

Package ‘greatR’

January 9, 2024

Title Gene Registration from Expression and Time-Courses in R

Version 1.1.0

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, furrr, future, ggplot2, neldermead, optimization, scales, stats

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.2.3

Config/testthat.edition 3

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 2024-01-09 01:50:07 UTC

R topics documented:

calculate_distance	2
get_approximate_stretch	2
plot_heatmap	3

plot_registration_results	4
register	4
summarise_registration	6

calculate_distance *Calculate distance between sample data before and after registration*

Description

Calculate distance between sample data before and after registration

Usage

```
calculate_distance(results)
```

Arguments

results Result of registration process using [register](#).

Value

This function returns a list of data frames which includes:

registered	distance between scaled reference and query expressions using registered time points.
original	distance between scaled reference and query expressions using original time points.

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

```
get_approximate_stretch(data, reference = "ref", query = "query")
```

Arguments

data	Input data frame, either containing all replicates of gene expression or not.
reference	Accession name of reference data.
query	Accession name of query data.

Value

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

plot_heatmap	<i>Visualise distances between samples from different time points</i>
--------------	-----------------------------------------------------------------------

Description

plot_heatmap() is a function that 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(  
  results,  
  type = c("registered", "original"),  
  match_timepoints = FALSE,  
  title = NULL,  
  axis_fontsize = NULL  
)
```

Arguments

results	Results containing distances between two different reference and query data, output of calculate_distance .
type	Type of plot, determines whether to use "registered" or "original" time points. By default, "registered".
match_timepoints	If TRUE, will match query time points to reference time points.
title	Optional plot title.
axis_fontsize	Font size of X and Y axes labels.

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

```
plot_registration_results(
  results,
  type = c("registered", "original"),
  genes_list = NULL,
  title = NULL,
  ncol = NULL,
  plot_mean_data = FALSE
)
```

Arguments

<code>results</code>	Registration results, output of the <code>register</code> registration process.
<code>type</code>	Type of plot, determines whether to use "registered" or "original" time points. By default, "registered".
<code>genes_list</code>	Optional vector indicating the gene_id values to be plotted.
<code>title</code>	Optional plot title.
<code>ncol</code>	Number of columns in the plot grid. By default this is calculated automatically.
<code>plot_mean_data</code>	Whether mean data is displayed or not.

Value

Plot of genes of interest after registration process (`type = "registered"`) or showing original time points (`type = "original"`).

register*Register or synchronize different expression profiles*

Description

`register()` is a function to register expression profiles a user wishes to compare.

Usage

```
register(  
  input,  
  stretches = NA,  
  shifts = NA,  
  reference,  
  query,  
  scaling_method = c("none", "z-score", "min-max"),  
  overlapping_percent = 50,  
  optimise_registration_parameters = TRUE,  
  optimisation_method = c("nm", "lbfgsb", "sa"),  
  optimisation_config = NULL,  
  exp_sd = NA,  
  num_cores = NA  
)
```

Arguments

<code>input</code>	Input data frame containing all replicates of gene expression in each genotype at each time point.
<code>stretches</code>	Candidate registration stretch factors to apply to query data, only required if <code>optimise_registration_parameters</code> = FALSE.
<code>shifts</code>	Candidate registration shift values to apply to query data, only required if <code>optimise_registration_parameters</code> = FALSE.
<code>reference</code>	Accession name of reference data.
<code>query</code>	Accession name of query data.
<code>scaling_method</code>	Scaling method applied to data prior to registration process. Either none (default), z-score, or min-max.
<code>overlapping_percent</code>	Minimum percentage of overlapping time points on the reference data. Shifts will be only considered if it leaves at least this percentage of overlapping time points after applying the registration function.
<code>optimise_registration_parameters</code>	Whether to optimise registration parameters. By default, TRUE.
<code>optimisation_method</code>	Optimisation method to use. Either "nm" for Nelder-Mead (default), "lbfgsb" for L-BFGS-B, or "sa" for Simulated Annealing.
<code>optimisation_config</code>	Optional list with arguments to override the default optimisation configuration.
<code>exp_sd</code>	Optional experimental standard deviation on the expression replicates.
<code>num_cores</code>	Number of cores to use if the user wants to register genes asynchronously (in parallel) in the background on the same machine. By default, NA, the registration will be run without parallelisation.

Value

This function returns a list of data frames, containing:

- data** a table containing the scaled input data and an additional `timepoint_reg` column after applying registration parameters to the query data.
- model_comparison** 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
data_path <- system.file("extdata/brapa_arabidopsis_all_replicates.csv", package = "greatR")
all_data <- utils::read.csv(data_path)

# Running the registration
registration_results <- register(
  input = all_data,
  reference = "Ro18",
  query = "Col0"
)

## End(Not run)
```

summarise_registration

Summarise registration results

Description

Summarise registration results

Usage

```
summarise_registration(results)
```

Arguments

- results** Registration results, output of the `register` registration process.

Value

This function returns a list containing:

- summary** contains result summaries of the registration results.
- registered_genes** vector of gene accessions which were successfully registered.
- non_registered_genes** vector of non-registered gene accessions.

Index

calculate_distance, 2, 3
get_approximate_stretch, 2
plot_heatmap, 3
plot_registration_results, 4
register, 2, 4, 4, 6
summarise_registration, 6