Package 'iraceplot'

February 11, 2025

Title Plots for Visualizing the Data Produced by the 'irace' Package

Version 2.1.0

Description Graphical visualization tools for analyzing the data produced by 'irace'. The 'iraceplot' package enables users to analyze the performance and the parameter space data sampled by the configuration during the search process. It provides a set of functions that generate different plots to visualize the configurations sampled during the execution of 'irace' and their performance. The functions just require the log file generated by 'irace' and, in some cases, they can be used with user-provided data.

License MIT + file LICENSE

Depends R (>= 4.0.0)

Imports irace (>= 4.0.0), cli, data.table, dplyr, DT, forcats, fs, ggforce, ggplot2 (>= 3.3.6), gridExtra, knitr, labeling, matrixStats (>= 0.55), plotly, rlang, rmarkdown (>= 2.7), stats, tibble, tidyr, truncnorm, utils, viridisLite, withr

Suggests testthat (>= 3.0.0)

URL https://auto-optimization.github.io/iraceplot/, https://github.com/auto-optimization/iraceplot/

BugReports https://github.com/auto-optimization/iraceplot/issues

Config/testthat/edition 3

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author Manuel López-Ibáñez [aut, cre] (<https://orcid.org/0000-0001-9974-1295>), Pablo Oñate Marín [aut], Leslie Pérez Cáceres [aut] (<https://orcid.org/0000-0001-5553-6150>)

Maintainer Manuel López-Ibáñez <manuel.lopez-ibanez@manchester.ac.uk>

Repository CRAN

Date/Publication 2025-02-11 15:30:01 UTC

Contents

`iraceplot-package` *The iraceplot package: Plots for Visualizing the Data Produced by the 'irace' Package*

Description

Graphical visualization tools for analyzing the data produced by 'irace'. The 'iraceplot' package enables users to analyze the performance and the parameter space data sampled by the configuration during the search process. It provides a set of functions that generate different plots to visualize the configurations sampled during the execution of 'irace' and their performance. The functions just require the log file generated by 'irace' and, in some cases, they can be used with user-provided data.

Details

boxplot_performance; boxplot_test; boxplot_training; parallel_cat; plot_configurations; parallel_coord; plot_experiments_matrix; plot_model; report; sampling_distance; sampling_frequency; sampling_frequency_iteration; sampling_heatmap2; sampling_heatmap; sampling_pie; scatter_performance; scatter_test; scatter_training;

If you need information about any function you can write: `?name_function`

If you need more information, go to the following page: https://auto-optimization.github.io/iraceplot/

Author(s)

Maintainer: Manuel López-Ibáñez <manuel.lopez-ibanez@manchester.ac.uk> (ORCID)

Authors:

- Pablo Oñate Marín <pablo.onate.m@gmail.com>
- Leslie Pérez Cáceres <leslie.perez@pucv.cl> (ORCID)

See Also

Useful links:

- https://auto-optimization.github.io/iraceplot/
- https://github.com/auto-optimization/iraceplot/
- Report bugs at https://github.com/auto-optimization/iraceplot/issues

---

ablation_plot *Create plot from an ablation log*

Description

Create plot from an ablation log

Usage

```
ablation_plot(
  ablog,
  type = c("mean", "boxplot", "rank"),
  n = 0L,
  ylab = "Mean configuration cost",
  ylim = NULL,
  rotate_labs = TRUE,
  rename_labs = NULL,
  filename = NULL
)
```

Arguments

ablog	list()|character(1) Ablation log object returned by irace::ablation(). Alternatively, the path to an .Rdata file, e.g., "log-ablation.Rdata", from which the object will be loaded.
type	Type of plot. Supported values are "mean" and "boxplot". Adding "rank" will plot rank per instance instead of raw cost value.
n	integer(1) Number of steps included in the plot. By default all steps from source to target are included.
ylab	Label of y-axis.
ylim	Numeric vector of length 2 giving the y-axis range.
rotate_labs	logical(1) Whether to rotate labels in x-axis. They are rotated by default because they are typically large.
rename_labs	character() Renaming table for nicer labels. For example, c("No value"="NA", "LongParameterName"="LPN").
filename	(character(1)) File name to save the plot, for example "~/path/example/filename.png".

Value

ggplot2::ggplot() boxplot object

Author(s)

Manuel López-Ibáñez

See Also

irace::ablation(), irace::plotAblation()

Examples

```
ablog <- read_ablogfile(system.file(package="irace", "exdata", "log-ablation.Rdata"))
ablation_plot(ablog)
ablation_plot(ablog, type="boxplot", rotate_labs = FALSE)
ablation_plot(ablog, type = "rank,boxplot", rename_labs =
              c("localsearch"="ls", algorithm="algo", source="default"))
ablation_plot(ablog, type="rank,mean,boxplot", n = 4, rotate_labs = FALSE)
ablog <- system.file(package="iraceplot", "exdata", "log-ablation-autoMOPSODTLZ.Rdata")
ablation_plot(ablog, type="rank,mean,boxplot")
```

boxplot_performance *Box Plot of the performance of a set of configurations*

Description

Creates a box plot that displays the performance of a set of configurations which can be displayed by iteration.

