

Package ‘IOHanalyzer’

June 24, 2020

Type Package

Title Data Analysis Part of 'IOHprofiler'

Version 0.1.4

Author Hao Wang [cre, aut],
Diederick Vermetten [aut],
Carola Doerr [aut],
Thomas Bäck [aut]

Maintainer Hao Wang <h.wang@liacs.leidenuniv.nl>

Description The data analysis module for the Iterative Optimization Heuristics Profiler ('IOHprofiler'). This module provides statistical analysis methods for the benchmark data generated by optimization heuristics, which can be visualized through a web-based interface. The benchmark data is usually generated by the experimentation module, called 'IOHexperimenter'. 'IOHanalyzer' also supports the widely used 'COCO' (Comparing Continuous Optimisers) data format for benchmarking.

License BSD_3_clause + file LICENSE

Encoding UTF-8

LazyData true

URL <http://iohprofiler.liacs.nl>,
<https://github.com/IOHprofiler/IOHanalyzer>

BugReports <https://github.com/IOHprofiler/IOHanalyzer/issues>

Imports Rcpp, magrittr, dplyr, data.table, ggplot2, plotly,
colorspace, colorRamps, RColorBrewer, shiny, markdown,
reshape2, shinyjs, colourpicker, bsplus, DT, igraph,
kableExtra, PlayerRatings, stringi

LinkingTo Rcpp

SystemRequirements C++11

RoxygenNote 7.0.0

Suggests testthat, withr, shinydashboard

Depends R (>= 2.10)

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-06-24 17:50:02 UTC

R topics documented:

==.DataSet	4
arrange	4
as.character.DataSet	5
AUC	5
bootstrap_RT	6
c.DataSet	7
c.DataSetList	7
cat.DataSet	8
check_format	8
clean_DataSetList	9
DataSet	9
DataSetList	10
dsl	11
dsl_large	11
ECDF	12
generate_data.Aggr	12
generate_data.AUC	13
generate_data.ECDF	14
generate_data.hist	14
generate_data.Parameters	15
generate_data.PMF	15
generate_data.Single_Function	16
get_algId	16
get_color_scheme	17
get_color_scheme_dt	17
get_default_ECDF_targets	18
get_dim	18
get_ECDF_targets	19
get_ERT	19
get_funcId	20
get_funvals	21
get_FV_overview	21
get_FV_sample	22
get_FV_summary	23
get_line_style	23
get_maxRT	24
get_overview	25
get_parId	25
get_PAR_name	26
get_PAR_sample	27
get_PAR_summary	28
get_RT_overview	29

get_RT_sample	29
get_RT_summary	30
get_runtimes	31
get_target_dt	32
glicko2_ranking	32
IOHanalyzer	33
IOH_plot_ly_default	34
limit.data	34
max_ERTs	35
mean_FVs	35
pairwise.test	36
Plot.FV.Aggregated	37
Plot.FV.ECDF_AUC	38
Plot.FV.ECDF_Per_Target	38
Plot.FV.ECDF_Single_Func	39
Plot.FV.Histogram	40
Plot.FV.Multi_Func	41
Plot.FV.Parameters	42
Plot.FV.PDF	43
Plot.FV.Single_Func	43
Plot.RT.Aggregated	44
Plot.RT.ECDF_AUC	45
Plot.RT.ECDF_Multi_Func	46
Plot.RT.ECDF_Per_Target	47
Plot.RT.ECDF_Single_Func	48
Plot.RT.Histogram	49
Plot.RT.Multi_Func	49
Plot.RT.Parameters	50
Plot.RT.PMF	51
Plot.RT.Single_Func	52
Plot.Stats.Glicko2_Candlestick	53
Plot.Stats.Significance_Graph	54
Plot.Stats.Significance_Heatmap	54
plot_general_data	55
print.DataSet	56
print.DataSetList	57
read_index_file	57
runServer	58
save_plotly	58
scan_index_file	59
seq_FV	59
seq_RT	60
set_color_scheme	61
SP	62
subset.DataSetList	62
summary.DataSet	63
summary.DataSetList	63
[.DataSetList	64

Index**65**

```
==.DataSet          S3 generic == operator for DataSets
```

Description

S3 generic == operator for DataSets

Usage

```
## S3 method for class 'DataSet'
dsL == dsR
```

Arguments

```
dsL          A 'DataSet' object
dsR          A 'DataSet' object
```

Value

True if the DataSets contain the same function, dimension and algorithm, and have the exact same attributes

Examples

```
ds1[[1]] == ds1[[2]]
```

```
arrange          S3 sort function for DataSetList
```

Description

Sorts a DataSetList based on the custom specified attributes ('algId', 'DIM' or 'funcId'). Default is as ascending, can be made descending by adding a - in front of the attribute. Sorting accross multiple attributes is supported, in the order they are specified.

Usage

```
arrange(ds1, ...)
```

```
## S3 method for class 'DataSetList'
arrange(ds1, ...)
```

Arguments

```
ds1          The DataSetList to sort
...          attribute by which 'ds1' is sorted. Multiple attributes can be specified.
```

Examples

```
arrange(dsl, DIM, -funcId, algId)
```

as.character.DataSet *S3 generic as.character operator for DataSet*

Description

S3 generic as.character operator for DataSet

Usage

```
## S3 method for class 'DataSet'  
as.character(x, verbose = F, ...)
```

Arguments

x	A DataSet object
verbose	Verbose mode, currently not implemented
...	Arguments passed to other methods

Value

A short description of the DataSet

Examples

```
as.character(dsl[[[1]])
```

AUC *Area Under Curve (Empirical Cumulative Distribution Function)*

Description

Area Under Curve (Empirical Cumulative Distribution Function)

Usage

```
AUC(fun, from = NULL, to = NULL)
```

```
## S3 method for class 'ECDF'  
AUC(fun, from = NULL, to = NULL)
```

Arguments

fun A ECDF object.
 from double. Starting point of the area on x-axis
 to double. Ending point of the area on x-axis

Value

a object of type 'ECDF'

Examples

```
ecdf <- ECDF(ds1,c(12,14))
AUC(ecdf, 0, 100)
```

bootstrap_RT *Bootstrapping for running time samples*

Description

Bootstrapping for running time samples

Usage

```
bootstrap_RT(x, max_eval, bootstrap.size)
```

Arguments

x A numeric vector. A sample of the running time.
 max_eval A numeric vector, containing the maximal running time in each run. It should have the same size as x
 bootstrap.size integer, the size of the bootstrapped sample

Value

A numeric vector of the bootstrapped running time sample

Examples

```
ds <- ds1[[1]]
x <- get_RT_sample(ds, ftarget = 16, output = 'long')
max_eval <- get_maxRT(ds1, output = 'long')
bootstrap_RT(x$RT, max_eval$maxRT, bootstrap.size = 30)
```

c.DataSet	<i>S3 concatenation function for DataSet</i>
-----------	--

Description

Concatenation for DataSets. Combines multiple runs from separate DataSets into a single DataSet object if all provided arguments have the same dimension, function ID and algorithm ID, and each contains only a single run. Currently does not support parameter tracking

Usage

```
## S3 method for class 'DataSet'  
c(...)
```

Arguments

... The DataSets to concatenate

Value

A new DataSet

Examples

```
c(ds1[[1]], ds1[[1]])
```

c.DataSetList	<i>S3 concatenation function for DataSetList</i>
---------------	--

Description

S3 concatenation function for DataSetList

Usage

```
## S3 method for class 'DataSetList'  
c(...)
```

Arguments

... The DataSetLists to concatenate

Value

A new DataSetList

Examples

```
c(ds1[1], ds1[3])
```

cat.DataSet	<i>S3 generic cat operator for DataSet</i>
-------------	--

Description

S3 generic cat operator for DataSet

Usage

```
cat.DataSet(x)
```

Arguments

x A DataSet object

Value

A short description of the DataSet

Examples

```
cat.DataSet(ds1[[1]])
```

check_format	<i>Check the format of data</i>
--------------	---------------------------------

Description

Throws a warning when multiple formats are found in the same folder.

Usage

```
check_format(path)
```

Arguments

path The path to the folder to check

Value

The format of the data in the given folder. Either 'COCO', 'IOHprofiler', 'NEVERGRAD' or 'SOS'.

Examples

```
path <- system.file("extdata", "ONE_PLUS_LAMDA_EA", package = "IOHanalyzer")
check_format(path)
```

clean_DataSetList	<i>Clean DataSetList object by concatenating DataSets</i>
-------------------	---

Description

Concatenates all DataSets with the same algorithm name, function id and dimension

Usage

```
clean_DataSetList(dsList)
```

Arguments

dsList The DataSetList object to clean

Examples

```
clean_DataSetList(dsl)
```

DataSet	<i>Constructor of S3 class 'DataSet'</i>
---------	--

Description

DataSet contains the following attributes * funId * DIM * algId * datafile * instance * maxEvals * finalFunEvals * comment * Additional attributes based on the original format

Usage

```
DataSet(info, verbose = F, maximization = NULL, format = IOHprofiler,
        subsampling = FALSE)
```

Arguments

info A List. Contains a set of in a *.info file.
 verbose Logical.
 maximization Logical. Whether the underlying optimization algorithm performs a maximization? Set to NULL to determine automatically based on format
 format A character. The format of data source, either 'IOHProfiler', 'COCO' or 'TWO_COL'
 subsampling Logical. Whether *.cdat files are subsampled?

