Package ‘greatR’

October 13, 2022

Title  Gene Registration from Expression and Time-Courses in R
Version  0.2.0
Description  A tool for registering (aligning) gene expression profiles between two species (reference data and data to transform).
License  GPL (>= 3)
URL  https://ruthkr.github.io/greatR/,
     https://github.com/ruthkr/greatR/
BugReports  https://github.com/ruthkr/greatR/issues/
Depends  R (>= 3.5.0)
Imports  cli, data.table, dplyr, ggplot2, optimization, magrittr,
         rlang, scales, stats, stringr, purrr
Suggests  knitr, rmarkdown, testthat
VignetteBuilder  knitr
Encoding  UTF-8
RoxygenNote  7.1.2
NeedsCompilation  no
Author  Ruth Kristianingsih [aut, cre]
        (<https://orcid.org/0000-0003-1873-6203>),
        Alex Calderwood [aut] (<https://orcid.org/0000-0003-3749-035X>)
Maintainer  Ruth Kristianingsih <ruth.kristianingsih30@gmail.com>
Repository  CRAN
Date/Publication  2022-06-08 22:30:02 UTC

R topics documented:

  calculate_between_sample_distance .......................... 2
  get_approximate_stretch ................................. 3
  get_expression_of_interest .............................. 4
  get_mean_data ........................................... 5
calculate_between_sample_distance

Description
Calculate distance between sample data before and after registration

Usage
```
calculate_between_sample_distance(
    registration_results,
    gene_col = "locus_name",
    compare_ref_vs_transform = TRUE,
    accession_data_ref
)
```

Arguments
- `registration_results`: Result of registration process using `scale_and_register_data`.
- `gene_col`: Column name of gene accession, default is `locus_name`.
- `compare_ref_vs_transform`: If `TRUE`, the default, only comparison between reference data and data to transform is considered.
- `accession_data_ref`: Accession name of reference data.

Value
This function returns a list of data frames which includes:
- `distance_mean_df`: distance of mean expression values.
- `distance_scaled_mean_df`: distance of scaled mean expression (all genes).
- `distance_scaled_mean_df_only_nonreg`: distance of scaled mean expression (only non-registered genes).
- `distance_scaled_mean_df_only_reg`: distance of scaled mean expression (only registered genes).
**get_approximate_stretch**

- **distance_registered_df**
  distance of registered & scaled mean expression (all genes).
- **distance_registered_df_only_reg**
  distance of registered & scaled mean expression (only registered genes).

---

**get_approximate_stretch**

*Get approximate stretch factor*

**Description**

`get_approximate_stretch()` is a function to get a stretch factor estimation given input data. This function will take the time point ranges of both reference and query data and compare them to estimate the stretch factor.

**Usage**

```r
get_approximate_stretch(
  input_df,
  accession_data_to_transform,
  accession_data_ref
)
```

**Arguments**

- **input_df**
  Input data frame containing all replicates of gene expression in each genotype at each time point.
- **accession_data_to_transform**
  Accession name of data which will be transformed.
- **accession_data_ref**
  Accession name of reference data.

**Value**

This function returns an estimation of a stretch factor for registering the data.
get_expression_of_interest

Get expression of interest

Description

Get expression of interest

Usage

```r
get_expression_of_interest(
  data_ref,
  data_to_transform,
  id_table,
  lookup_col_ref_and_id_table = "CDS.model",
  lookup_col_ref_and_to_transform = "locus_name",
  colnames_wanted = NULL,
  tissue_wanted = NULL,
  gene_of_interest_acc,
  sum_exp_data_ref = FALSE,
  accession_data_to_transform = "Col0"
)
```

Arguments

- `data_ref`: Data frame of reference data.
- `data_to_transform`: Data frame of data to transform.
- `id_table`: Data frame of ID table connecting both reference data and data to transform.
- `lookup_col_ref_and_id_table`: Column names shared by both reference data and ID table.
- `lookup_col_ref_and_to_transform`: Column names shared by both reference data and data to transform.
- `colnames_wanted`: List of column names to keep from both reference data and data to transform.
- `tissue_wanted`: Name of tissue from which data will be compared.
- `gene_of_interest_acc`: Gene accession list from data to transform.
- `sum_exp_data_ref`: If TRUE then sum all gene data. Default is FALSE.
- `accession_data_to_transform`: Accession name of data which will be transformed.