Usage

```
boxplot_performance(
  experiments,
  allElites = NULL,
  type = c("all", "ibest"),
  first_is_best = TRUE,
  rpd = TRUE,
  show_points = TRUE,
  best_color = "#08bfaa",
  xlab = "Configurations",
  boxplot = FALSE,
  filename = NULL,
  interactive = base::interactive()
)
```

Arguments

experiments	matrix() Experiment matrix obtained from irace training or testing data. Configurations in columns and instances in rows. As in irace, column names (configurations IDs) should be characters.
allElites	List or vector of configuration ids, (default NULL). These configurations should be included in the plot. If the argument is not provided all configurations in experiments are included. If allElites is a vector all configurations are assumed without iteration unless argument `type="ibest"` is provided, in which case each configuration is assumed to be from a different iteration. If `allElites` is a list, each element of the list is assumed as an iteration.
type	String, (default "all") possible values are "all" or "ibest". "all" shows all the selected configurations showing iterations if the information is provided. "ibest" shows the elite configurations of each iteration, note that the best configuration is always assumed to be first in the vector of each iteration.
first_is_best	Boolean (default TRUE) Enables the display in a different color the best configuration identified as the first one in a vector. If FALSE, all configurations are shown in the same color.
rpd	(`logical(1)`) TRUE to plot performance as the relative percentage deviation to best results per instance, FALSE to plot raw performance.

show_points	Logical, (default TRUE) TRUE to plot performance points together with the box plot.
best_color	String, (default "#08bfaa") color to display best configurations.
xlab	String, (default "Configurations") label for the x axis.
boxplot	By default, display a violin plot (ggplot2::geom_violin()). If TRUE, show a classical boxplot.
filename	(character(1)) File name to save the plot, for example "~/path/example/filename.png".
interactive	(logical(1)) TRUE if the report may use interactive features (using plotly::ggplotly(), plotly::plot_ly() and DT::renderDataTable()) or FALSE if such features must be disabled. Defaults to the value returned by interactive(),

Details

The performance data is obtained from the experiment matrix provided in the experiments argument. The configurations can be selected using the allElites argument and this argument can be also used to define the iteration of each elite configuration was evaluated.

Value

ggplot2::ggplot() boxplot object

See Also

boxplot_test() boxplot_training()

Examples

```
iraceResults <- read_logfile(system.file(package="irace", "exdata",
                                         "irace-acotsp.Rdata", mustWork = TRUE))
boxplot_performance(iraceResults$experiments, iraceResults$allElites)

boxplot_performance(iraceResults$testing$experiments, iraceResults$iterationElites)
```

---

boxplot_test *Box Plot Testing Performance*

Description

Creates a box plot that displays the performance of a set of configurations on the test instances.

Usage

```
boxplot_test(irace_results, type = c("all", "ibest", "best"), ...)
```

Arguments

`irace_results`	The data generated when loading the `.Rdata` file created by `irace` (or the filename of that file).
`type`	String, (default `"all"`) possible values are `"all"`, "ibest" or "best". "all" shows all the configurations included in the test, "best" shows the elite configurations of the last iteration and "ibest" shows the elite configurations of each iteration (requires that irace includes the iteration elites in the testing).
`...`	Other arguments passed to `boxplot_performance()`.

Details

The performance data is obtained from the test evaluations performed by irace. Note that the testing is not a default feature in irace and should be enabled in the setup (see the irace package user guide for more details).

Value

`ggplot2::ggplot()` boxplot object

See Also

`boxplot_training()` `boxplot_performance()`

Examples

```
iraceResults <- read_logfile(system.file(package="iraceplot", "exdata",
                                         "guide-example.Rdata", mustWork = TRUE))
boxplot_test(iraceResults)
```

`boxplot_training` *Box Plot Training*

Description

Creates a box plot that displays the performance of a set of configurations on the training instances. Performance data is obtained from the evaluations performed by irace during the execution process. This implies that the number of evaluations can differ between configurations.