Value

A S3 object 'DataSet'

Examples

```
path <- system.file('extdata', 'ONE_PLUS_LAMDA_EA', package = 'IOHanalyzer')
info <- read_index_file(file.path(path, 'IOHprofiler_f1_i1.info'))
DataSet(info[[1]])
```

DataSetList

S3 constructor of the 'DataSetList'

Description

Attributes funId DIM algId

Usage

```
DataSetList(path = NULL, verbose = T, print_fun = NULL,
             maximization = NULL, format = IOHprofiler, subsampling = FALSE)
```

Arguments

path	Path to the data files. Will look for all .info-files in this directory and use the corresponding datafiles to create the DataSetList
verbose	Logical.
print_fun	Function used to print output when in verbose mode
maximization	Logical. Whether the underlying optimization algorithm performs a maximization?
format	A character. The format of data source, options are: <ul style="list-style-type: none"> • 'IOHProfiler' • 'COCO' • 'TWO_COL' • 'COCO_BIOBJ' • 'NEVERGRAD' • 'SOS' These formats are specified in more detail in our github wiki.
subsampling	Logical. Whether *.cdat files are subsampled?

Value

A DataSetList object

Examples

```
path <- system.file("extdata", "ONE_PLUS_LAMDA_EA", package = "IOHanalyzer")
DataSetList(path)
```

dsl	<i>Example DataSetList used in tests / examples</i>
-----	---

Description

A DataSetList containing DataSets on 2 IOHProfiler functions from 2 algorithms in 16D

Usage

```
dsl
```

Format

```
DataSetList
```

Examples

```
summary(dsl)
```

dsl_large	<i>Larger example DataSetList used in tests / examples</i>
-----------	--

Description

A DataSetList containing DataSets on all IOHProfiler functions from 11 algorithms in 100D

Usage

```
dsl_large
```

Format

```
DataSetList
```

Examples

```
summary(dsl_large)
```

ECDF	<i>Empirical Cumulative Distribution Function of Runtime of a single data set</i>
------	---

Description

Empirical Cumulative Distribution Function of Runtime of a single data set

Usage

```
ECDF(ds, ftarget, ...)

## S3 method for class 'DataSet'
ECDF(ds, ftarget, ...)

## S3 method for class 'DataSetList'
ECDF(ds, ftarget, ...)
```

Arguments

ds	A DataSet or DataSetList object.
ftarget	A Numerical vector. Function values at which runtime values are consumed
...	Arguments passed to other methods

Value

a object of type 'ECDF'

Examples

```
ECDF(ds1,c(12,14))
ECDF(ds1[[1]],c(12,14))
```

generate_data.Aggr	<i>Generate dataframe of a single function/dimension pair</i>
--------------------	---

Description

This function generates a dataframe which can be easily plotted using the 'plot_general_data'-function

Usage

```
generate_data.Aggr(dsList, aggr_on = "funcId", targets = NULL,
  which = "by_RT")
```

Arguments

dsList	The DataSetList object
aggr_on	Which attribute to use for aggregation. Either 'funcId' or 'DIM'
targets	Optional list of target values (Runtime or target value)
which	Whether to use a fixed-target 'by_RT' perspective or fixed-budget 'by_FV'

Examples

```
generate_data.Aggr(ds1)
```

generate_data.AUC	<i>Generate dataframe of a single function/dimension pair</i>
-------------------	---

Description

This function generates a dataframe which can be easily plotted using the 'plot_general_data'-function

Usage

```
generate_data.AUC(dsList, targets, which = "by_RT")
```

Arguments

dsList	The DataSetList object
targets	A list of the target value for which to calculate the AUC (Runtime or target value)
which	Whether to use a fixed-target 'by_RT' perspective or fixed-budget 'by_FV'

Examples

```
generate_data.AUC(subset(ds1, funcId == 1), c(12, 16))
```

generate_data.ECDF *Generate dataframe of a single function/dimension pair*

Description

This function generates a dataframe which can be easily plotted using the 'plot_general_data'-function

Usage

```
generate_data.ECDF(dsList, targets, scale_log = F, which = "by_RT")
```

Arguments

dsList	The DataSetList object
targets	A list or data.table containing the targets per function / dimension. If this is a data.table, it needs columns 'target', 'DIM' and 'funcId'
scale_log	Whether to use logarithmic scaling or not
which	Whether to use a fixed-target 'by_RT' perspective or fixed-budget 'by_FV'

Examples

```
generate_data.ECDF(subset(dsl, funcId == 1), c(10, 15, 16))
```

generate_data.hist *Generate dataframe of a single function/dimension pair*

Description

This function generates a dataframe which can be easily plotted using the 'plot_general_data'-function

Usage

```
generate_data.hist(dsList, target, use.equal.bins = F, which = "by_RT")
```

Arguments

dsList	The DataSetList object
target	The target value (Runtime or target value)
use.equal.bins	Whether all bins should be equal size for each algorithm or not
which	Whether to use a fixed-target 'by_RT' perspective or fixed-budget 'by_FV'

Examples

```
generate_data.hist(subset(dsl, funcId == 1), target = 15, which = 'by_RT')
```

```
generate_data.Parameters
```

Generate dataframe of a single function/dimension pair

Description

This function generates a dataframe which can be easily plotted using the 'plot_general_data'-function

Usage

```
generate_data.Parameters(dsList, which = "by_RT", scale_log = F)
```

Arguments

dsList	The DataSetList object
which	Whether to use a fixed-target 'by_RT' perspective or fixed-budget 'by_FV'
scale_log	Whether to use logarithmic scaling or not

Examples

```
generate_data.Parameters(subset(ds1, funcId == 1))
```

```
generate_data.PMF
```

Generate dataframe of a single function/dimension pair for creating PDF or PMF plots

Description

This function generates a dataframe which can be easily plotted using the 'plot_general_data'-function

Usage

```
generate_data.PMF(dsList, target, which = "by_RT")
```

Arguments

dsList	The DataSetList object
target	The target value (Runtime or target value)
which	Whether to use a fixed-target 'by_RT' perspective or fixed-budget 'by_FV'

Examples

```
generate_data.PMF(subset(ds1, funcId == 1), target = 15, which = 'by_RT')
```

```
generate_data.Single_Function
```

Generate dataframe of a single function/dimension pair

Description

This function generates a dataframe which can be easily plotted using the 'plot_general_data'-function

Usage

```
generate_data.Single_Function(dsList, start = NULL, stop = NULL,
  scale_log = F, which = "by_RT", include_opts = F)
```

Arguments

dsList	The DataSetList object
start	Optional start value (Runtime or target value)
stop	Optional end value (Runtime or target value)
scale_log	Whether to use logarithmic scaling or not
which	Whether to use a fixed-target 'by_RT' perspective or fixed-budget 'by_FV'
include_opts	Whether or not to also include the best value hit by each algorithm to the generated datapoints

Examples

```
generate_data.Single_Function(subset(dsl, funcId == 1), which = 'by_RT')
```

```
get_algId
```

Get all algorithm ids present in a DataSetList

Description

Get all algorithm ids present in a DataSetList

Usage

```
get_algId(dsList)
```

Arguments

dsList	The DataSetList
--------	-----------------

Value

A sorted list of all unique algorithm ids which occur in the DataSetList

Examples

```
get_algId(dsl)
```

<code>get_color_scheme</code>	<i>Get colors according to the current colorScheme of the IOHanalyzer</i>
-------------------------------	---

Description

Get colors according to the current colorScheme of the IOHanalyzer

Usage

```
get_color_scheme(alnames_in)
```

Arguments

`alnames_in` List of algorithms for which to get colors

Examples

```
get_color_scheme(get_algId(dsl))
```

<code>get_color_scheme_dt</code>	<i>Get datatable of current color (and linestyle) scheme to file</i>
----------------------------------	--

Description

Get datatable of current color (and linestyle) scheme to file

Usage

```
get_color_scheme_dt()
```

Value

data.table object with 3 columns: alnames, colors, linestyles

Examples

```
get_color_scheme_dt()
```

`get_default_ECDF_targets`*Generate ECDF targets for a DataSetList*

Description

Generate ECDF targets for a DataSetList

Usage

```
get_default_ECDF_targets(data, format_func = as.integer)
```

Arguments

<code>data</code>	A DataSetList
<code>format_func</code>	function to format the targets

