Package ‘CyTOFpower’

May 10, 2024

Type Package

Title Power analysis for CyTOF experiments

Version 1.10.0

Description This package is a tool to predict the power of CyTOF experiments in the context of differential state analyses. The package provides a shiny app with two options to predict the power of an experiment: i. generation of in-sicilico CyTOF data, using users input ii. browsing in a grid of parameters for which the power was already precomputed.

biocViews FlowCytometry, SingleCell, CellBiology, StatisticalMethod, Software

License LGPL-3

Encoding UTF-8

Depends R (>= 4.1)

Imports CytoGLMM, diffcyt, DT, dplyr, ggplot2, magrittr, methods, rlang, stats, shiny, shinyFeedback, shinyjs, shinyMatrix, SummarizedExperiment, tibble, tidyr

RoxygenNote 7.1.2

VignetteBuilder knitr

Suggests testthat (>= 3.0.0), BiocStyle, knitr

Config/testthat/edition 3

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

git_branch RELEASE_3_19

git_last_commit 5ec8e4f

git_last_commit_date 2024-04-30

Repository Bioconductor 3.19

Date/Publication 2024-05-10

Author Anne-Maud Ferreira [cre, aut] (<https://orcid.org/0000-0002-4749-746X>),
Catherine Blish [aut],
Susan Holmes [aut]

Maintainer Anne-Maud Ferreira <anne-maud.ferreira@stanford.edu>

Contents

`compute_effectsize` *Compute effect size*

Description

Compute effect size

Usage

```
compute_effectsize(raw_data_lg)
```

Arguments

`raw_data_lg`	data.frame, cells values in long format.

Details

Compute observed Cohen's effect size and observed fold change.

Value

data.frame, Cohen's effect size and fold change that were observed in the data for each marker.

`compute_pwr` *Compute power*

Description

Compute power

Usage

```
compute_pwr(model_values, alpha = 0.05)
```

Arguments

`model_values`	data.frame, output of run_models().
`alpha`	numeric, significance level.

Details

Compute the power based on the model values.

Value

numeric, power.

`compute_variance` *Compute variance*

Description

Compute variance

Usage

```
compute_variance(raw_data_lg)
```

Arguments

`raw_data_lg`	data.frame, cells values in long format.

Details

Compute the observed variance in the data.

Value

numeric, observed variance mean.

`CyTOFpower` *CyTOFpower: R-package to predict the power of a CyTOF experiment*

Description

This package provides functions and a shiny app to predict the power of a CyTOF experiment.

Interactive shiny app to predict the power of a CyTOF experiment.

Usage

```
CyTOFpower()
```

Value

Interactive shiny app.

Examples

```
# Launch the shiny app
if (interactive()) {
  CyTOFpower()
}
```

`function_apply_modelcomputations_modelchoice` *Run DS tests for one simulation.*

Description

Run DS tests for one simulation.

Usage

```
function_apply_modelcomputations_modelchoice(
  list_combined_output,
  model = c("cytoglmm", "cytoglm", "testDS_limma_random", "testDS_limma_fixed",
    "testDS_lmm")
)
```

Arguments

`list_combined_output`	list, of simulated data (output of the function_apply_onesimulation_withmarkerinfo function).
`model`	vector, name(s) of models to test.

Details

Wrapper to run the models through the different simulations.

Value

data.frame of results for each simulation.

`function_apply_onesimulation_withmarkerinfo`

One simulation with markers NB information.

Description

One simulation with markers NB information.

Usage

```
function_apply_onesimulation_withmarkerinfo(variation)
```

Arguments

`variation`	list, list of list containing the different input parameter variations @describeIn function_create_mock_dataset_withmarkerinfo.

Details

Apply one simulation for a given parameter combination number times (iterations), when we have prior information about the markers distribution parameters (mean and dispersion of the negative binomial).

Value

list of simulated data. Each list member contains 5 slots: - variation: the variation of input paramaters which has been used in input; - df_info: experimental information (donor IDs, group IDs, samples IDs); - DE_markers_names: name of the differentially expressed markers; - ls_mock_data: list of data.frames, each data.frame being one simulation of the cell values using the input parameters provided (list is length nb_sim).

`function_check_nbDEmarkers`

Number of DE markers greater than 1.

Description

Number of DE markers greater than 1.

Usage

```
function_check_nbDEmarkers(nb_DEmarker)
```

Arguments

`nb_DEmarker`	numeric, number of differentially expressed markers.

Details

Function to check that the number of DE markers is greater than 1.

Value

error message

`function_check_nbmarkers`

Number of markers greater than 2.

Description

Number of markers greater than 2.

Usage

```
function_check_nbmarkers(nb_marker)
```

Arguments

`nb_marker`	numeric, total number of markers.

Details

Function to check that the number of markers is greater than 2.

Value

error message if the total number of markers is lower than 3.

`function_compute_diffcyt_features` *Compute cells counts and medians.*

Description

Compute cells counts and medians.

Usage

```
function_compute_diffcyt_features(mock_flowset)
```

Arguments

`mock_flowset`	data.frame, cell values for each marker.