Usage

```
boxplot_training(
  irace_results,
  iteration = NULL,
  id_configurations = NULL,
  ...
)
```

Arguments

irace_results	The data generated when loading the .Rdata file created by irace (or the filename of that file).
iteration	Numeric, iteration number that should be included in the plot (example: iteration = 5) When no iteration and no id_condigurations are provided, the iterations is assumed to be the last one performed by irace. The performance data is obtained from the evaluations performed by irace during the execution process. This implies that the number of evaluations can differ between configurations due to the elimination process applied by irace. This plot, consequently, does not provide a complete compaarison of two configurations, for a fair comparison use the test data plot.
id_configurations	Numeric vector, configurations ids whose performance should be included in the plot. If no ids are provided, the configurations ids are set as the elite configuration ids of the selected iteration (last iteration by default) (example: id_configurations = c(20,50,100,300,500,600,700)).
...	Other arguments passed to boxplot_performance().

Value

ggplot2::ggplot() boxplot object

See Also

boxplot_test() boxplot_performance()

Examples

```
iraceResults <- read_logfile(system.file(package="irace", "exdata",
                                         "irace-acotsp.Rdata", mustWork = TRUE))
boxplot_training(iraceResults)

boxplot_training(iraceResults, iteration = 5)
boxplot_training(iraceResults, id_configurations = c(23,28,29))
```

---

configurations_display

The configurations by iteration and instance

Description

This is a simplified version of the visualization you can obtain with acviz. This function is currently VERY SLOW.

Usage

```
configurations_display(
  irace_results,
  rpd = TRUE,
  filename = NULL,
  interactive = base::interactive()
)
```

Arguments

irace_results	The data generated when loading the .Rdata file created by irace (or the filename of that file).
rpd	(logical(1)) TRUE to plot performance as the relative percentage deviation to best results per instance, FALSE to plot raw performance.
filename	(character(1)) File name to save the plot, for example "~/path/example/filename.png".
interactive	(logical(1)) TRUE if the report may use interactive features (using plotly::ggplotly(), plotly::plot_ly() and DT::renderDataTable()) or FALSE if such features must be disabled. Defaults to the value returned by interactive(),

Value

ggplot2::ggplot() object

Examples

```
iraceResults <- read_logfile(system.file(package="iraceplot", "exdata",
                                         "guide-example.Rdata", mustWork = TRUE))
configurations_display(iraceResults)
```

---

distance_config *Distance between configurations*

Description

Calculate the difference between a configuration and the others in the irace data.

Usage

```
distance_config(irace_results, id_configuration, t = 0.05)
```

Arguments

irace_results	The data generated when loading the .Rdata file created by irace (or the filename of that file).
id_configuration	Numeric, configuration id which should be compared to others (example: id_configuration = c(806,809))
t	Numeric, (default 0.05) threshold that defines the distance (percentage of the domain size) to consider a parameter value equal to other.

Value

numeric

Examples

```
NULL
```

---

parallel_cat *Parallel Coordinates Category*

Description

Parallel categories plot of selected configurations. Numerical parameters are discretized to maximum n_bins intervals. To visualize configurations of other iterations these must be provided setting the argument iterations, groups of configurations of different iterations are shown in different colors. Specific configurations can be selected providing their ids in the id_configurations argument.

Usage

```
parallel_cat(
  irace_results,
  id_configurations = NULL,
  param_names = NULL,
  iterations = NULL,
  by_n_param = NULL,
  n_bins = 3,
  filename = NULL
)
```

Arguments

irace_results	The data generated when loading the .Rdata file created by irace (or the filename of that file).
id_configurations	Configuration ids to be included in the plot. Example: c(20,50,100,300,500,600,700)
param_names	(character()) Parameters to be included in the plot. Example: c("algorithm","alpha","rho","q0",

iterations	Numeric vector, iterations from which configuration should be obtained (example: iterations = c(1,4,5))
by_n_param	Numeric (optional), maximum number of parameters to be displayed.
n_bins	Numeric (default 3), number of intervals to generate for numerical parameters.
filename	(character(1)) File name to save the plot, for example "~/path/example/filename.png".

Details

The parameters to be included in the plot can be selected with the param_names argument. Additionally, the maximum number of parameters to be displayed in one plot. A list of plots is returned by this function in several plots are required to display the selected data.

Value

parallel categories plot

See Also

parallel_coord() plot_configurations()

Examples

```
iraceResults <- read_logfile(system.file(package="irace", "exdata",
                                         "irace-acotsp.Rdata", mustWork = TRUE))
parallel_cat(iraceResults)

parallel_cat(iraceResults, by_n_param = 6)
parallel_cat(iraceResults, id_configurations = c(20, 50, 100))
parallel_cat(iraceResults, param_names = c("algorithm", "alpha", "rho", "q0", "rasrank"))
parallel_cat(iraceResults, iterations = c(1, 4, 6), n_bins=4)
```

---

parallel_coord *Parallel Coordinates Plot*

Description

Parallel coordinates plot of a set of selected configurations. Each line in the plot represents a configuration. By default, the final elite configurations are shown. To visualize configurations of other iterations these must be provided setting the argument iterations, configurations of different iterations are shown in different colors. Setting the only_elites argument to FALSE displays all configurations in the selected iterations, specific configurations can be selected providing their ids in the id_configuration argument.