Value

a vector of targets

Examples

```
get_default_ECDF_targets(dsl)
```

`get_dim`*Get all dimensions present in a DataSetList*

Description

Get all dimensions present in a DataSetList

Usage

```
get_dim(dsList)
```

Arguments

<code>dsList</code>	The DataSetList
---------------------	-----------------

Value

A sorted list of all unique dimensions which occur in the DataSetList

Examples

```
get_dim(dsl)
```

get_ECDF_targets *Generation of default ECDF-targets*

Description

Generation of default ECDF-targets

Usage

```
get_ECDF_targets(dsList, type = "log-linear", number_targets = 10)
```

Arguments

dsList The DataSetList object for which to generate the targets

type The way to generate the targets. Either 'log-linear', 'linear' or 'bbob' (51 fixed targets, equal for all functions / dimensions)

number_targets The amount of targets to generate

Value

A data.table with 3 columns: funcId, DIM and target

Examples

```
get_ECDF_targets(dsl, 'linear', 10)
```

get_ERT *Get Expected RunTime*

Description

Get Expected RunTime

Usage

```
get_ERT(ds, ftarget, ...)

## S3 method for class 'DataSet'
get_ERT(ds, ftarget, ...)

## S3 method for class 'DataSetList'
get_ERT(ds, ftarget, algorithm = "all", ...)
```

Arguments

ds	A DataSet or DataSetList object
ftarget	The function target(s) for which to get the ERT
...	Arguments passed to other methods
algorithm	Which algorithms in the DataSetList to consider.

Value

A data.table containing the runtime samples for each provided target function value

Examples

```
get_ERT(dsl, 14)
get_ERT(dsl[[1]], 14)
```

get_funcId	<i>Get all function ids present in a DataSetList</i>
------------	--

Description

Get all function ids present in a DataSetList

Usage

```
get_funcId(dsList)
```

Arguments

dsList	The DataSetList
--------	-----------------

Value

A sorted list of all unique function ids which occur in the DataSetList

Examples

```
get_funcId(dsl)
```

get_funvals	<i>Get all function values present in a DataSetList</i>
-------------	---

Description

Get all function values present in a DataSetList

Usage

```
get_funvals(dsList)
```

Arguments

dsList	The DataSetList
--------	-----------------

Value

A list matrices of all function values which occur in the DataSetList

Examples

```
get_funvals(dsl)
```

get_FV_overview	<i>Get Function Value condensed overview</i>
-----------------	--

Description

Get Function Value condensed overview

Usage

```
get_FV_overview(ds, ...)

## S3 method for class 'DataSet'
get_FV_overview(ds, ...)

## S3 method for class 'DataSetList'
get_FV_overview(ds, algorithm = "all", ...)
```

Arguments

ds	A 'DataSet' or 'DataSetList' object
...	Arguments passed to other methods
algorithm	Which algorithms in the DataSetList to consider.

Value

A data.table containing the algorithm ID, best, worst and mean reached function values, the number of runs and available budget for the DataSet

Examples

```
get_FV_overview(dsl)
get_FV_overview(dsl[[1]])
get_FV_overview(dsl, algorithm = '(1+1)_greedy_hill_climber_1')
```

get_FV_sample	<i>Get Funtion Value Samples</i>
---------------	----------------------------------

Description

Get Funtion Value Samples

Usage

```
get_FV_sample(ds, ...)

## S3 method for class 'DataSet'
get_FV_sample(ds, runtime, output = "wide", ...)

## S3 method for class 'DataSetList'
get_FV_sample(ds, runtime, algorithm = "all", ...)
```

Arguments

ds	A DataSet or DataSetList object
...	Arguments passed to other methods
runtime	A Numerical vector. Runtimes at which function values are reached
output	A String. The format of the output data: 'wide' or 'long'
algorithm	Which algorithms in the DataSetList to consider.

Value

A data.table containing the function value samples for each provided target runtime

Examples

```
get_FV_sample(dsl, 100)
get_FV_sample(dsl[[1]], 100)
```

get_FV_summary	<i>Get Function Value Summary</i>
----------------	-----------------------------------

Description

Get Function Value Summary

Usage

```
get_FV_summary(ds, ...)

## S3 method for class 'DataSet'
get_FV_summary(ds, runtime, ...)

## S3 method for class 'DataSetList'
get_FV_summary(ds, runtime, algorithm = "all", ...)
```

Arguments

ds	A DataSet or DataSetList object
...	Arguments passed to other methods
runtime	A Numerical vector. Runtimes at which function values are reached
algorithm	Which algorithms in the DataSetList to consider.

Value

A data.table containing the function value statistics for each provided target runtime value

Examples

```
get_FV_summary(dsl, 100)
get_FV_summary(dsl[[1]], 100)
```

get_line_style	<i>Get line styles according to the current styleScheme of the IOHAnalyzer</i>
----------------	--

Description

Get line styles according to the current styleScheme of the IOHAnalyzer

Usage

```
get_line_style(algnames_in)
```

Arguments

algnames_in List of algorithms for which to get linestyles

Examples

```
get_line_style(get_algId(dsl))
```

get_maxRT	<i>Get the maximal running time</i>
-----------	-------------------------------------

Description

Get the maximal running time

Usage

```
get_maxRT(ds, ...)

## S3 method for class 'DataSet'
get_maxRT(ds, output = "wide", ...)

## S3 method for class 'DataSetList'
get_maxRT(ds, algorithm = "all", ...)
```

Arguments

ds A DataSet or DataSetList object

... Arguments passed to other methods

output The format of the outputted table: 'wide' or 'long'

algorithm Which algorithms in the DataSetList to consider.

Value

A data.table object containing the algorithm ID and the running time when the algorithm terminates in each run

Examples

```
get_maxRT(dsl)
get_maxRT(dsl[[1]])
```

get_overview	<i>Get condensed overview of datasets</i>
--------------	---

Description

Get condensed overview of datasets

Usage

```
get_overview(ds, ...)  
  
## S3 method for class 'DataSet'  
get_overview(ds, ...)  
  
## S3 method for class 'DataSetList'  
get_overview(ds, ...)
```

Arguments

ds	A DataSet or DataSetList object
...	Arguments passed to other methods

Value

A data.table containing some basic information about the provided DataSet(List)

Examples

```
get_overview(dsl)  
get_overview(dsl[[1]])
```

get_parId	<i>Get all parameter ids present in a DataSetList</i>
-----------	---

Description

Get all parameter ids present in a DataSetList

Usage

```
get_parId(dsList, which = "by_FV")
```

Arguments

dsList	The DataSetList
which	A string takes values in 'c('by_FV', 'by_RT')'. To choose the parameters aligned by the running time (RT) or the function value (FV). Note that parameters in each case are not necessary the same.

Value

A sorted list of all unique parameter ids which occur in the DataSetList

Examples

```
get_parId(ds1)
```

get_PAR_name	<i>Get the parameter names of the algorithm</i>
--------------	---

Description

Get the parameter names of the algorithm

Usage

```
get_PAR_name(ds, which)

## S3 method for class 'DataSet'
get_PAR_name(ds, which = "by_FV")
```

Arguments

ds	A DataSet object
which	a string takes it value in 'c('by_FV', 'by_RT')', indicating the parameters aligned against the running time (RT) or function value (FV). 'by_FV' is the default value.

Value

a character list of paramter names, if recorded in the data set

Examples

```
get_PAR_name(ds1[[1]])
```

get_PAR_sample	<i>Get Parameter Value Samples</i>
----------------	------------------------------------

Description

Get Parameter Value Samples

Usage

```
get_PAR_sample(ds, idxValue, ...)

## S3 method for class 'DataSet'
get_PAR_sample(ds, idxValue, parId = "all",
  which = "by_FV", output = "wide", ...)

## S3 method for class 'DataSetList'
get_PAR_sample(ds, idxValue, algorithm = "all",
  ...)
```

Arguments

ds	A DataSet or DataSetList object
idxValue	A Numerical vector. Index values at which parameter values are observed. The index value can either take its value in the range of running times, or function values. Such a value type is signified by 'which' parameter.
...	Arguments passed to other methods
parId	A character vector. Either 'all' or the name of parameters to be retrieved
which	A string takes values in 'c('by_FV', 'by_RT')', indicating the parameters to be retrieved are aligned against the running time (RT) or function value (FV). 'by_FV' is the default value.
output	A character. The format of the output data: 'wide' or 'long'
algorithm	Which algorithms in the DataSetList to consider.