Details

Function to calculate features for diffcyt package models.

Value

list with cell counts and medians for each markers.

`function_create_mock_dataset_withmarkerinfo` *Compute simulated cell values for one simulation with markers NB information.*

Description

Compute simulated cell values for one simulation with markers NB information.

Usage

```
function_create_mock_dataset_withmarkerinfo(variation)
```

Arguments

`variation`	list, list of data.frames containing the different variable input paramaters to generate the data: - marker_name: name of the marker (character); - nb_donor: number of donors; - rho: fold change; - subject_effect: standard deviation for the normal distribution from which the donor's means will be drawn; - mu0: mean of the negative binomial for the gamma distribution from which the means of the different donor will be drawn; - dispersion: dispersion of the negative binomial from which the DE marker's cell values will be drawn; - nb_cell_per_sample: number of cells per sample.

Details

Function to compute the simulated cell values using a combination of variable parameters, when we have prior information about the markers distribution parameters (mean and dispersion of the negative binomial).

Value

list with 4 slots: - df_info: data.frame of experimental information; - DEmarkers_names: vector of DE marker names; - raw_data: data.frame of raw cell values; - data: data.frame of transformed cell values.

`function_DEmarkers_sup_nbmarkers`

Check on the number of DE markers.

Description

Check on the number of DE markers.

Usage

```
function_DEmarkers_sup_nbmarkers(nb_markers, nb_DEmarker)
```

Arguments

`nb_markers`	numeric, total number of markers.
`nb_DEmarker`	numeric, number of differentially expressed markers.

Details

Function to check that number of DE markers greater than number of markers.

Value

error message if the number of DE markers is greater than the total number of markers.

`function_desigmat_contrast_diffcytDSlimma_fixedeffect`
Design and contrast matrices for diffcyt-DS-limma with fixed effect.

Description

Design and contrast matrices for diffcyt-DS-limma with fixed effect.

Usage

```
function_desigmat_contrast_diffcytDSlimma_fixedeffect(df_experiment_info)
```

Arguments

`df_experiment_info`	data.frame, information about the experiment (donor IDs, group IDs and sample IDs).

Details

Function to create the design matrix and contrast for the diffcyt-DS-limma model with fixed effect.

Value

list with 3 slots: - design_matrix: the design matrix; - contrast: the contrast matrix; - effect: the specification of the "fixed" effect.

`function_desigmat_contrast_diffcytDSlimma_randomeffect`
Design and contrast matrices for diffcyt-DS-limma with random effect.

Description

Design and contrast matrices for diffcyt-DS-limma with random effect.

Usage

```
function_desigmat_contrast_diffcytDSlimma_randomeffect(df_experiment_info)
```

Arguments

`df_experiment_info`	data.frame, information about the experiment (donor IDs, group IDs and sample IDs).

Details

Function to create the design matrix and contrast for the diffcyt-DS-limma model with random effect from the diffcyt package.

Value

list with 3 slots: - design_matrix: the design matrix; - contrast: the contrast matrix; - effect: the specification of the "random" effect.

`function_extract_marker_names`

Extract marker names.

Description

Extract marker names.

Usage

```
function_extract_marker_names(mock_dataset)
```

Arguments

`mock_dataset`	data.frame, containing the cell values for each marker.

Details

Function to extract the marker names.

Value

vector of marker names.

`function_formula_contrast_diffcytDSLMM_randomeffect`

Formula and contrast matrix for diffcyt-DS-LMM with random effect.

Description

Formula and contrast matrix for diffcyt-DS-LMM with random effect.

Usage

```
function_formula_contrast_diffcytDSLMM_randomeffect(df_experiment_info)
```

Arguments

`df_experiment_info`	data.frame, information about the experiment (donor IDs, group IDs and sample IDs).

Details

Function to create formula and contrast for diffcyt-DS-LMM with random effect.

Value

list with 3 slots: - formula: the formula for the model; - contrast: the contrast matrix.

`function_is_data_paired` *Data paired.*

Description

Data paired.

Usage

```
function_is_data_paired(data)
```

Arguments

`data`	data.frame, experimental information containing (sample IDS, donor IDs).

Details

Function to check if the data is paired.

Value

logical, TRUE is the data are paired - FALSE if the data are not paired

---

`function_names_DE_markers` *Generate name of the DE markers.*

Description

Generate name of the DE markers.

Usage

```
function_names_DE_markers(total_nb_marker, nb_DE_marker)
```

Arguments

`total_nb_marker`	numeric, total number of markers.
`nb_DE_marker`	numeric, number of DE markers.

Details

Function to generate the name of the DE markers.

Value

vactor of marker names.

`function_run_bootstrapcytoGLMM` *Run cytoglm.*

Description

Run cytoglm.

Usage

```
function_run_bootstrapcytoGLMM(mock_dataset, nb_bootstrap = 500)
```

Arguments

`mock_dataset`	data.frame, cell values for each marker.
`nb_bootstrap`	numeric, number of bootstrap (by defaults nb_bootstrap = 1000).