Usage

```
parallel_coord(
  irace_results,
  id_configurations = NULL,
  param_names = NULL,
  iterations = NULL,
  only_elite = TRUE,
  by_n_param = NULL,
  color_by_instances = TRUE,
  filename = NULL
)
```

Arguments

`irace_results`	The data generated when loading the `.Rdata` file created by `irace` (or the filename of that file).
`id_configurations`	Configuration ids to be included in the plot. Example: `c(20,50,100,300,500,600,700)`
`param_names`	(`character()`) Parameters to be included in the plot. Example: `c("algorithm","alpha","rho","q0",`
`iterations`	Numeric vector, iteration number that should be included in the plot (example: `iterations = c(1,4,5)`)
`only_elite`	logical (default `TRUE`), only print elite configurations (argument ignored when `id_configurations` is provided)
`by_n_param`	Numeric (optional), maximum number of parameters to be displayed.
`color_by_instances`	Logical (default TRUE), choose how to color the lines. TRUE shows the number of instances evaluated by the configuration in the colores. FALSE to show the iteration number where the configuration was sampled.
`filename`	(`character(1)`) File name to save the plot, for example `"~/path/example/filename.png"`.

Details

The parameters to be included in the plot can be selected with the `param_names` argument. Additionally, the maximum number of parameters to be displayed in one plot. A list of plots is returned by this function if several plots are required to display the selected data.

To export the plot to a file, it is possible to do it so manually using the functionality provided by `plotly` (https://plotly-r.com/exporting-static-images). If a filename is provided, orca server will be used to export the plots and thus, it requires the library to be installed (https://github.com/plotly/orca).

Value

parallel coordinates plot

Examples

```
iraceResults <- read_logfile(system.file(package="irace", "exdata",
                                         "irace-acotsp.Rdata", mustWork = TRUE))
parallel_coord(iraceResults)

parallel_coord(iraceResults, by_n_param = 5)
parallel_coord(iraceResults, only_elite = FALSE)
parallel_coord(iraceResults, id_configurations = c(20, 30, 40, 50, 100))
parallel_coord(iraceResults, param_names = c("algorithm", "alpha", "rho", "q0", "rasrank"))
parallel_coord(iraceResults, iterations = c(1, 4, 6))
```

`parameters_summarise` *Summarise parameters space*

Description

Summarise parameters space

Usage

```
parameters_summarise(parameters)
```

Arguments

`parameters`	(`list()`) Parameter space in irace format. See the function `irace::readParameters()`.

Value

`tibble::tibble()`

Author(s)

Manuel López-Ibáñez

Examples

```
# Read the parameters directly from text.
parameters_tab <- '
a "" i (2, 10)
b "" c (yes, no) | a < 5
c "" o (low, medium, high) | (a == 2) | (b == "yes")
d "" r (a, 50)'
parameters <- irace::readParameters(text=parameters_tab)
parameters_summarise(parameters)
```

parameters_tree *Print parameter dependencies as a tree*

Description

Print parameter dependencies as a tree

Usage

```
parameters_tree(parameters)
```

Arguments

parameters	(list()) Parameter space in irace format. See the function irace::readParameters().

Author(s)

Manuel López-Ibáñez

Examples

```
# Read the parameters directly from text.
parameters_tab <- '
a "" i (2, 10)
b "" c (yes, no) | a < 5
c "" o (low, medium, high) | (a == 2) | (b == "yes")
d "" r (a, 50)
'
parameters <- irace::readParameters(text=parameters_tab)
parameters_tree(parameters)
```

plot_configurations *Plot parameter configurations using parallel coordinates*

Description

Parallel coordinates plot of a set of provided configurations. Each line in the plot represents a configuration. The parameters to be included in the plot can be selected with the param_names argument. Additionally, the maximum number of parameters to be displayed in one plot. A list of plots is returned by this function in several plots are required to display the selected data.

Usage

```
plot_configurations(
  configurations,
  parameters,
  param_names = parameters$names,
  by_n_param = NULL,
  filename = NULL
)
```

Arguments

configurations	Data frame, configurations in irace format (example: configurations = iraceResults$allConfigura
parameters	(list()) Parameter space in irace format. See the function irace::readParameters().
param_names	(character()) Parameters to be included in the plot. Example: c("algorithm","alpha","rho","q0",
by_n_param	Numeric (optional), maximum number of parameters to be displayed
filename	(character(1)) File name to save the plot, for example "~/path/example/filename.png".

