" or "test" formulation group. For a given dissolution run, the remaining columns of the data frame contains the individual run's dissolution measurements sorted in time. Alternatively, the user may provide a data object of class dis_data containing the dissolution data. See the `make_dis_data()` function for the particular structure of the data object.
tp	An optional vector of time points at which the dissolution data is measured at.
pch	A vector of two elements specifying the plotting character to use for each group. If only one value is passed then the plotting character is the same for both groups.
color	A vector of two elements specifying the color in the plot to associate with each group. If only one value is passed then the color choice is the same for both groups.
groups	A vector of two elements specifying the name to use for each group in the plot.
legend_location	A string that denotes the location of where the legend should appear. Possible options are "left", "top", "bottom", "right", and any logical combination of the four, e.g., "bottomright" or "topleft".
xlab	A string specifying the x-axis label.
ylab	A string specifying the y-axis label.
mean	logical; if `TRUE`, plot the connected mean dissolution values for each group
var	logical; if `TRUE`, calculate the variance of the dissolution data at each time point for each group. The values are placed at the top of the plot over the corresponding time point.

var_label	logical; if TRUE, use the group labels when printing out the variances.
...	other graphical parameters commonly found in plot.default

Value

The function returns a plot of the dissolution data.

Examples

```
### dis_data comes loaded with the package
dissplot(dis_data)
```

---

dis_data *A dissolution data set taken from Ocana et al. (2009).*

Description

A dissolution data set that consists of dissolution measurements taken on oral tablets made with metoclopramide hydrochloride. Of interest is to test the similarity of metoclopramide hydrochloride tablets made with and without the ingredient tensioactive.

Usage

```
dis_data
```

Format

A data frame with 24 rows and 9 columns:

group An indicator of whether the dissolution run came from the reference or test group

X1 The first time point at which measurements are made at.

X2 The second time point at which measurements are made at.

X3 The third time point at which measurements are made at.

X4 The fourth time point at which measurements are made at.

X5 The fifth time point at which measurements are made at.

X6 The sixth time point at which measurements are made at.

X7 The seventh time point at which measurements are made at.

X8 The eight time point at which measurements are made at. ...

Source

Ocana et al. (2009) doi:10.1016/j.chemolab.2009.07.010

`f2bayes` *Calculation of a Bayesian 100*prob% credible interval for the F2 parameter*

Description

This function calculates a 100*prob% credible interval for the F2 parameter using Bayesian methods. The model assumes a version of the Jerffreys' prior with a pooled variance-covariance matrix from based on the reference and test data sets. See Novick (2015) for more details of the model.

Usage

```
f2bayes(
  dis_data,
  prob = 0.9,
  B = 1000,
  ci.type = c("quantile", "HPD"),
  get.dist = FALSE
)
```

Arguments

`dis_data`	A data frame containing the dissolution data. The first column of the data frame should denote the group labels identifying whether a given dissolution belongs to the "reference" or "test" formulation group. For a given dissolution run, the remaining columns of the data frame contains the individual run's dissolution measurements sorted in time. Alternatively, the user may provide a data object of class dis_data containing the dissolution data. See the `make_dis_data()` function for the particular structure of the data object.
`prob`	The probability associated with the credible interval. A value between 0 and 1.
`B`	A positive integer specifying the number of Monte Carlo samples.
`ci.type`	The type of credible interval to report. Specifying `quantile` returns a credible interval based on the posterior sample quantiles of the F2 distribution. Specifying `HPD` returns a highest posterior density interval.
`get.dist`	logical; if `TRUE`, returns the posterior samples of the F2 distribution.

Value

The function returns a 100*prob% credible interval for the F2 parameter calculated from the observed dissolution data.

Note

Use the `plotdiss()` or `ggplotdiss()` function to visually check if it's appropriate to calculate the f2 statistic.

References

Novick, S., Shen, Y., Yang, H., Peterson, J., LeBlond, D., and Altan, S. (2015). Dissolution Curve Comparisons Through the F2 Parameter, a Bayesian Extension of the f2 Statistic. Journal of Biopharmaceutical Statistics, 25(2):351-371.

Pourmohamad, T., Oliva Aviles, C.M., and Richardson, R. Gaussian Process Modeling for Dissolution Curve Comparisons. Journal of the Royal Statistical Society, Series C, 71(2):331-351.