Details

Function to run the Generalized Linear Model with Bootstrap, from the CytoGLMM package.

Value

list of 3 slots: - model_fit: fit of the model; - plot: plot of the effects; - results_summary: data.frame containing the results of the models for each marker.

`function_run_cytoGLMM` *Run cytoglmm*

Description

Run cytoglmm

Usage

```
function_run_cytoGLMM(mock_dataset)
```

Arguments

`mock_dataset`	data.frame, cell values for each marker.

Details

Function to run GLMM model from CytoGLMM package.

Value

list of 3 slots: - model_fit: fit of the model; - plot: plot of the effects; - results_summary: data.frame containing the results of the models for each marker.

`function_run_diffcytDSlimma` *Run diffcyt-DS-limma model.*

Description

Run diffcyt-DS-limma model.

Usage

```
function_run_diffcytDSlimma(
  ls_desigmat_contrast,
  df_experiment_info,
  ls_features
)
```

Arguments

`ls_desigmat_contrast`	list, with design and contrast matrices.
`df_experiment_info`	data.frame, information about the experiment (donor IDs, group IDs and sample IDs).
`ls_features`	list, with cell counts and medians for each markers.

Details

Function to run diffcyt-DS-limma model from the diffcyt package.

Value

list with 2 slots: - model_fit: the model fit; - result_summary: results of the model for each marker.

`function_run_diffcytDSLMM`

Run diffcyt-DS-LMM model.

Description

Run diffcyt-DS-LMM model.

Usage

```
function_run_diffcytDSLMM(ls_form_contrast, df_experiment_info, ls_features)
```

Arguments

`ls_form_contrast`	list, with formula and constrat matrix.
`df_experiment_info`	data.frame, information about the experiment (donor IDs, group IDs and sample IDs).
`ls_features`	list, with cell counts and medians for each markers.

Details

Function to run diffcyt-DS-LMM model from the diffcyt package.

Value

list with 2 slots: - model_fit: the model fit; - result_summary: results of the model for each marker.

`function_run_diffcyt_full_pipeline`
Run diffcyt pipeline

Description

Run diffcyt pipeline

Usage

```
function_run_diffcyt_full_pipeline(
  onevariation,
  model = c("limma", "LMM"),
  effect = c("random", "fixed")
)
```

Arguments

`onevariation`	list, of simulated data (output of the function_wrapper_apply_simulation_nbtimes function).
`model`	character, model to run: "limma" or "LMM".
`effect`	character, effect: "random" or "fixed".

Details

Function to run diffcyt pipeline. We do not used the diffcyt function directly because of the limma model run differently with the effects.

Value

list with 2 slots: - model_fit: the model fit; - result_summary: results of the model for each marker.

`function_summary_results_models`
Summarize data and results.

Description

Summarize data and results.

Usage

```
function_summary_results_models(summary_from_model, package)
```

Arguments

`summary_from_model`	list, output from the functions running the models.
`package`	character, package used to run the test: "CytoGLMM" or "diffcyt".

Details

Function to do a summary of the tested data and model's results for the CytoGLMM and diffcyt packages.

Value

data.frame of results for each simulation.

`function_to_compute_model_computation_onesimulation_modelchoice`

Run DS tests for one simulation.

Description

Run DS tests for one simulation.

Usage

```
function_to_compute_model_computation_onesimulation_modelchoice(
  onevariation,
  model
)
```

Arguments

`onevariation`	list, of simulated data (output of the function_wrapper_apply_simulation_nbtimes function).
`model`	vector, name(s) of models to test.

Details

Function to run the DS tests through the different models.

Value

data.frame of results for each simulation all models combined.

`function_to_transform_data` *arcsinh transformation.*

Description

arcsinh transformation.

Usage

```
function_to_transform_data(data, cofactor = 5)
```

Arguments

`data`	data.frame, cell values to transform.
`cofactor`	numeric, co-factor used in the arcsinh (by default cofactor = 5).

Details

Function to transform the data with the recommended transformation for cyTOF data: arcsinh with cofactor equals to 5.

Value

data.frame of transformed data.

`function_value_onemarker` *Compute simulated cell values for one marker with markers NB informations*

Description

Compute simulated cell values for one marker with markers NB informations

Usage

```
function_value_onemarker(
  marker_name,
  mu0,
  dispersion,
  subject_effect,
  nb_donor,
  nb_cell_per_sample,
  rho = 1
)
```

Arguments

`marker_name`	character, name of the marker.
`mu0`	numeric, general donor mean from which the individual mu0i will be drawn.
`dispersion`	numeric, dispersion of the markers.
`subject_effect`	numeric, standard deviation for the normal distribution from which the donor's means will be drawn (by default subject_effect = 0.01).
`nb_donor`	numeric, number of donors.
`nb_cell_per_sample`	numeric, number of cells per sample (by default nb_cell_per_sample = 500).
`rho`	numeric, fold change.

Details

Function to generate value for one marker with a mean and dispersion specified for the negative binomiale.

Value

data.frame of cell values.

Index