Details

To export the plot to a file, it is possible to do it so manually using the functionality provided by plotly (https://plotly-r.com/exporting-static-images). If a filename is provided, orca server will be used to export the plots and thus, it requires the library to be installed (https://github.com/plotly/orca).

Value

parallel coordinates plot

Examples

```
iraceResults <- read_logfile(system.file(package="irace", "exdata",
                                         "irace-acotsp.Rdata", mustWork = TRUE))
parameters <- iraceResults$scenario$parameters
plot_configurations(iraceResults$allConfigurations[iraceResults$iterationElites,],
                parameters = parameters)
plot_configurations(iraceResults$allConfigurations[iraceResults$iterationElites,],
                parameters = parameters,
                param_names = c("algorithm", "alpha", "rho", "q0", "rasrank"))
plot_configurations(iraceResults$allConfigurations[iraceResults$iterationElites,],
                parameters = parameters, by_n_param = 5)
```

plot_experiments_matrix *Heat Map Plot*

Description

Creates a heatmap plot that shows all performance data seen by irace. Configurations are shown in the x-axis in the order in which they are created in the configuration process. Instances are shown in the y-axis in the order in which they where seen during the configuration run. This plot gives a general idea of the configuration process progression, the number of evaluations of each configuration show how long they survived in the iterated racing procedure. Rejected configurations are shown with a red X.

Usage

```
plot_experiments_matrix(
  experiments,
  filename = NULL,
  metric = c("raw", "rpd", "rank"),
  show_conf_ids = FALSE,
  interactive = base::interactive()
)
```

Arguments

experiments	matrix() Experiment matrix obtained from irace training or testing data. Configurations in columns and instances in rows. As in irace, column names (configurations IDs) should be characters.
filename	(character(1)) File name to save the plot, for example "~/path/example/filename.png".
metric	Cost metric shown in the plot: "raw" shows the raw values, "rpd" shows relative percentage deviation per instance and "rank" shows rank per instance.
show_conf_ids	(logical(1)) If TRUE, it shows the configuration IDs in the x-axis. The default NA, only shows them if there are no more than 25.
interactive	(logical(1)) TRUE if the report may use interactive features (using plotly::ggplotly(), plotly::plot_ly() and DT::renderDataTable()) or FALSE if such features must be disabled. Defaults to the value returned by interactive(),

Value

ggplot2::ggplot() object

Note

Alternatively, experiments could be the data generated when loading the .Rdata file created by irace (or the filename of that file), from which the experiments matrix will be loaded.

Examples

```
iraceResults <- read_logfile(system.file(package="irace", "exdata",
                                         "irace-acotsp.Rdata", mustWork = TRUE))
plot_experiments_matrix(iraceResults)

plot_experiments_matrix(read_logfile(system.file(package="iraceplot", "exdata",
                                         "dummy-reject.Rdata", mustWork = TRUE)))
```

---

plot_model *Plot the sampling models used by irace*

Description

Display the sampling models from which irace generated parameter values for new configurations during the configurations process.

For categorical parameters a stacked bar plot is created. This plot shows the sampling probabilities of the parameter values for the elite configurations in the iterations of the configuration process.

For numerical parameters a sampling distributions plot of the numerical parameters for the elite configurations of an iteration. This plot shows de density function of the truncated normal distributions associated to each parameter for each elite configuration on each iteration.

Usage

```
plot_model(irace_results, param_name, filename = NULL)
```

Arguments

`irace_results`	The data generated when loading the `.Rdata` file created by `irace` (or the filename of that file).
`param_name`	String, parameter to be included in the plot, e.g., `param_name = "algorithm"`
`filename`	(`character(1)`) File name to save the plot, for example `"~/path/example/filename.png"`.

Value

sampling model plot

Examples

```
iraceResults <- read_logfile(system.file(package="irace", "exdata",
                                         "irace-acotsp.Rdata", mustWork = TRUE))
plot_model(iraceResults, param_name="algorithm")

plot_model(iraceResults, param_name="alpha")
```

report *Create HTML Report from irace data*

Description

This function creates an HTML report of the most relevant irace data. This report provides general statistics and plots that show the best configurations and their performance. Example: https://auto-optimization.github.io/iraceplot/articles/example/report_example.html

Usage

```
report(
  irace_results,
  filename = "report",
  sections = list(experiments_matrix = NULL, convergence = FALSE),
  interactive = base::interactive()
)
```

Arguments

irace_results	The data generated when loading the .Rdata file created by irace (or the filename of that file).
filename	(character(1)) Filename indicating where to save the report (example: "~/path-to/filename").
sections	(list()) List of sections to enable/disable. This is useful for disabling sections that may cause problems, such as out-of-memory errors. NA means automatically enable/disable a section depending on the memory required.
interactive	(logical(1)) TRUE if the report may use interactive features (using plotly::ggplotly(), plotly::plot_ly() and DT::renderDataTable()) or FALSE if such features must be disabled. Defaults to the value returned by interactive(),

Value

filename where the report was created or it opens the report in the default browser (interactive).