Value

A data.table object containing parameter values aligned at each given target value

Examples

```
get_PAR_sample(dsl, 14)
get_PAR_sample(dsl[[1]], 14)
```

get_PAR_summary	<i>Get Parameter Value Summary</i>
-----------------	------------------------------------

Description

Get Parameter Value Summary

Usage

```
get_PAR_summary(ds, idxValue, ...)

## S3 method for class 'DataSet'
get_PAR_summary(ds, idxValue, parId = "all",
  which = "by_FV", ...)

## S3 method for class 'DataSetList'
get_PAR_summary(ds, idxValue, algorithm = "all",
  ...)
```

Arguments

ds	A DataSet or DataSetList object
idxValue	A Numerical vector. Index values at which parameter values are observed. The index value can either take its value in the range of running times, or function values. Such a value type is signified by 'which' parameter.
...	Arguments passed to other methods
parId	A character vector. Either 'all' or the name of parameters to be retrieved
which	A string takes values in 'c('by_FV', 'by_RT')', indicating the parameters to be retrieved are aligned against the running time (RT) or function value (FV). 'by_FV' is the default value.
algorithm	Which algorithms in the DataSetList to consider.

Value

A data.table object containing basic statistics of parameter values aligned at each given target value

Examples

```
get_PAR_summary(ds1, 14)
get_PAR_summary(ds1[[1]], 14)
```

get_RT_overview	<i>Get Runtime Value condensed overview</i>
-----------------	---

Description

Get Runtime Value condensed overview

Usage

```
get_RT_overview(ds, ...)  
  
## S3 method for class 'DataSet'  
get_RT_overview(ds, ...)  
  
## S3 method for class 'DataSetList'  
get_RT_overview(ds, algorithm = "all", ...)
```

Arguments

ds	A DataSet or DataSetList object
...	Arguments passed to other methods
algorithm	Which algorithms in the DataSetList to consider.

Value

A data.table containing the algorithm ID, minimum and maximum used evaluations, number of runs and available budget for the DataSet

Examples

```
get_RT_overview(ds1)  
get_RT_overview(ds1[[1]])
```

get_RT_sample	<i>Get RunTime Sample</i>
---------------	---------------------------

Description

Get RunTime Sample

Usage

```

get_RT_sample(ds, ftarget, ...)

## S3 method for class 'DataSet'
get_RT_sample(ds, ftarget, output = "wide", ...)

## S3 method for class 'DataSetList'
get_RT_sample(ds, ftarget, algorithm = "all", ...)

```

Arguments

ds	A DataSet or DataSetList object
ftarget	A Numerical vector. Function values at which runtime values are consumed
...	Arguments passed to other methods
output	A character determining the format of output data.table: 'wide' or 'long'
algorithm	Which algorithms in the DataSetList to consider.

Value

A data.table containing the runtime samples for each provided target function value

Examples

```

get_RT_sample(dsl, 14)
get_RT_sample(dsl[[1]], 14)

```

get_RT_summary	<i>Get RunTime Summary</i>
----------------	----------------------------

Description

Get RunTime Summary

Usage

```

get_RT_summary(ds, ftarget, ...)

## S3 method for class 'DataSet'
get_RT_summary(ds, ftarget, ...)

## S3 method for class 'DataSetList'
get_RT_summary(ds, ftarget, algorithm = "all", ...)

```

Arguments

- `ds` A DataSet or DataSetList object
- `ftarget` The function target(s) for which to get the runtime summary
- `...` Arguments passed to other methods
- `algorithm` Which algorithms in the DataSetList to consider.

Value

A data.table containing the runtime statistics for each provided target function value

Examples

```
get_RT_summary(dsl, 14)
get_RT_summary(dsl[[1]], 14)
```

<code>get_runtimes</code>	<i>Get all runtime values present in a DataSetList</i>
---------------------------	--

Description

Get all runtime values present in a DataSetList

Usage

```
get_runtimes(dsList)
```

Arguments

- `dsList` The DataSetList

Value

A list matrices of all runtime values which occur in the DataSetList

Examples

```
get_runtimes(dsl)
```

get_target_dt	<i>Generate datatables of runtime or function value targets for a DataSetList</i>
---------------	---

Description

Only one target is generated per (function, dimension)-pair, as opposed to the function 'get_default_ECDF_targets', which generates multiple targets.

Usage

```
get_target_dt(dsList, which = "by_RT")
```

Arguments

dsList	A DataSetList
which	Whether to generate fixed-target ('by_FV') or fixed-budget ('by_RT') targets

Value

a data.table of targets

Examples

```
get_target_dt(dsl)
```

glicko2_ranking	<i>Glicko2 ranking of algorithms</i>
-----------------	--------------------------------------

Description

This procedure ranks algorithms based on a glicko2-procedure. Every round (total nr_rounds), for every function and dimension of the datasetlist, each pair of algorithms competes. This competition samples a random runtime for the provided target (defaults to best achieved target). Whichever algorithm has the lower runtime wins the game. Then, from these games, the glicko2-rating is determined.

Usage

```
glicko2_ranking(dsl, nr_rounds = 100, which = "by_FV",
  target_dt = NULL)
```


Arguments

<code>dsl</code>	The DataSetList, can contain multiple functions and dimensions, but should have the same algorithms for all of them
<code>nr_rounds</code>	The number of rounds to run. More rounds leads to a more accurate ranking.
<code>which</code>	Whether to use fixed-target ('by_FV') or fixed-budget ('by_RT') perspective
<code>target_dt</code>	Custom data.table target value to use. When NULL, this is selected automatically.

Value

A dataframe containing the glicko2-ratings and some additional info

Examples

```
glicko2_ranking(dsl, nr_round = 25)
glicko2_ranking(dsl, nr_round = 25, which = 'by_RT')
```

IOHanalyzer

IOHanalyzer: Data Analysis Part of IOHprofiler

Description

The data analysis module for the Iterative Optimization Heuristics Profiler (IOHprofiler). This module provides statistical analysis methods for the benchmark data generated by optimization heuristics, which can be visualized through a web-based interface. The benchmark data is usually generated by the experimentation module, called IOHexperimenter. IOHanalyzer also supports the widely used COCO (Comparing Continuous Optimisers) data format for benchmarking.

Functions

The IOHanalyzer consists of 3 main functionalities:

- Reading and aligning data from different heuristics, such as IOHexperimenter. This is done using the [DataSet](#) and [DataSetList](#) functions
- Processing and summarizing this data
- Creating various plots

Examples

```
path <- system.file("extdata", "ONE_PLUS_LAMDA_EA", package="IOHanalyzer")
dsList <- DataSetList(path)
summary(dsList)
Plot.RT.Single_Func(dsList[1])

## Not run:
runServer()

## End(Not run)
```

`IOH_plot_ly_default` *Template for creating plots in the IOHanalyzer-style*

Description

Template for creating plots in the IOHanalyzer-style

Usage

```
IOH_plot_ly_default(title = NULL, x.title = NULL, y.title = NULL)
```

Arguments

<code>title</code>	Title for the plot
<code>x.title</code>	X-axis label
<code>y.title</code>	Y-axis label

Examples

```
IOH_plot_ly_default("Example plot", "x-axis", "y-axis")
```

`limit.data` *Reduce the size of the data set by evenly subsampling the records*

Description

Reduce the size of the data set by evenly subsampling the records

Usage

```
limit.data(df, n)
```

Arguments

<code>df</code>	The data to subsample
<code>n</code>	The amount of samples

Value

A smaller data.frame

max_ERTs	<i>Get the ERT-values for all DataSets in a DataSetList at certain targets</i>
----------	--

Description

Get the ERT-values for all DataSets in a DataSetList at certain targets

Usage

```
max_ERTs(dsList, aggr_on = "funcId", targets = NULL, maximize = T)
```

```
## S3 method for class 'DataSetList'
max_ERTs(dsList, aggr_on = "funcId",
  targets = NULL, maximize = T)
```

Arguments

dsList	The DataSetList
aggr_on	Whether to aggregate on 'funcId' or 'DIM'.
targets	Predifined target function-values. Should be one for each function/dimension
maximize	Whether the DataSetList is from a maximization or minimization problem

Value

A data.table containing ERT-values

Examples

```
max_ERTs(dsl)
```

mean_FVs	<i>Get the expected function-values for all DataSets in a DataSetList at certain runtimes</i>
----------	---

Description

Get the expected function-values for all DataSets in a DataSetList at certain runtimes

Usage

```
mean_FVs(dsList, aggr_on = "funcId", runtimes = NULL)
```

```
## S3 method for class 'DataSetList'
mean_FVs(dsList, aggr_on = "funcId",
  runtimes = NULL)
```

Arguments

dsList	The DataSetList
aggr_on	Whether to aggregate on 'funcId' or 'DIM'.
runtimes	Predifined target runtimes-values. Should be one for each function/dimension

Value

A data.table containing expected function-values

Examples

```
mean_FVs(dsl)
```

pairwise.test	<i>Performs a pairwise Kolmogorov-Smirnov test on the bootstrapped running times among a data set</i>
---------------	---

Description

This function performs a Kolmogorov-Smirnov test on each pair of algorithms in the input x to determine which algorithm gives a significantly smaller running time. The resulting p-values are arranged in a matrix, where each cell (i, j) contains a p-value from the test with alternative hypothesis: the running time of algorithm i is smaller (thus better) than that of j.