Examples

```
### dis_data comes loaded with the package
f2bayes(dis_data, prob = 0.9, B = 1000)
```

`f2bca` *Calculation of a biased-corrected and accepted 100*level% confidence interval for the F2 parameter*

Description

This function calculates a 100*level% confidence interval for the F2 parameter using biased-correctd and accelerated (BCa) boostrap

Usage

```
f2bca(
  dis_data,
  level = 0.9,
  B = 1000,
  ci.type = c("quantile", "HPD"),
  get.dist = FALSE
)
```

Arguments

`dis_data`	A data frame containing the dissolution data. The first column of the data frame should denote the group labels identifying whether a given dissolution belongs to the "reference" or "test" formulation group. For a given dissolution run, the remaining columns of the data frame contains the individual run's dissolution measurements sorted in time. Alternatively, the user may provide a data object of class dis_data containing the dissolution data. See the `make_dis_data()` function for the particular structure of the data object.
`level`	The confidence level. A value between 0 and 1.
`B`	A positive integer specifying the number of bootstrap samples.
`ci.type`	The type of confidence interval to report. Specifying `quantile` returns a bootstrap confidence interval based on the sample quantiles. Specifying `HPD` returns a highest density region interval.
`get.dist`	logical; if `TRUE`, returns the posterior samples of the F2 distribution.

Value

The function returns a 100*level% confidence interval for the F2 parameter calculated from the observed dissolution data.

Note

Use the `plotdiss()` or `ggplotdiss()` function to visually check if it's appropriate to calculate the f2 statistic.

References

Liu, S. and Cai, X. and Shen, M. and Tsong, Y. (2023). In vitro dissolution profile comparison using bootstrap bias corrected similarity factor, f2. Journal of Biopharmaceutical Statistics, 34(1):78-89.

Examples

```
### dis_data comes loaded with the package
f2bca(dis_data, level = 0.9, B = 1000)
```

---

f2boot *Calculation of a bootstrap 100*level% confidence interval for the F2 parameter*

Description

This function calculates a 100*level% confidence interval for the F2 parameter using a nonparametric bootstrap

Usage

```
f2boot(
  dis_data,
  level = 0.9,
  B = 1000,
  ci.type = c("quantile", "HPD"),
  get.dist = FALSE
)
```

Arguments

`dis_data`	A data frame containing the dissolution data. The first column of the data frame should denote the group labels identifying whether a given dissolution belongs to the "reference" or "test" formulation group. For a given dissolution run, the remaining columns of the data frame contains the individual run's dissolution measurements sorted in time. Alternatively, the user may provide a data object of class dis_data containing the dissolution data. See the `make_dis_data()` function for the particular structure of the data object.

level	The confidence level. A value between 0 and 1.
B	A positive integer specifying the number of bootstrap samples.
ci.type	The type of confidence interval to report. Specifying quantile returns a bootstrap confidence interval based on the sample quantiles. Specifying HPD returns a highest density region interval.
get.dist	logical; if TRUE, returns the posterior samples of the F2 distribution.

Value

The function returns a 100*level% confidence interval for the F2 parameter calculated from the observed dissolution data.

Note

Use the plotdiss() or ggplotdiss() function to visually check if it's appropriate to calculate the f2 statistic.

References

Liu, S. and Cai, X. and Shen, M. and Tsong, Y. (2023). In vitro dissolution profile comparison using bootstrap bias corrected similarity factor, f2. Journal of Biopharmaceutical Statistics, 34(1):78-89.

Examples

```
### dis_data comes loaded with the package
f2boot(dis_data, level = 0.9, B = 1000)
```

---

f2calc *Calculation of the f2 Statistic*

Description

This function calculates the f2 statistic as described in Moore and Flanner (1996).

Usage

```
f2calc(dis_data)
```

Arguments

dis_data	A data frame containing the dissolution data. The first column of the data frame should denote the group labels identifying whether a given dissolution belongs to the "reference" or "test" formulation group. For a given dissolution run, the remaining columns of the data frame contains the individual run's dissolution measurements sorted in time. Alternatively, the user may provide a data object of class dis_data containing the dissolution data. See the make_dis_data() function for the particular structure of the data object.

Value

The function returns the f2 statistic calculated from the observed dissolution data.

Note

Use the `plotdiss()` or `ggplotdiss()` function to visually check if it's appropriate to calculate the f2 statistic.

References

Moore, J.W. and Flanner, H.H. (1996). Mathematical comparison of distribution profiles. Pharmaceutical Technology, 20(6):64-74.

Examples

```
### dis_data comes loaded with the package
f2calc(dis_data)
```

---

f2gpq *Calculation of a generalized pivotal quantity 100*level% confidence interval for the F2 parameter*

Description

This function calculates a 100*level% confidence interval for the F2 parameter using generalized pivotal quantity methods based on a two variance component model with means for Time x Group, i.e., Dissolution ~ Time x Group + (1|Tablet:Group).