Examples

```
withr::with_tempdir({
  iraceResults <- read_logfile(system.file(package="irace", "exdata",
                                           "irace-acotsp.Rdata", mustWork = TRUE))
  report(iraceResults, filename = file.path(getwd(), "report"))
}, clean = !base::interactive())
```

`sampling_distance` *Sampling distance Plot*

Description

The `sampling_distance` function creates a plot that displays the mean of the distance between the configurations that were executed in each iteration.

For categorical parameters the distance is calculated as the hamming distance, for numerical parameters a equality interval is defined by a threshold specified by argument t and hamming distance is calculated using this interval.

Usage

```
sampling_distance(
  irace_results,
  type = c("boxplot", "line", "both"),
  t = 0.05,
  filename = NULL
)
```

Arguments

`irace_results`	The data generated when loading the `.Rdata` file created by `irace` (or the filename of that file).
`type`	String, (default "boxplot") Type of plot to be produces, either "line", "boxplot" or "both". The "boxplot" setting shows a boxplot of the mean distance of all configurations, "line" shows the mean distance of the solution population in each iteration, "both" shows both plots.
`t`	Numeric, (default 0.05) percentage factor that will determine a distance to define equal numerical parameter values. If the numerical parameter values to be compared are v1 and v2 they are considered equal if `|v1-v2| <= |ub-lb|*t`.
`filename`	(`character(1)`) File name to save the plot, for example `"~/path/example/filename.png"`.

Value

line or box plot

Examples

```
iraceResults <- read_logfile(system.file(package="irace", "exdata",
                                         "irace-acotsp.Rdata", mustWork = TRUE))
sampling_distance(iraceResults)

sampling_distance(iraceResults, type = "boxplot", t=0.07)
```

`sampling_frequency` *Parameter Frequency and Density Plot*

Description

Frequency or density plot that depicts the sampling performed by irace across the iterations of the configuration process. For categorical parameters a frequency plot is created, while for numerical parameters a histogram and density plots are created. The plots are shown in groups of maximum 9, the parameters included in the plot can be specified by setting the param_names argument.

Usage

```
sampling_frequency(
  configurations,
  parameters,
  param_names = NULL,
  n = NULL,
  filename = NULL
)
```

Arguments

`configurations`	(`data.frame()`) Configurations in `irace` format. Example: `iraceResults$allConfigurations`.
`parameters`	(`list()`) Parameters object in `irace` format. If this argument is missing, the first parameter is taken as the `iraceResults` data generated when loading the `.Rdata` file created by `irace` and `configurations=iraceResults$allConfigurations` and `parameters = iraceResults$scenario$parameters`.
`param_names`	(`character()`) Parameters to be included in the plot. Example: `c("algorithm","alpha","rho","q0",`
`n`	Numeric, for scenarios with large parameter sets, it selects a subset of 9 parameters. For example, `n=1` selects the first 9 (1 to 9) parameters, `n=2` selects the next 9 (10 to 18) parameters and so on.
`filename`	(`character(1)`) File name to save the plot, for example `"~/path/example/filename.png"`.

Value

Frequency and/or density plot

Note

If there are more than 9 parameters, a pdf file extension is recommended as it allows to create a multi-page document. Otherwise, you can use the `n` argument of the function to generate the plot of a subset of the parameters.

Examples

```
# Either use iraceResults
iraceResults <- read_logfile(system.file(package="iraceplot", "exdata",
                                         "guide-example.Rdata", mustWork = TRUE))
sampling_frequency(iraceResults)

sampling_frequency(iraceResults, n = 2)
sampling_frequency(iraceResults, param_names = c("alpha"))
sampling_frequency(iraceResults, param_names = c("algorithm", "alpha", "rho", "q0", "rasrank"))

# Or explicitly specify the configurations and parameters.
parameters <- iraceResults$scenario$parameters
sampling_frequency(iraceResults$allConfigurations, parameters)

sampling_frequency(iraceResults$allConfigurations, parameters, n = 2)
sampling_frequency(iraceResults$allConfigurations, parameters,
                   param_names = c("alpha"))
sampling_frequency(iraceResults$allConfigurations, parameters,
                   param_names = c("algorithm", "alpha", "rho", "q0", "rasrank"))
```

---

sampling_frequency_iteration

Frequency and Density plot based on its iteration

Description

The function will return a frequency plot used for categorical data (its values are string, show a bar plot) or numeric data (show a histogram and density plot) by each iteration

Usage

```
sampling_frequency_iteration(
  irace_results,
  param_name,
  numerical_type = "both",
  filename = NULL
)
```

Arguments

irace_results	The data generated when loading the .Rdata file created by irace (or the filename of that file).
param_name	String, name of the parameter to be included (example: param_name = "algorithm")
numerical_type	String, (default "both") Indicates the type of plot to be displayed for numerical parameters. "density" shows a density plot, "frequency" shows a frequency plot and "both" show both frequency and density.
filename	(character(1)) File name to save the plot, for example "~/path/example/filename.png".