Usage

```
pairwise.test(x, ...)

## S3 method for class 'list'
pairwise.test(x, max_eval, bootstrap.size = 30, ...)

## S3 method for class 'DataSetList'
pairwise.test(x, ftarget, bootstrap.size = 0,
  which = "by_FV", ...)
```

Arguments

x	either a list that contains running time sample for each algorithm as sub-lists, or a DataSetList object
...	all other options
max_eval	list that contains the maximal running time for each algorithm as sub-lists
bootstrap.size	integer, the size of the bootstrapped sample. Set to 0 to disable bootstrapping
ftarget	float, the target value used to determine the running / hitting
which	whether to do fixed-target ('by_FV') or fixed-budget ('by_RT') comparison time

Value

A matrix containing p-values of the test

Examples

```
pairwise.test(subset(dsl, funcId == 1), 16)
```

Plot.FV.Agregated	<i>Plot expected function value-based comparison over multiple functions or dimensions</i>
-------------------	--

Description

Plot expected function value-based comparison over multiple functions or dimensions

Usage

```
Plot.FV.Agregated(dsList, aggr_on = "funcId", runtimes = NULL,
  plot_mode = "radar", use_rank = F, scale.ylog = T, fvs = NULL)
```

```
## S3 method for class 'DataSetList'
Plot.FV.Agregated(dsList, aggr_on = "funcId",
  runtimes = NULL, plot_mode = "radar", use_rank = F,
  scale.ylog = T, fvs = NULL)
```

Arguments

dsList	A DataSetList (should consist of only one function OR dimension).
aggr_on	Whether to compare on functions ('funcId') or dimensions ('DIM')
runtimes	Custom list of function-value targets, one for each function or dimension.
plot_mode	How the plots should be created. Can be 'line' or 'radar'
use_rank	Wheter to use a ranking system. If False, the actual expected function- values will be used.
scale.ylog	Whether or not to scale the y-axis logarithmically
fvs	Pre-calculated expected function-values for the provided runtimes Created by the max_ERTs function of DataSetList. Can be provided to prevent needless computation in recalculating ERTs when recreating this plot.

Value

A plot of expected function value-based comparison on the provided functions or dimensions of the DataSetList

Examples

```
Plot.FV.Agregated(dsl)
```

Plot.FV.ECDF_AUC	<i>Radarplot of the area under the aggregated ECDF-curve of a DataSetList.</i>
------------------	--

Description

Radarplot of the area under the aggregated ECDF-curve of a DataSetList.

Usage

```
Plot.FV.ECDF_AUC(dsList, rt_min = NULL, rt_max = NULL,
  rt_step = NULL)
```

```
## S3 method for class 'DataSetList'
Plot.FV.ECDF_AUC(dsList, rt_min = NULL,
  rt_max = NULL, rt_step = NULL)
```

Arguments

dsList	A DataSetList (should consist of only one function and dimension).
rt_min	The starting runtime
rt_max	The final runtime
rt_step	The spacing between starting and final runtimes

Value

A radarplot of the area under the aggregated ECDF-curve of the DataSetList

Examples

```
Plot.FV.ECDF_AUC(subset(ds1, funcId == 1))
```

Plot.FV.ECDF_Per_Target	<i>Plot the empirical cumulative distribution as a function of the target values of a DataSetList at certain target runtimes</i>
-------------------------	--

Description

Plot the empirical cumulative distribution as a function of the target values of a DataSetList at certain target runtimes

Usage

```
Plot.FV.ECDF_Per_Target(dsList, runtimes, scale.xlog = F,
  scale.reverse = F)
```

```
## S3 method for class 'DataSetList'
Plot.FV.ECDF_Per_Target(dsList, runtimes,
  scale.xlog = F, scale.reverse = F)
```

Arguments

dsList	A DataSetList (should consist of only one function and dimension).
runtimes	The target runtimes
scale.xlog	Whether or not to scale the x-axis logarithmically
scale.reverse	Whether or not to reverse the x-axis (when using minimization)

Value

A plot of the empirical cumulative distribution as a function of the function values of the DataSetList at the target runtimes

Examples

```
Plot.FV.ECDF_Per_Target(subset(dsl, funcId == 1), 10)
```

```
Plot.FV.ECDF_Single_Func
```

Plot the aggregated empirical cumulative distribution as a function of the function values of a DataSetList.

Description

Plot the aggregated empirical cumulative distribution as a function of the function values of a DataSetList.

Usage

```
Plot.FV.ECDF_Single_Func(dsList, rt_min = NULL, rt_max = NULL,
  rt_step = NULL, scale.xlog = F, show.per_target = F,
  scale.reverse = F)
```

```
## S3 method for class 'DataSetList'
Plot.FV.ECDF_Single_Func(dsList, rt_min = NULL,
  rt_max = NULL, rt_step = NULL, scale.xlog = F,
  show.per_target = F, scale.reverse = F)
```

Arguments

<code>dsList</code>	A DataSetList (should consist of only one function and dimension).
<code>rt_min</code>	The starting runtime
<code>rt_max</code>	The final runtime
<code>rt_step</code>	The spacing between starting and final runtimes
<code>scale.xlog</code>	Whether or not to scale the x-axis logarithmically
<code>show.per_target</code>	Whether or not to show the individual ECDF-curves for each runtime
<code>scale.reverse</code>	Whether or not to reverse the x-axis (when using minimization)

Value

A plot of the empirical cumulative distribution as a function of the function values of the DataSetList

Examples

```
Plot.FV.ECDF_Single_Func(subset(ds1, funcId == 1))
```

<code>Plot.FV.Histogram</code>	<i>Plot histograms of the function values of a DataSetList at a certain target runtime</i>
--------------------------------	--

Description

Plot histograms of the function values of a DataSetList at a certain target runtime

Usage

```
Plot.FV.Histogram(dsList, runtime, plot_mode = "overlay",
  use.equal.bins = F)
```

```
## S3 method for class 'DataSetList'
Plot.FV.Histogram(dsList, runtime,
  plot_mode = "overlay", use.equal.bins = F)
```

Arguments

<code>dsList</code>	A DataSetList (should consist of only one function and dimension).
<code>runtime</code>	The target runtime
<code>plot_mode</code>	How to plot the different histograms for each algorithm. Can be either 'overlay' to show all algorithms on one plot, or 'subplot' to have one plot per algorithm.
<code>use.equal.bins</code>	Whether to determine one bin size for all plots or have individual bin sizes for each algorithm

Value

A plot of the histograms of the function values at a the target runtime of the DataSetList

Examples

```
Plot.FV.Histogram(subset(dsl, funcId == 1), 100)
```

Plot.FV.Multi_Func *Plot FV-plots for multiple functions or dimensions*

Description

Plot FV-plots for multiple functions or dimensions

Usage

```
Plot.FV.Multi_Func(dsList, scale.xlog = F, scale.ylog = F,  
  backend = NULL)
```

```
## S3 method for class 'DataSetList'  
Plot.FV.Multi_Func(dsList, scale.xlog = F,  
  scale.ylog = F, backend = NULL)
```

Arguments

dsList	A DataSetList (should consist of only one function OR dimension).
scale.xlog	Whether or not to scale the x-axis logarithmically
scale.ylog	Whether or not to scale the y-axis logarithmically
backend	Which plotting library to use. Either 'plotly' or 'ggplot2'.

Value

A plot of Function-values of the DataSetList

Examples

```
Plot.FV.Multi_Func(dsl)
```

Plot.FV.Parameters	<i>Plot the parameter values recorded in a DataSetList (aligned by budget)</i>
--------------------	--

Description

Plot the parameter values recorded in a DataSetList (aligned by budget)

Usage

```
Plot.FV.Parameters(dsList, rt_min = NULL, rt_max = NULL,
  algids = "all", par_name = NULL, scale.xlog = F, scale.ylog = F,
  show.mean = T, show.median = F, show.CI = F)
```

```
## S3 method for class 'DataSetList'
Plot.FV.Parameters(dsList, rt_min = NULL,
  rt_max = NULL, algids = "all", par_name = NULL, scale.xlog = F,
  scale.ylog = F, show.mean = T, show.median = F, show.CI = F)
```

Arguments

dsList	A DataSetList (should consist of only one function and dimension).
rt_min	The starting budget value.
rt_max	The final budget value.
algids	Which algorithms from dsList to use
par_name	Which parameters to create plots for; set to NULL to use all parameters found in dsList.
scale.xlog	Whether or not to scale the x-axis logarithmically
scale.ylog	Whether or not to scale the y-axis logarithmically
show.mean	Whether or not to show the mean parameter values
show.median	Whether or not to show the median parameter values
show.CI	Whether or not to show the standard deviation