Usage

```
f2gpq(
  dis_data,
  level = 0.9,
  B = 10000,
  ci.type = c("quantile", "HPD"),
  get.dist = FALSE
)
```

Arguments

`dis_data`	A data frame containing the dissolution data. The first column of the data frame should denote the group labels identifying whether a given dissolution belongs to the "reference" or "test" formulation group. For a given dissolution run, the remaining columns of the data frame contains the individual run's dissolution measurements sorted in time. Alternatively, the user may provide a data object of class dis_data containing the dissolution data. See the `make_dis_data()` function for the particular structure of the data object.

level	The confidence level. A value between 0 and 1.
B	The number of generalized pivotal quantity samples.
ci.type	The type of confidence interval to report. Specifying quantile returns a bootstrap confidence interval based on the sample quantiles. Specifying HPD returns a highest density region interval.
get.dist	logical; if TRUE, returns the posterior samples of the F2 distribution.

Value

The function returns a 100*level% confidence interval for the F2 parameter calculated from the observed dissolution data.

Note

Use the plotdiss() or ggplotdiss() function to visually check if it's appropriate to calculate the f2 statistic.

Examples

```
### dis_data comes loaded with the package
f2gpq(dis_data, level = 0.9, B = 10000)
```

---

f2pbs *Calculation of a parametric bootstrap 100*level% confidence interval for the F2 parameter*

Description

This function calculates a 100*level% confidence interval for the F2 parameter using a parametric bootstrap based on a two variance component model with means for Time x Group, i.e., Dissolution ~ Time x Group + (1|Tablet:Group).

Usage

```
f2pbs(
  dis_data,
  level = 0.9,
  B = 1000,
  ci.type = c("quantile", "HPD"),
  get.dist = FALSE
)
```

Arguments

dis_data	A data frame containing the dissolution data. The first column of the data frame should denote the group labels identifying whether a given dissolution belongs to the "reference" or "test" formulation group. For a given dissolution run, the remaining columns of the data frame contains the individual run's dissolution measurements sorted in time. Alternatively, the user may provide a data object of class dis_data containing the dissolution data. See the `make_dis_data()` function for the particular structure of the data object.
level	The confidence level. A value between 0 and 1.
B	A positive integer specifying the number of bootstrap samples.
ci.type	The type of confidence interval to report. Specifying `quantile` returns a bootstrap confidence interval based on the sample quantiles. Specifying `HPD` returns a highest density region interval.
get.dist	logical; if `TRUE`, returns the posterior samples of the F2 distribution.

Value

The function returns a 100*level% confidence interval for the F2 parameter calculated from the observed dissolution data.

Note

Use the `plotdiss()` or `ggplotdiss()` function to visually check if it's appropriate to calculate the f2 statistic.

Examples

```
### dis_data comes loaded with the package
f2pbs(dis_data, level = 0.9, B = 1000)
```

---

ggdissplot *Dissolution Data Plot*

Description

Minimalist ggplot function for plotting dissolution data sets.

Usage

```
ggdissplot(dis_data, show.mean = FALSE, show.SD = FALSE)
```

Arguments

dis_data	A data frame containing the dissolution data. The first column of the data frame should denote the group labels identifying whether a given dissolution belongs to the "reference" or "test" formulation group. For a given dissolution run, the remaining columns of the data frame contains the individual run's dissolution measurements sorted in time. Alternatively, the user may provide a data object of class dis_data containing the dissolution data. See the `make_dis_data()` function for the particular structure of the data object.
show.mean	logical; if `TRUE`, plot the connected mean dissolution values for each group.
show.SD	logical; if `TRUE`, calculate the variance of the dissolution data at each time point for each group. The values are placed at the top of the plot over the corresponding time point.

Value

The function returns a plot of the dissolution data.

Examples

```
### dis_data comes loaded with the package
ggdissplot(dis_data)
```

---

hgp *Hierarchical Gaussian Process Model for Dissolution Profile Modeling*

Description

This function implements the Bayesian hierarchical Gaussian process model described in Pourmohamad et al (2022).

