Package 'ProteoBayes'

July 19, 2023

Title Bayesian Statistical Tools for Quantitative Proteomics

Version 1.0.0

Description Bayesian toolbox for quantitative proteomics. In particular, this
package provides functions to generate synthetic datasets, execute Bayesian
differential analysis methods, and display results as, described in the
associated article Marie Chion and Arthur Leroy (2023) <arXiv:2307.08975>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

Imports dplyr, ggplot2, magrittr, mvtnorm, tibble, tidyr, rlang,
extraDistr

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

URL https://mariechion.github.io/ProteoBayes/

NeedsCompilation no

Author Arthur Leroy [aut, cre] (<https://orcid.org/0000-0003-0806-8934>),
Marie Chion [aut] (<https://orcid.org/0000-0001-8956-8388>)

Maintainer Arthur Leroy <arthur.leroy.pro@gmail.com>

Repository CRAN

Date/Publication 2023-07-19 15:20:05 UTC

R topics documented:

---

identify_diff   *Identify posterior mean differences*

Description

Compute a criterion based on Credible Intervals (CI) to determine whether the posterior t-distributions of groups should be considered different enough to deserve further examination. Two groups are considered probably 'distinct' if the Credible Interval of level `CI_level` of their respective posterior t-distributions do not overlap.

Usage

```
identify_diff(posterior, CI_level = 0.05, nb_samples = 1000)
```

Arguments

posterior   A tibble, typically coming from a `posterior_mean()` function, containing the parameters of the multivariate posterior t-distributions for the mean of the considered groups and draws for each peptide.

CI_level   A number, defining the order of quantile chosen to assess differences between groups.

nb_samples   A number (optional), indicating the number of samples to draw from the posteriors for computing mean and credible intervals . Only used if `posterior` is multivariate, typically coming from a `multi_posterior_mean()` function.

Value

A tibble, indicating which peptides and groups seem to be different

Examples

```
TRUE
```

---

multi_posterior_mean   *Multivariate posterior distribution of the means*

Description

Compute the multivariate posterior distribution of the means between multiple groups, for multiple correlated peptides. The function accounts for multiple imputations through the `Draw` identifier in the dataset.

Usage

```
multi_posterior_mean(
  data,
  mu_0 = NULL,
  lambda_0 = 1,
  Sigma_0 = NULL,
  nu_0 = 10,
  vectorised = FALSE
)
```

Arguments

data	A tibble or data frame containing imputed data sets for all groups. Required columns: `Peptide`, `Group`, `Sample`, `Output`. If missing data have been estimated from multiple imputations, each imputation should be identified in an optional `Draw` column.
mu_0	A vector, corresponding to the prior mean. If NULL, all groups are initialised with the same empirical mean for each peptide.
lambda_0	A number, corresponding to the prior covariance scaling parameter.
Sigma_0	A matrix, corresponding to the prior covariance parameter. If NULL, the identity matrix will be used by default.
nu_0	A number, corresponding to the prior degrees of freedom.
vectorised	A boolean, indicating whether we should used a vectorised version of the function. Default when nb_peptides < 30. If nb_peptides > 30, there is a high risk that the vectorised version would be slower.

Value

A tibble providing the parameters of the multivariate posterior t-distribution for the mean of the considered groups and draws for each peptide.

Examples

```
TRUE
```

---

plot_distrib *Plot the posterior distribution of the difference of means*

Description

Display the posterior distribution of the difference of means between two groups for a specific peptide. If only one group is provide, the function display the posterior distribution of the mean for this specific group instead. The function provides additional tools to represent information to help inference regarding the difference between groups (reference at 0 on the x-axis, probability of group1 > group2 and conversely).

Usage

```
plot_distrib(
  sample_distrib,
  group1 = NULL,
  group2 = NULL,
  peptide = NULL,
  prob_CI = 0.95,
  show_prob = TRUE,
  mean_bar = TRUE,
  index_group1 = NULL,
  index_group2 = NULL
)
```

Arguments

sample_distrib	A data frame, typically coming from the sample_distrib() function, containing the following columns: Peptide, Group and Sample. This argument should contain the empirical posterior distributions to be displayed.
group1	A character string, corresponding to the name of the group for which we plot the posterior distribution of the mean. If NULL (default), the first group appearing in sample_distrib is displayed. If group2 is provided, the posterior difference of the groups is displayed instead.
group2	A character string, corresponding to the name of the group we want to compare to group1. If NULL (default), only the posterior distribution of the mean for group1 is displayed.
peptide	A character string, corresponding to the name of the peptide for which we plot the posterior distribution of the mean. If NULL (default), only the first appearing in sample_distrib is displayed.
prob_CI	A number, between 0 and 1, corresponding the level of the Credible Interval (CI), represented as side regions (in red) of the posterior distribution. The default value (0.95) display the 95% CI, meaning that the central region (in blue) contains 95% of the probability distribution of the mean.
show_prob	A boolean, indicating whether we display the label of probability comparisons between the two groups.
mean_bar	A boolean, indicating whether we display the vertical bar corresponding to 0 on the x-axis (when comparing two groups), of the mean value of the distribution (when displaying a unique group).
index_group1	A character string, used as the index of group1 in the legends. If NULL (default), group1 is used.
index_group2	A character string, used as the index of group2 in the legends. If NULL (default), group2 is used.