Value

A plot of for every recorded parameter in the DataSetList

Examples

```
Plot.FV.Parameters(subset(ds1, funcId == 1))
```

Plot.FV.PDF	<i>Plot probability density function of the function values of a DataSetList at a certain target runtime</i>
-------------	--

Description

Plot probability density function of the function values of a DataSetList at a certain target runtime

Usage

```
Plot.FV.PDF(dsList, runtime, show.sample = F, scale.ylog = F)
```

```
## S3 method for class 'DataSetList'  
Plot.FV.PDF(dsList, runtime, show.sample = F,  
            scale.ylog = F)
```

Arguments

dsList	A DataSetList (should consist of only one function and dimension).
runtime	The target runtime
show.sample	Whether or not to show the individual function value samples
scale.ylog	Whether or not to scale the y-axis logarithmically

Value

A plot of the probability density function of the runtimes at a the target function value of the DataSetList

Examples

```
Plot.FV.PDF(subset(ds1, funcId == 1), 100)
```

Plot.FV.Single_Func	<i>Plot lineplot of the expected function values of a DataSetList</i>
---------------------	---

Description

Plot lineplot of the expected function values of a DataSetList

Usage

```
Plot.FV.Single_Func(dsList, RTstart = NULL, RTstop = NULL,
  show.CI = F, show.mean = T, show.median = F, backend = NULL,
  scale.xlog = F, scale.ylog = F, scale.reverse = F)

## S3 method for class 'DataSetList'
Plot.FV.Single_Func(dsList, RTstart = NULL,
  RTstop = NULL, show.CI = F, show.mean = T, show.median = F,
  backend = NULL, scale.xlog = F, scale.ylog = F,
  scale.reverse = F)
```

Arguments

<code>dsList</code>	A <code>DataSetList</code> (should consist of only one function and dimension).
<code>RTstart</code>	The starting runtime value.
<code>RTstop</code>	The final runtime value.
<code>show.CI</code>	Whether or not to show the standard deviations
<code>show.mean</code>	Whether or not to show the mean runtimes
<code>show.median</code>	Whether or not to show the median runtimes
<code>backend</code>	Which plotting library to use. Can be 'plotly' or 'ggplot2'
<code>scale.xlog</code>	Whether or not to scale the x-axis logarithmically
<code>scale.ylog</code>	Whether or not to scale the y-axis logarithmically
<code>scale.reverse</code>	Whether or not to reverse the x-axis (when using minimization)

Value

A plot of ERT-values of the `DataSetList`

Examples

```
Plot.FV.Single_Func(subset(ds1, funcId == 1))
```

Plot.RT.Aggregated	<i>Plot ERT-based comparison over multiple functions or dimensions</i>
--------------------	--

Description

Plot ERT-based comparison over multiple functions or dimensions

Usage

```

Plot.RT.Agregated(dsList, aggr_on = "funcId", targets = NULL,
  plot_mode = "radar", use_rank = F, scale.ylog = T, maximize = T,
  erts = NULL, inf.action = "overlap")

## S3 method for class 'DataSetList'
Plot.RT.Agregated(dsList, aggr_on = "funcId",
  targets = NULL, plot_mode = "radar", use_rank = F,
  scale.ylog = T, maximize = T, erts = NULL,
  inf.action = "overlap")

```

Arguments

dsList	A DataSetList (should consist of only one function OR dimension).
aggr_on	Whether to compare on functions ('funcId') or dimensions ('DIM')
targets	Custom list of function-value targets, one for each function or dimension.
plot_mode	How the plots should be created. Can be 'line' or 'radar'
use_rank	Wheter to use a ranking system. If False, the actual ERT-values will be used.
scale.ylog	Whether or not to scale the y-axis logarithmically
maximize	Wheter or not to the data is of a maximization problem
erts	Pre-calculated ERT-values for the provided targets. Created by the max_ERTs function of DataSetList. Can be provided to prevent needless computation in recalculating ERTs when recreating this plot.
inf.action	How to handle infinite ERTs ('overlap' or 'jitter')

Value

A plot of ERT-based comparison on the provided functions or dimensions of the DataSetList

Examples

```
Plot.RT.Agregated(ds1)
```

Plot.RT.ECDF_AUC	<i>Radarplot of the area under the aggregated ECDF-curve of a DataSetList.</i>
------------------	--

Description

Radarplot of the area under the aggregated ECDF-curve of a DataSetList.

Usage

```
Plot.RT.ECDF_AUC(dsList, fstart = NULL, fstop = NULL, fstep = NULL,
  fval_formatter = as.integer)

## S3 method for class 'DataSetList'
Plot.RT.ECDF_AUC(dsList, fstart = NULL,
  fstop = NULL, fstep = NULL, fval_formatter = as.integer)
```

Arguments

dsList	A DataSetList (should consist of only one function and dimension).
fstart	The starting function value
fstop	The final function value
fstep	The spacing between starting and final function values
fval_formatter	Function to format the function-value labels

Value

A radarplot of the area under the aggregated ECDF-curve of the DataSetList

Examples

```
Plot.RT.ECDF_AUC(subset(ds1, funcId == 1))
```

```
Plot.RT.ECDF_Multi_Func
```

Plot the aggregated empirical cumulative distribution as a function of the running times of a DataSetList. Aggregated over multiple functions or dimensions.

Description

Plot the aggregated empirical cumulative distribution as a function of the running times of a DataSetList. Aggregated over multiple functions or dimensions.

Usage

```
Plot.RT.ECDF_Multi_Func(dsList, targets = NULL, scale.xlog = F)

## S3 method for class 'DataSetList'
Plot.RT.ECDF_Multi_Func(dsList, targets = NULL,
  scale.xlog = F)
```

Arguments

<code>dsList</code>	A DataSetList.
<code>targets</code>	The target function values. Specified in a data.frame, as can be generated
<code>scale.xlog</code>	Whether or not to scale the x-axis logarithmically by the function 'get_default_ECDF_targets'

Value

A plot of the empirical cumulative distribution as a function of the running times of the DataSetList

Examples

```
Plot.RT.ECDF_Multi_Func(ds1)
```

```
Plot.RT.ECDF_Per_Target
```

Plot the empirical cumulative distribution as a function of the running times of a DataSetList at certain target function values

Description

Plot the empirical cumulative distribution as a function of the running times of a DataSetList at certain target function values

Usage

```
Plot.RT.ECDF_Per_Target(dsList, ftargets, scale.xlog = F)
```

```
## S3 method for class 'DataSetList'
Plot.RT.ECDF_Per_Target(dsList, ftargets,
  scale.xlog = F)
```

Arguments

<code>dsList</code>	A DataSetList (should consist of only one function and dimension).
<code>ftargets</code>	The target function values
<code>scale.xlog</code>	Whether or not to scale the x-axis logarithmically

Value

A plot of the empirical cumulative distribution as a function of the running times of the DataSetList at the target function values

Examples

```
Plot.RT.ECDF_Per_Target(subset(ds1, funcId == 1), 14)
```

Plot.RT.ECDF_Single_Func

Plot the aggregated empirical cumulative distribution as a function of the running times of a DataSetList.

Description

Plot the aggregated empirical cumulative distribution as a function of the running times of a DataSetList.

Usage

```
Plot.RT.ECDF_Single_Func(dsList, fstart = NULL, fstop = NULL,
  fstep = NULL, show.per_target = F, scale.xlog = F)

## S3 method for class 'DataSetList'
Plot.RT.ECDF_Single_Func(dsList, fstart = NULL,
  fstop = NULL, fstep = NULL, show.per_target = F, scale.xlog = F)
```

Arguments

dsList	A DataSetList (should consist of only one function and dimension).
fstart	The starting function value
fstop	The final function value
fstep	The spacing between starting and final function values
show.per_target	Whether or not to show the individual ECDF-curves for each target
scale.xlog	Whether or not to scale the x-axis logarithmically

Value

A plot of the empirical cumulative distribution as a function of the running times of the DataSetList

Examples

```
Plot.RT.ECDF_Single_Func(subset(dsl, funcId == 1))
```

Plot.RT.Histogram	<i>Plot histograms of the runtimes of a DataSetList at a certain target function value</i>
-------------------	--

Description

Plot histograms of the runtimes of a DataSetList at a certain target function value

Usage

```
Plot.RT.Histogram(dsList, ftarget, plot_mode = "overlay",
  use.equal.bins = F)
```

```
## S3 method for class 'DataSetList'
Plot.RT.Histogram(dsList, ftarget,
  plot_mode = "overlay", use.equal.bins = F)
```

Arguments

dsList	A DataSetList (should consist of only one function and dimension).
ftarget	The target function value.
plot_mode	How to plot the different histograms for each algorithm. Can be either 'overlay' to show all algorithms on one plot, or 'subplot' to have one plot per algorithm.
use.equal.bins	Whether to determine one bin size for all plots or have individual bin sizes for each algorithm

Value

A plot of the histograms of the runtimes at a the target function value of the DataSetList

Examples

```
Plot.RT.Histogram(subset(dsl, funcId == 1), 14)
```

Plot.RT.Multi_Func	<i>Plot ERT-plots for multiple functions or dimensions</i>
--------------------	--

Description

Plot ERT-plots for multiple functions or dimensions

Usage

```
Plot.RT.Multi_Func(dsList, scale.xlog = F, scale.ylog = F,
  scale.reverse = F, backend = NULL)
```

```
## S3 method for class 'DataSetList'
Plot.RT.Multi_Func(dsList, scale.xlog = F,
  scale.ylog = F, scale.reverse = F, backend = NULL)
```

Arguments

dsList	A DataSetList (should consist of only one function OR dimension).
scale.xlog	Whether or not to scale the x-axis logarithmically
scale.ylog	Whether or not to scale the y-axis logarithmically
scale.reverse	Whether or not to reverse the x-axis (when using minimization)
backend	Which plotting library to use. Either 'plotly' or 'ggplot2'.