Usage

```
hgp(
  dis_data,
  locs,
  B = 1000,
  n_interp = 30,
  control = list(),
  adaptive = FALSE
)
```

Arguments

dis_data	A data frame containing the dissolution data. The first column of the data frame should denote the group labels identifying whether a given dissolution belongs to the "reference" or "test" formulation group. For a given dissolution run, the remaining columns of the data frame contains the individual run's dissolution measurements sorted in time. Alternatively, the user may provide a data object of class dis_data containing the dissolution data. See the `make_dis_data()` function for the particular structure of the data object.
locs	A vector in ascending order that corresponds to each time point the dissolution data was measured at.
B	A positive integer specifying the number of posterior samples to draw. By default `B` is set to 10000.
n_interp	An integer value specifying the number of time points to interpolate at. This sets the interploated points to be to `seq(1st time point, last time point, length = n_interp)`.
control	An optional list of priors and initial values, otherwise the default values/strategies found in Pourmohamad et al (2022) will be used. More specifically, `control` can be used to define the following settings: • `sigma2_starting`: starting value for sigma^2 • `tau2_starting`: starting value for tau^2 • `phi_starting`: starting value for phi • `psi_starting`: starting value for psi • `sigma2_alpha` and `sigma2_beta`: parameters for the inverse gamma prior for sigma^2 • `tau2_alpha` and `tau2_beta`: parameters for the inverse gamma prior for tau^2 • `phi_alpha` and `phi_beta`: parameters for the gamma prior for phi • `psi_alpha` and `psi_beta`: parameters for the gamma prior for psi • `prop_phi`: proposal variance for the parameter phi • `prop_psi`: proposal variance for the parameter psi
adaptive	logical; an option for using adaptive MCMC. If `adaptive = TRUE`, this will replace both `prop_phi` and `prop_psi` by using past MCMC draws to inform the proposal variance.

Value

The function returns a list of summary statistics and B posterior samples for parameters of the model. More specifically it returns:

- delta: The average delta value over the posterior samples of delta. The definition of delta is given in Novick et. al 2015.
- f2: The average f2 value over the posterior samples of f2.
- mcmc_chains: A list of posterior samples for delta, f2, the mean parameters (`mu_pars`), and the covariance parameters (`cov_pars`).

Note

You should always check MCMC diagnostics on the posterior samples before drawing conclusions. Likewise, plots of the predicted dissolution curves should also be checked to evaluate if the model fit to the observed data seems reasonable.

References

Novick, S., Shen, Y., Yang, H., Peterson, J., LeBlond, D., and Altan, S. (2015). Dissolution Curve Comparisons Through the F2 Parameter, a Bayesian Extension of the f2 Statistic. Journal of Biopharmaceutical Statistics, 25(2):351-371.

Pourmohamad, T., Oliva Aviles, C.M., and Richardson, R. Gaussian Process Modeling for Dissolution Curve Comparisons. Journal of the Royal Statistical Society, Series C, 71(2):331-351.