Value

Frequency and/or density plot

Examples

```
iraceResults <- read_logfile(system.file(package="irace", "exdata",
                                         "irace-acotsp.Rdata", mustWork = TRUE))
sampling_frequency_iteration(iraceResults, param_name = "alpha")

sampling_frequency_iteration(iraceResults, param_name = "alpha", numerical_type="density")
```

`sampling_heatmap` *Sampling heat map plot*

Description

Heatmap that displays the frequency of sampling values of two parameters.

Usage

```
sampling_heatmap(
  irace_results,
  param_names,
  sizes = c(0, 0),
  iterations = NULL,
  only_elite = TRUE,
  filename = NULL
)
```

Arguments

`irace_results`	The data generated when loading the `.Rdata` file created by `irace` (or the filename of that file).
`param_names`	(`character()`) Parameters to be included in the plot. Example: `c("algorithm","alpha","rho","q0",`
`sizes`	Numeric vector that indicated the number of intervals to be considered for numerical parameters. This argument is positional with respect to param_names. By default, numerical parameters are displayed using 10 intervals. (example sizes = c(0,10))
`iterations`	Numeric vector, iteration number that should be included in the plot (example: iterations = c(1,4,5))
`only_elite`	logical (default TRUE), only print elite configurations.
`filename`	(`character(1)`) File name to save the plot, for example `"~/path/example/filename.png"`.

Value

sampling heat map plot

Examples

```
iraceResults <- read_logfile(system.file(package="irace", "exdata",
                                         "irace-acotsp.Rdata", mustWork = TRUE))
sampling_heatmap(iraceResults, param_names=c("beta", "alpha"))
sampling_heatmap(iraceResults, param_names=c("beta", "alpha"), iterations = c(3,4))
sampling_heatmap(iraceResults, param_names=c("beta", "alpha"), only_elite = FALSE)
```

`sampling_heatmap2` *Sampling heat map plot*

Description

Heatmap that displays the frequency of sampling values of two parameters.

Usage

```
sampling_heatmap2(
  configurations,
  parameters,
  param_names,
  sizes = c(0, 0),
  filename = NULL
)
```

Arguments

`configurations`	Data frame, configurations in `irace` format (example: `configurations = iraceResults$allConfigura`
`parameters`	(`list()`) Parameter space in irace format. See the function `irace::readParameters()`.
`param_names`	String vector of size 2, names of the parameters that should be included in the plot (example: param_names = c("beta","alpha"))
`sizes`	Numeric vector that indicated the number of intervals to be considered for numerical parameters. This argument is positional with respect to param_names. By default, numerical parameters are displayed using 10 intervals. (example sizes = c(0,10))
`filename`	(`character(1)`) File name to save the plot, for example `"~/path/example/filename.png"`.

Value

sampling heat map plot

Examples

```
iraceResults <- read_logfile(system.file(package="irace", "exdata",
                                         "irace-acotsp.Rdata", mustWork = TRUE))
sampling_heatmap2(iraceResults$allConfigurations, iraceResults$scenario$parameters,
                  param_names=c("beta", "alpha"))
```

sampling_pie *Sampling pie plot*

Description

This function creates a pie plot of the values sampled of a set of selected parameters. Numerical parameters are discretized to maximum n_bins intervals. The size of the slices are proportional to the number of configurations that have assigned a parameter value within the rank or the value assigned to that slice. Parameters can be selected by providing their names in the param_names argument.

Usage

```
sampling_pie(irace_results, param_names = NULL, n_bins = 3, filename = NULL)
```

Arguments

irace_results	The data generated when loading the .Rdata file created by irace (or the filename of that file).
param_names	String vector, A set of parameters to be included (example: param_names = c("algorithm","dlb"))
n_bins	Numeric (default 3), number of intervals to generate for numerical parameters.
filename	(character(1)) File name to save the plot, for example "~/path/example/filename.png".

Value

Sampling pie plot

Examples

```
iraceResults <- read_logfile(system.file(package="irace", "exdata",
                                         "irace-acotsp.Rdata", mustWork = TRUE))
sampling_pie(iraceResults)

sampling_pie(iraceResults, param_names = c("algorithm", "dlb", "ants"))
```

scatter_performance *Performance Scatter Plot of Two Configurations*

Description

Create a scatter plot that displays the performance of two configurations on a provided experiment matrix. Each point in the plot represents an instance and the color of the points indicates if one configuration is better than the other.