Value

A data frame contains both reference data and data to transform for selected gene of interest.
**get_mean_data**

Calculate mean expression values from all expression data with replicates

**Description**

Calculate mean expression values from all expression data with replicates

**Usage**

```r
get_mean_data(
    exp,
    expression_value_threshold = 5,
    accession_data_to_transform,
    is_data_normalised = FALSE
)
```

**Arguments**

- **exp**: Input data frame containing all replicates of gene expression in each genotype at each time point.
- **expression_value_threshold**: Expression value threshold. Remove expressions if maximum is less than the threshold. If NULL keep all data.
- **accession_data_to_transform**: Accession name of data which will be transformed.
- **is_data_normalised**: TRUE if dataset has been normalised prior to registration process.

**Value**

A data frame contains only mean expression data.

---

**optimise_registration_params**

Optimise registration parameters with Simulated Annealing

**Description**

Optimise registration parameters with Simulated Annealing
Usage

```r
optimise_registration_params(
  input_df,
  genes = NULL,
  stretches_bound = NA,
  shifts_bound = NA,
  initial_rescale = FALSE,
  do_rescale = TRUE,
  min_num_overlapping_points = 4,
  maintain_min_num_overlapping_points = FALSE,
  accession_data_to_transform,
  accession_data_ref,
  start_timepoint = c("reference", "transform", "zero"),
  expression_value_threshold = 5,
  is_data_normalised = FALSE,
  num_iterations = 60
)
```

Arguments

- **input_df**: Input data frame containing all replicates of gene expression in each genotype at each time point.
- **genes**: List of genes to optimise.
- **stretches_bound**: Optional candidate registration stretch factors define search space, otherwise automatic.
- **shifts_bound**: Optional candidate registration shift values to define search space, otherwise automatic.
- **initial_rescale**: Scaling gene expression prior to registration if TRUE.
- **do_rescale**: Scaling gene expression using only overlapping time points during registration.
- **min_num_overlapping_points**: Number of minimum overlapping time points. Shifts will be only considered if it leaves at least these many overlapping points after applying the registration function.
- **maintain_min_num_overlapping_points**: Whether to automatically calculate extreme (minimum and maximum) values of shifts to maintain specified min_num_overlapping_points condition. By default, FALSE.
- **accession_data_to_transform**: Accession name of data which will be transformed.
- **accession_data_ref**: Accession name of reference data.
- **start_timepoint**: Time points to be added in both reference data and data to transform after shifting and stretching. Can be either "reference" (the default), "transform", or "zero".
plot_heatmap

expression_value_threshold
Expression value threshold. Remove expressions if maximum is less than the
threshold. If NULL keep all data.

is_data_normalised
TRUE if dataset has been normalised prior to registration process.

num_iterations Maximum number of iterations of the algorithm. Default is 100.

Value
List of optimum registration parameters, optimum_params_df, and other candidate registration pa-
rameters, candidate_params_df for all genes.

Value
Distance heatmap of gene expression profiles over time between two different species.

Description
Function plot_heatmap() allows users to plot distances between samples from different time
points to investigate the similarity of progression of gene expression states between species before
or after registration.

Usage
plot_heatmap(
    sample_dist_df,
    title = NULL,
    axis_fontsize = NULL,
    same_min_timepoint = FALSE,
    same_max_timepoint = FALSE
)

Arguments
sample_dist_df Input data frame containing sample distance between two different species.
title Optional plot title.
axis_fontsize Font size of X and Y axes labels.
same_min_timepoint If FALSE, the default, will not take data with the same minimum time point.
same_max_timepoint If FALSE, the default, will not take data with the same maximum time point.

Value
Distance heatmap of gene expression profiles over time between two different species.
**plot_registration_results**

*Plot gene of interest after registration*

---

**Description**

Plot gene of interest after registration

**Usage**

```r
plot_registration_results(
    reg_result_df,
    model_comparison_df = NULL,
    gene_accession = "first_genes",
    title = NULL,
    ncol = NULL,
    sync_timepoints = FALSE
)
```

**Arguments**

- `reg_result_df`: Data frame of registration results, output from registration process.
- `model_comparison_df`: Data frame of model comparison, also output from registration process.
- `gene_accession`: List of gene accessions, default is `first_genes` which will take first 25 genes.
- `title`: Optional plot title.
- `ncol`: Number of columns in the plot grid. By default this is calculated automatically.
- `sync_timepoints`: Whether to synchronise maximum time points for each accession, by default FALSE.