Value

A plot of ERT-values of the DataSetList

Examples

```
Plot.RT.Multi_Func(dsl)
```

Plot.RT.Parameters	<i>Plot the parameter values recorded in a DataSetList (aligned by function value)</i>
--------------------	--

Description

Plot the parameter values recorded in a DataSetList (aligned by function value)

Usage

```
Plot.RT.Parameters(dsList, f_min = NULL, f_max = NULL,
  algids = "all", par_name = NULL, scale.xlog = F, scale.ylog = F,
  show.mean = T, show.median = F, show.CI = F)
```

```
## S3 method for class 'DataSetList'
Plot.RT.Parameters(dsList, f_min = NULL,
  f_max = NULL, algids = "all", par_name = NULL, scale.xlog = F,
  scale.ylog = F, show.mean = T, show.median = F, show.CI = F)
```

Arguments

<code>dsList</code>	A DataSetList (should consist of only one function and dimension).
<code>f_min</code>	The starting function value.
<code>f_max</code>	The final function value.
<code>algids</code>	Which algorithms from dsList to use
<code>par_name</code>	Which parameters to create plots for; set to NULL to use all parameters found in dsList.
<code>scale.xlog</code>	Whether or not to scale the x-axis logarithmically
<code>scale.ylog</code>	Whether or not to scale the y-axis logarithmically
<code>show.mean</code>	Whether or not to show the mean parameter values
<code>show.median</code>	Whether or not to show the median parameter values
<code>show.CI</code>	Whether or not to show the standard deviation

Value

A plot of for every recorded parameter in the DataSetList

Examples

```
Plot.RT.Parameters(subset(ds1, funcId == 1))
```

<code>Plot.RT.PMF</code>	<i>Plot probability mass function of the runtimes of a DataSetList at a certain target function value</i>
--------------------------	---

Description

Plot probability mass function of the runtimes of a DataSetList at a certain target function value

Usage

```
Plot.RT.PMF(dsList, ftarget, show.sample = F, scale.ylog = F,
            backend = NULL)
```

```
## S3 method for class 'DataSetList'
Plot.RT.PMF(dsList, ftarget, show.sample = F,
            scale.ylog = F, backend = NULL)
```

Arguments

<code>dsList</code>	A DataSetList (should consist of only one function and dimension).
<code>ftarget</code>	The target function value.
<code>show.sample</code>	Whether or not to show the individual runtime samples
<code>scale.ylog</code>	Whether or not to scale the y-axis logarithmically
<code>backend</code>	Which plotting library to use. Can be 'plotly' or 'ggplot2'

Value

A plot of the probability mass function of the runtimes at a the target function value of the DataSetList

Examples

```
Plot.RT.PMF(subset(ds1, funcId == 1), 14)
```

Plot.RT.Single_Func *Plot lineplot of the ERTs of a DataSetList*

Description

Plot lineplot of the ERTs of a DataSetList

Usage

```
Plot.RT.Single_Func(dsList, Fstart = NULL, Fstop = NULL,
  show.ERT = T, show.CI = F, show.mean = F, show.median = F,
  backend = NULL, scale.xlog = F, scale.ylog = F,
  scale.reverse = F, includeOpts = F, p = NULL)

## S3 method for class 'DataSetList'
Plot.RT.Single_Func(dsList, Fstart = NULL,
  Fstop = NULL, show.ERT = T, show.CI = T, show.mean = F,
  show.median = F, backend = NULL, scale.xlog = F, scale.ylog = F,
  scale.reverse = F, includeOpts = F, p = NULL)
```

Arguments

dsList	A DataSetList (should consist of only one function and dimension).
Fstart	The starting function value.
Fstop	The final function value.
show.ERT	Whether or not to show the ERT-values
show.CI	Whether or not to show the standard deviations
show.mean	Whether or not to show the mean hitting times
show.median	Whether or not to show the median hitting times
backend	Which plotting library to use. Can be 'plotly' or 'ggplot2'
scale.xlog	Whether or not to scale the x-axis logarithmically
scale.ylog	Whether or not to scale the y-axis logarithmically
scale.reverse	Wheter or not to reverse the x-axis (when using minimization)
includeOpts	Whether or not to include all best points reached by each algorithm
p	Existing plot to which to add the current data

Value

A plot of ERT-values of the DataSetList

Examples

```
Plot.RT.Single_Func(subset(ds1, funcId == 1))
```

```
Plot.Stats.Glicko2_Candlestick
```

Create a candlestick plot of Glicko2-rankings

Description

Create a candlestick plot of Glicko2-rankings

Usage

```
Plot.Stats.Glicko2_Candlestick(dsList, nr_rounds = 100,  
  glicko2_rank_df = NULL, which = "by_FV", target_dt = NULL)
```

```
## S3 method for class 'DataSetList'  
Plot.Stats.Glicko2_Candlestick(dsList,  
  nr_rounds = 100, glicko2_rank_df = NULL, which = "by_FV",  
  target_dt = NULL)
```

Arguments

dsList	A DataSetList
nr_rounds	The number of rounds in the tournament
glicko2_rank_df	Optional. Dataframe containing the glicko2 rating to avoid needless recalculation.
which	Whether to use fixed-target ('by_FV') or fixed-budget ('by_RT') perspective
target_dt	Optional: data table containing the targets for each function and dimension

Examples

```
Plot.Stats.Glicko2_Candlestick(ds1, nr_rounds=2)
```

Plot.Stats.Significance_Graph

Plot a network graph showing the statistically different algorithms

Description

Plot a network graph showing the statistically different algorithms

Usage

```
Plot.Stats.Significance_Graph(dsList, ftarget, alpha = 0.01,  
  bootstrap.size = 30, which = "by_FV")
```

```
## S3 method for class 'DataSetList'  
Plot.Stats.Significance_Graph(dsList, ftarget,  
  alpha = 0.01, bootstrap.size = 30, which = "by_FV")
```

Arguments

dsList	A DataSetList (should consist of only one function and dimension).
ftarget	The target function value to use
alpha	The cutoff for statistical significance
bootstrap.size	The amount of bootstrapped samples used
which	Whether to use fixed-target ('by_FV') or fixed-budget ('by_RT') perspective

Value

A graph showing the statistical significance between algorithms

Examples

```
Plot.Stats.Significance_Graph(subset(dsl, funcId == 2), 16)
```

Plot.Stats.Significance_Heatmap

Plot a heatmap showing the statistically different algorithms

Description

Plot a heatmap showing the statistically different algorithms

Usage

```
Plot.Stats.Significance_Heatmap(dsList, ftarget, alpha = 0.01,
  bootstrap.size = 30, which = "by_FV")

## S3 method for class 'DataSetList'
Plot.Stats.Significance_Heatmap(dsList, ftarget,
  alpha = 0.01, bootstrap.size = 30, which = "by_FV")
```

Arguments

dsList	A DataSetList (should consist of only one function and dimension).
ftarget	The target function value to use
alpha	The cutoff for statistical significance
bootstrap.size	The amount of bootstrapped samples used
which	Whether to use fixed-target ('by_FV') or fixed-budget ('by_RT') perspective

Value

A heatmap showing the statistical significance between algorithms

Examples

```
Plot.Stats.Significance_Heatmap(subset(dsl, funcId == 2), 16)
```

plot_general_data	<i>General function for plotting within IOHanalyzer</i>
-------------------	---

Description

General function for plotting within IOHanalyzer

Usage

```
plot_general_data(df, x_attr = "algId", y_attr = "vals",
  type = "violin", legend_attr = "algId", scale.xlog = F,
  scale.ylog = F, scale.reverse = F, p = NULL, x_title = NULL,
  y_title = NULL, plot_title = NULL, upper_attr = NULL,
  lower_attr = NULL, subplot_attr = NULL, show.legend = F,
  inf.action = "none", ...)
```