Examples

```
### dis_data comes loaded with the package
### We set B = 100 to obtain 100 posterior samples, you probably want to run it
### longer for say, B = 100000, but B = 100 runs fast for illustrative purposes
### and passing CRAN checks
B <- 100

tp <- seq(10, 80, 10) # Time points
burnin <- B * 0.1     # A 10% burn-in
thin <- 10            # Keep every 10th sample, i.e., thinning
post <- hgp(dis_data, locs = tp, B = B, n_interp = 100)

### Example: Removing burn-in and then thinning the posterior samples for the covariance parameters
###          and then plotting the chains
phi <- post$mcmc_chains$cov_pars$phi[-c(1:burnin)]
phi <- phi[seq(1, (B-burnin), thin)]
psi <- post$mcmc_chains$cov_pars$psi[-c(1:burnin)]
psi <- psi[seq(1, (B-burnin), thin)]
sigma_R <- post$mcmc_chains$cov_pars$sigma_R[-c(1:burnin)]
sigma_R <- sigma_R[seq(1, (B-burnin), thin)]
sigma_T <- post$mcmc_chains$cov_pars$sigma_T[-c(1:burnin)]
sigma_T <- sigma_T[seq(1, (B-burnin), thin)]
tau <- post$mcmc_chains$cov_pars$tau[-c(1:burnin)]
tau <- tau[seq(1, (B-burnin), thin)]

chains <- data.frame( # Data frame holding posterior samples
samples = rep(1:((B-burnin)/thin), times = 5),
parameter = rep(c("phi", "psi", "sigma_R", "sigma_T", "tau"),
                each = (B-burnin)/thin),
values = c(phi, psi, sigma_R, sigma_T, tau))
chains$parameter <- factor(chains$parameter,
                           labels = c(expression(phi),
                                      expression(psi),
                                      expression(sigma[R]),
                                      expression(sigma[T]),
                                      expression(tau)))
ggplot2::ggplot(chains, ggplot2::aes(samples, values)) +
```

```
 ggplot2::geom_line() +
 ggplot2::labs(x = "Iterations", y = "Posterior Sample Values") +
 ggplot2::facet_wrap(~parameter, scales = "free",
             labeller = ggplot2::label_parsed) +
 ggplot2::theme(text = ggplot2::element_text(size = 22))

ggplot2::ggplot(chains, ggplot2::aes(values)) +
 ggplot2::geom_density() +
 ggplot2::labs(x = "Values", y = "Posterior Density") +
 ggplot2::facet_wrap(~parameter, scales = "free",
            labeller = ggplot2::label_parsed) +
 ggplot2::theme(text = ggplot2::element_text(size = 22))

### Plotting the predicted dissolution profiles
dissplot(dis_data, tp)
grid <- sort(c(tp, seq(min(tp), max(tp), length = 100)))
grid1 <- (1:B)[-(1:burnin)][seq(1, (B-burnin), thin)]
grid2 <- ((B+1):(2*B))[-(1:burnin)][seq(1, (B-burnin), thin)]
lines(grid, apply(post$mcmc_chains$mu_pars[,grid1], 1, mean),
      col = "gray65", lwd = 2)
lines(grid, apply(post$mcmc_chains$mu_pars[,grid2], 1, mean),
      col = "black", lwd = 2)
lower <- apply(post$mcmc_chains$mu_pars[,grid1], 1,
               quantile, prob = 0.025)
upper <- apply(post$mcmc_chains$mu_pars[,grid1], 1,
               quantile, prob = 0.975)
polygon(c(grid, rev(grid)), c(lower, rev(upper)),
        col = scales::alpha("gray65",.2), border = NA)
lower <- apply(post$mcmc_chains$mu_pars[,grid2], 1,
              quantile, prob = 0.025)
upper <- apply(post$mcmc_chains$mu_pars[,grid2], 1,
               quantile, prob = 0.975)
polygon(c(grid, rev(grid)), c(lower, rev(upper)),
        col = scales::alpha("black",.2), border = NA)

### If we want to calculate the Pr(f2 > 50 & delta < 15)
prob <- sum(post$mcmc_chains$f2[grid1] > 50 &
            post$mcmc_chains$delta[grid1] < 15) / ((B - burnin)/thin)
```

`make_dis_data` *Class dis_data creation*

Description

This function creates a data object of class dis_data.

Usage

```
make_dis_data(yRef, yTest)
```

Arguments

yRef	A data frame or matrix containing the dissolution data for the reference group data. The rows of the data set correspond to the individual dissolution runs. The columns of the data frame contains the individual run's dissolution measurements sorted in time.
yTest	A data frame or matrix containing the dissolution data for the test group data. The rows of the data set correspond to the individual dissolution runs. The columns of the data frame contains the individual run's dissolution measurements sorted in time.

Value

The function returns a data object of class dis_data.

Examples

```
### dis_data comes loaded with the package
### but need to update dis_data to be an object of class dis_data
new_dis_data <- make_dis_data(yRef = dis_data[dis_data$group == "Reference", -1],
                              yTest = dis_data[dis_data$group == "Test", -1])
```

---

process_results *Helper function for processing results*

Description

This function helps process the final results for the different f2 functions (e.g., f2bayes).

Usage

```
process_results(
  name,
  f2.dist,
  ci.type = c("quantile", "HPD"),
  level,
  get.dist = FALSE
)
```

Arguments

name	A character string denoting the type of method used to calculate the interval.
f2.dist	A vector of samples for the F2 parameter or f2 statistic.
ci.type	The type of confidence, or credible, interval to return. The option quantile returns sample quantile based intervals, while the option HPD returns intervals based on highest density regions.

level	The confidence level or probability associated with the confidence or credible interval, respectively. Must be a value between 0 and 1.
get.dist	logical; if TRUE, returns the samples of the distribution.

Value

The function returns a data object of class dis_data.

Examples

```
### dis_data comes loaded with the package
out1 <- f2bayes(dis_data, prob = 0.9, B = 1000, get.dist = TRUE)

out2 <- process_results("bayes", out1$f2.dist, level = 0.9)

### out1 and out2 should contain the results for the info and intervals
```

---

runExample *Run BayesDissolution Shiny App*

Description

Runs a shiny application for calculating the different confidence and credible intervals for the F2 parameter. The different intervals are constructed using the `f2bayes()`, `f2bca()`, `f2boot()`, `f2gpq()`, and `f2pbs()` functions. The shiny application comes preloaded with an example excel data set based on the `dis_data` data set.

Usage

```
runExample()
```

Examples

```
### The function requires no input to run
if(FALSE){ ## Make me TRUE to run
  runExample()
}
```

Index