**Value**

Plot of gene of interest after registration process.

---

**scale_and_register_data**

*Register or synchronize different expression profiles*

---

**Description**

`scale_and_register_data()` is a function to register expression profiles a user wish to compare. This includes an option to scale data before registration, find and calculate score of optimal shifts and stretches, as well as apply the best shifts and stretches.
**scale_and_register_data**

Usage

```r
scale_and_register_data(
  input_df,
  stretches = NA,
  shifts = NA,
  min_num_overlapping_points,
  maintain_min_num_overlapping_points = FALSE,
  initial_rescale = FALSE,
  do_rescale = TRUE,
  accession_data_to_transform,
  accession_data_ref,
  start_timepoint = c("reference", "transform", "zero"),
  expression_value_threshold = 5,
  is_data_normalised = FALSE,
  optimise_registration_parameters = FALSE,
  num_iterations = 60
)
```

Arguments

- **input_df**: Input data frame containing all replicates of gene expression in each genotype at each time point.
- **stretches**: Candidate registration stretch factors to apply to data to transform.
- **shifts**: Candidate registration shift values to apply to data to transform.
- **min_num_overlapping_points**: Number of minimum overlapping time points. Shifts will be only considered if it leaves at least these many overlapping points after applying the registration function.
- **maintain_min_num_overlapping_points**: Whether to automatically calculate extreme (minimum and maximum) values of shifts to maintain specified `min_num_overlapping_points` condition. By default, FALSE.
- **initial_rescale**: Scaling gene expression prior to registration if TRUE.
- **do_rescale**: Scaling gene expression using only overlapping time points points during registration.
- **accession_data_to_transform**: Accession name of data which will be transformed.
- **accession_data_ref**: Accession name of reference data.
- **start_timepoint**: Time points to be added in both reference data and data to transform after shifting and stretching. Can be either "reference" (the default), "transform", or "zero".
- **expression_value_threshold**: Expression value threshold. Remove expressions if maximum is less than the threshold. If NULL keep all data.
scale_and_register_data

is_data_normalised
TRUE if dataset has been normalised prior to registration process.

optimise_registration_parameters
Whether to optimise registration parameters with Simulated Annealing. By default, FALSE.

num_iterations
Maximum number of iterations in the Simulated Annealing optimisation. By default, 60.

Value
This function returns a list of data frames, containing:

mean_df
a data frame containing mean expression value of each gene and accession for every time point.

mean_df_sc
identical to mean_df, with additional column sc.expression_value which the scaled mean expression values.

to_shift_df
a processed input data frame which is ready to be registered.

best_shifts
a data frame containing best shift factor for each given stretch.

shifted_mean_df
the registration result - after stretching and shifting.

imputed_mean_df
the imputed (transformed to be the same in a set of common time points) registration result.

all_shifts_df
a table containing candidates of registration parameters and a score after applying each parameter (stretch and shift factor).

model_comparison_df
a table comparing the optimal registration function for each gene (based on all_shifts_df scores) to model with no registration applied.

Examples

## Not run:
# Load a data frame from the sample data
all_data_df <- system.file("extdata/brapa_arabidopsis_all_replicates.csv", package = "greatR") %>%
  utils::read.csv()

# Running the registration
registration_results <- scale_and_register_data(
  input_df = all_data_df,
  stretches = c(3, 2.5, 2, 1.5, 1),
  shifts = seq(-4, 4, length.out = 33),
  min_num_overlapping_points = 4,
  initial_rescale = FALSE,
  do_rescale = TRUE,
  accession_data_to_transform = "Col0",
  accession_data_ref = "Ro18",
  start_timepoint = "reference"
)

## End(Not run)
**summary_model_comparison**

*Summarise registration results*

---

**Description**
Summarise registration results

**Usage**

```r
summary_model_comparison(model_comparison)
```

**Arguments**

- `model_comparison`
  
  Input data frame, element `model_comparison` of result list of `scale_and_register_data()`.

**Value**
List containing summary table, registered gene accessions, and non-registered gene accessions.
Index

calculate_between_sample_distance, 2
get_approximate_stretch, 3
get_expression_of_interest, 4
get_mean_data, 5
optimise_registration_params, 5
plot_heatmap, 7
plot_registration_results, 8
scale_and_register_data, 2, 8
summary_model_comparison, 11