Usage

```
scatter_performance(
  experiments,
  x_id,
  y_id,
  rpd = TRUE,
  filename = NULL,
  interactive = base::interactive(),
  instance_names = NULL
)

scatter_training(irace_results, ...)

scatter_test(irace_results, ...)
```

Arguments

`experiments`	Experiment matrix obtained from irace training or testing data. Configurations in columns and instances in rows. As in irace, column names (configurations ids) should be characters. Row names will be used as instance names.
`x_id, y_id`	Configuration IDs for x-axis and y-axis, respectively.
`rpd`	(`logical(1)`) TRUE to plot performance as the relative percentage deviation to best results per instance, FALSE to plot raw performance.
`filename`	(`character(1)`) File name to save the plot, for example `"~/path/example/filename.png"`.
`interactive`	(`logical(1)`) TRUE if the report may use interactive features (using `plotly::ggplotly()`, `plotly::plot_ly()` and `DT::renderDataTable()`) or FALSE if such features must be disabled. Defaults to the value returned by `interactive()`,
`instance_names`	Either a character vector of instance names in the same order as `rownames(experiments)` or a function that takes `rownames(experiments)` as input.
`irace_results`	The data generated when loading the `.Rdata` file created by `irace` (or the filename of that file).
`...`	Other arguments passed to `scatter_performance()`.

Details

The performance matrix is assumed to be provided in the format of the irace experiment matrix thus, NA values are allowed. Consequently the number of evaluations can differ between configurations due to the elimination process applied by irace. This plot only shows performance data only for instances in which both configurations are executed.

scatter_training() compares the performance of two configurations on the training instances. The performance data is obtained from the evaluations performed by irace during the execution process.

scatter_test() compares the performance of two configurations on the test instances. The performance data is obtained from the test evaluations performed by irace. Note that testing is not enabled by default in irace and should be enabled in the scenario setup. Moreover, configuration ids provided in x_id and y_id should belong to elite configuration set evaluated in the test (see the irace package user guide for more details).

Value

ggplot2::ggplot() object

Examples

```
iraceResults <- read_logfile(system.file(package="irace", "exdata",
                                         "irace-acotsp.Rdata", mustWork = TRUE))
best_id <- iraceResults$iterationElites[length(iraceResults$iterationElites)]
scatter_performance(iraceResults$experiments, x_id = 1, y_id = best_id)
iraceResults <- read_logfile(system.file(package="iraceplot", "exdata",
                                         "guide-example.Rdata", mustWork = TRUE))
scatter_training(iraceResults, x_id = 806, y_id = 809)

scatter_training(iraceResults, x_id = 806, y_id = 809, rpd = FALSE)

iraceResults <- read_logfile(system.file(package="iraceplot", "exdata",
                                         "guide-example.Rdata", mustWork = TRUE))
scatter_test(iraceResults, x_id = 92, y_id = 119)

scatter_test(iraceResults, x_id = 92, y_id = 119, rpd=FALSE)
```

---

summarise_by_configuration

Summarise by configuration

Description

Summarise by configuration

Usage

```
summarise_by_configuration(
  irace_results,
  elites_only = FALSE,
  instances = c("both", "train", "test")
)
```

Arguments

irace_results	The data generated when loading the .Rdata file created by irace (or the filename of that file).
elites_only	(logical(1)) If TRUE, only report the final elite configurations.
instances	(character(1)) Select data from the training instances ("train") or from the test instances if available ("test"). The default is from both ("both").

Value

tibble

Examples

```
irace_results <- read_logfile(system.file(package="irace", "exdata",
                                          "irace-acotsp.Rdata", mustWork = TRUE))
summarise_by_configuration(irace_results, instances = "train", elites_only = TRUE)
```

---

summarise_by_instance *Summarise by instance*

Description

Summarise by instance

Usage

```
summarise_by_instance(irace_results)
```

Arguments

irace_results	The data generated when loading the .Rdata file created by irace (or the filename of that file).

Value

tibble

Examples

```
irace_result <- read_logfile(system.file(package="irace", "exdata",
                                         "irace-acotsp.Rdata", mustWork = TRUE))
summarise_by_instance(irace_result)
```

---

summarise_by_iteration

Summarise by iteration

Description

Summarise by iteration

Usage

```
summarise_by_iteration(irace_results)
```

Arguments

irace_results	The data generated when loading the .Rdata file created by irace (or the filename of that file).

Value

tibble

Examples

```
irace_result <- read_logfile(system.file(package="irace", "exdata",
                                         "irace-acotsp.Rdata", mustWork = TRUE))
summarise_by_iteration(irace_result)
```

Index