Value

Plot of the required posterior distribution.

Examples

```
TRUE
```

---

posterior_mean	*Posterior distribution of the means*

Description

Compute the posterior distribution of the means between multiple groups. All peptides are considered independent from one another.

Usage

```
posterior_mean(data, mu_0 = NULL, lambda_0 = 1, beta_0 = 1, alpha_0 = 1)
```

Arguments

data	A tibble or data frame containing imputed data sets for all groups. Required columns: Peptide, Output, Group, Sample.
mu_0	A vector, corresponding to the prior mean.
lambda_0	A number, corresponding to the prior covariance scaling parameter.
beta_0	A matrix, corresponding to the prior covariance parameter.
alpha_0	A number, corresponding to the prior degrees of freedom.

Value

A tibble providing the empirical posterior distribution for the

Examples

```
TRUE
```

---

sample_distrib	*Sample from a t-distribution*

Description

Sample from a (possibly multivariate) t-distribution. This function can be used to sample both from a prior or posterior, depending on the value of parameters provided.

Usage

```
sample_distrib(posterior, nb_sample = 1000)
```

Arguments

posterior | A tibble or data frame, detailing for each `Peptide` and each `Group`, the value of the t-distribution parameters. The expected format is typically a return from a posterior_mean() function. Expected columns in the univariate case: `mu`, `lambda`, `alpha`, `beta`. Expected columns in the multivariate case: `mu`, `lambda`, `Sigma`, `nu`.

nb_sample | A number, indicating the number of samples generated for each couple `Peptide`-`Group`.

Value

A tibble containing the `Peptide`, `Group` and `Sample` columns. The samples of each `Peptide`-`Group` couple provide an empirical t-distribution that can be used to compute and display differences between groups.

Examples

```
TRUE
```

---

simu_db | *Generate a synthetic dataset tailored for ProteoBayes*

Description

Simulate a complete training dataset, which may be representative of various applications. Several flexible arguments allow adjustment of the number of peptides, of groups, and samples in each experiment. The values of several parameters controlling the data generation process can be modified.

Usage

```
simu_db(
  nb_peptide = 5,
  nb_group = 2,
  nb_sample = 5,
  multi_imp = FALSE,
  nb_draw = 5,
  range_peptide = c(0, 50),
  diff_group = 3,
  var_sample = 2,
  var_draw = 1
)
```

Arguments

nb_peptide	An integer, indicating the number of peptides in the data.
nb_group	An integer, indicating the number of groups/conditions.
nb_sample	An integer, indicating the number of samples in the data for each peptide (i.e the repetitions of the same experiment).
multi_imp	A boolean, indicating whether multiple imputations have been applied to obtain the dataset.
nb_draw	A number, indicating the number of imputation procedures applied to obtain this dataset.
range_peptide	A 2-sized vector, indicating the range of values from which to pick a mean value for each peptide.
diff_group	A number, indicating the mean difference between consecutive groups
var_sample	A number, indicating the noise variance for each new sample of a peptide.
var_draw	A number, indicating the noise variance for each imputation draw.

Value

A full dataset of synthetic data.

Examples

```
## Generate a dataset with 5 peptides in each of the 2 groups, observed for
##  3 different samples
data = simu_db(nb_peptide = 5, nb_group = 2, nb_sample = 3)

## Generate a dataset with 3 peptides in each of the 3 groups, observed for
## 4 different samples, for which 5 imputation draws are available.
data = simu_db(nb_peptide = 3, nb_group = 3, nb_sample = 4, nb_draw = 5)
```

---

vectorised_multi	*Vectorised version of multi_posterior_mean()*

Description

Alternative vectorised version, highly efficient when nb_peptide < 30.

Usage

```
vectorised_multi(data, mu_0 = NULL, lambda_0 = 1, Sigma_0 = NULL, nu_0 = 10)
```

Arguments

`data`	A tibble or data frame containing imputed data sets for all groups. Required columns: `Peptide`, `Group`, `Sample`, `Output`. If missing data have been estimated from multiple imputations, each imputation should be identified in an optional `Draw` column.
`mu_0`	A vector, corresponding to the prior mean. If NULL, all groups are initialised with the same empirical mean for each peptide.
`lambda_0`	A number, corresponding to the prior covariance scaling parameter.
`Sigma_0`	A matrix, corresponding to the prior covariance parameter. If NULL, the identity matrix will be used by default.
`nu_0`	A number, corresponding to the prior degrees of freedom.

Value

A tibble providing the parameters of the posterior t-distribution for the mean of the considered groups for each peptide.

Examples

TRUE

Index