Arguments

df	The dataframe containing the data to plot. It should contain at least two columns: 'x_attr' and 'y_attr'
x_attr	The column to specify the x-axis. Default is 'algId'
y_attr	The column to specify the y-axis
type	The type of plot to use. Currently available: 'violin', 'line', 'radar', 'hist' and 'ribbon'
legend_attr	Default is 'algId' This is also used for the selection of colorschemes
scale.xlog	Logarithmic scaling of x-axis
scale.ylog	Logarithmic scaling of y-axis
scale.reverse	Decreasing or increasing x-axis
p	A previously existing plot on which to add traces. If NULL, a new canvas is created
x_title	Title of x-axis. Defaults to x_attr
y_title	Title of x-axis. Defaults to x_attr
plot_title	Title of x-axis. Defaults to no title
upper_attr	When using ribbon-plot, this can be used to create a shaded area. Only works in combination with 'lower_attr' and 'type' == 'ribbon'
lower_attr	When using ribbon-plot, this can be used to create a shaded area. Only works in combination with 'upper_attr' and 'type' == 'ribbon'
subplot_attr	Which attribute of the dataframe to use for creating subplots
show.legend	Whether or not to include a legend
inf.action	How to deal with infinite values. Can be 'none', 'overlap' or 'jitter'
...	Additional parameters for the add_trace function

print.DataSet

S3 generic print operator for DataSet

Description

S3 generic print operator for DataSet

Usage

```
## S3 method for class 'DataSet'
print(x, ...)
```

Arguments

x	A DataSet object
...	Arguments passed to other methods

Value

A short description of the DataSet

Examples

```
print(dsl[[1]])
```

`print.DataSetList` *S3 print function for DataSetList*

Description

S3 print function for DataSetList

Usage

```
## S3 method for class 'DataSetList'  
print(x, ...)
```

Arguments

`x` The DataSetList to print
`...` Arguments for underlying print function?

Examples

```
print(dsl)
```

`read_index_file` *Read .info files and extract information*

Description

Read .info files and extract information

Usage

```
read_index_file(fname)
```

Arguments

`fname` The path to the .info file

Value

The data contained in the .info file

Examples

```
path <- system.file("extdata", "ONE_PLUS_LAMDA_EA", package="IOHanalyzer")
info <- read_index_file(file.path(path, "IOHprofiler_f1_i1.info"))
```

runServer

Create a shiny-server GUI to interactively use the IOHanalyzer

Description

Create a shiny-server GUI to interactively use the IOHanalyzer

Usage

```
runServer(port = getOption("shiny.port"))
```

Arguments

port Optional; which port the server should be opened at

Examples

```
## Not run:
runServer()

## End(Not run)
```

save_plotly

Save plotly figure in multiple format

Description

NOTE: This function requires orca to be installed

Usage

```
save_plotly(p, file, width = NULL, height = NULL, ...)
```

Arguments

p plotly object. The plot to be saved
file String. The name of the figure file, with the extension of the required file-format
width Optional. Width of the figure
height Optional. Height of the figure
... Additional arguments for orca

Examples

```
## Not run:
p <- Plot.RT.Single_Func(dsl[1])
save_plotly(p, 'example_file.png')

## End(Not run)
```

scan_index_file	<i>Scan *.info files for IOHProfiler or COCO</i>
-----------------	--

Description

Scan *.info files for IOHProfiler or COCO

Usage

```
scan_index_file(folder)
```

Arguments

folder The folder containing the .info files

Value

The paths to all found .info-files

Examples

```
path <- system.file("extdata", "ONE_PLUS_LAMDA_EA", package="IOHanalyzer")
scan_index_file(path)
```

seq_FV	<i>Function for generating sequences of function values</i>
--------	---

Description

Function for generating sequences of function values

Usage

```
seq_FV(FV, from = NULL, to = NULL, by = NULL, length.out = NULL,
       scale = NULL)
```

Arguments

FV	A list of function values
from	Starting function value. Will be replaced by min(FV) if it is NULL or too small
to	Stopping function value. Will be replaced by max(FV) if it is NULL or too large
by	Stepsize of the sequence. Will be replaced if it is too small
length.out	Number of values in the sequence. 'by' takes preference if both it and length.out are provided.
scale	Scaling of the sequence. Can be either 'linear' or 'log', indicating a linear or log-linear spacing respectively. If NULL, the scale will be predicted based on FV

Value

A sequence of function values

Examples

```
FVall <- get_runtimes(dsl)
seq_FV(FVall, 10, 16, 1, scale='linear')
```

seq_RT

Function for generating sequences of runtime values

Description

Function for generating sequences of runtime values

Usage

```
seq_RT(RT, from = NULL, to = NULL, by = NULL, length.out = NULL,
       scale = "linear")
```

Arguments

RT	A list of runtime values
from	Starting runtime value. Will be replaced by min(RT) if it is NULL or too small
to	Stopping runtime value. Will be replaced by max(RT) if it is NULL or too large
by	Stepsize of the sequence. Will be replaced if it is too small
length.out	Number of values in the sequence. 'by' takes preference if both it and length.out are provided.
scale	Scaling of the sequence. Can be either 'linear' or 'log', indicating a linear or log-linear spacing respectively.

Value

A sequence of runtime values

Examples

```
RTall <- get_runtimes(dsl)
seq_RT(RTall, 0, 500, length.out=10, scale='log')
```

set_color_scheme	<i>Set the colorScheme of the IOHanalyzer plots</i>
------------------	---

Description

Set the colorScheme of the IOHanalyzer plots

Usage

```
set_color_scheme(schemename, algnames, path = NULL)
```

Arguments

schemename	Three default colorschemes are implemented: <ul style="list-style-type: none"> • Default • Variant 1 • Variant 2 • Variant 3 <p>And it is also possible to select "Custom", which allows uploading of a custom set of colors</p>
algnames	The names of the algorithms for which to set the colors
path	The path to the file containing the colors to use. Only used if schemename is "Custom"

Examples

```
set_color_scheme("Default", get_algId(dsl))
```

SP *Estimator 'SP' for the Expected Running Time (ERT)*

Description

Estimator 'SP' for the Expected Running Time (ERT)

Usage

```
SP(data, max_runtime)
```

Arguments

data	A dataframe or matrix. Each row stores the runtime sample points from several runs
max_runtime	A Numerical vector. Should have the same size as columns of data

Value

A list containing ERTs, number of succesfull runs and the succes rate

Examples

```
SP(ds1[[1]]$RT, max(ds1[[1]]$RT))
```

subset.DataSetList *Filter a DataSetList by some criteria*

Description

Filter a DataSetList by some criteria

Usage

```
## S3 method for class 'DataSetList'
subset(x, ...)
```

Arguments

x	The DataSetLsit
...	The condition to filter on. Can be any expression which assigns True or False to a DataSet object, such as DIM == 625 or funcId == 2

Value

The filtered DataSetList

Examples

```
subset(dsl, funcId == 1)
```

summary.DataSet	<i>S3 generic summary operator for DataSet</i>
-----------------	--

Description

S3 generic summary operator for DataSet

Usage

```
## S3 method for class 'DataSet'
summary(object, ...)
```

Arguments

object	A DataSet object
...	Arguments passed to other methods

Value

A summary of the DataSet containing both function-value and runtime based statistics.

Examples

```
summary(dsl[[1]])
```

summary.DataSetList	<i>S3 summary function for DataSetList</i>
---------------------	--

Description

Prints the Function ID, Dimension, Algorithm Id, datafile location and comment for every DataSet in the DataSetList

Usage

```
## S3 method for class 'DataSetList'
summary(object, ...)
```

Arguments

object	The DataSetList to print
...	Arguments for underlying summary function?

Examples

```
summary(ds1)
```

[.DataSetList] *S3 extraction function for DataSetList*

Description

S3 extraction function for DataSetList

Usage

```
## S3 method for class 'DataSetList'  
x[i, drop = FALSE]
```

Arguments

x	The DataSetList to use
i	The indices to extract
drop	Currently unused parameter

Value

The DataSetList of the DataSets at indices i of DataSetList x

Examples

```
ds1[c(1, 3)]
```


Index

*Topic **datasets**

- dsl, 11
- dsl_large, 11
- ==.DataSet, 4
- [.DataSetList, 64

- arrange, 4
- as.character.DataSet, 5
- AUC, 5

- bootstrap_RT, 6

- c.DataSet, 7
- c.DataSetList, 7
- cat.DataSet, 8
- check_format, 8
- clean_DataSetList, 9

- DataSet, 9, 33
- DataSetList, 10, 33
- dsl, 11
- dsl_large, 11

- ECDF, 12

- generate_data.Aggr, 12
- generate_data.AUC, 13
- generate_data.ECDF, 14
- generate_data.hist, 14
- generate_data.Parameters, 15
- generate_data.PMF, 15
- generate_data.Single_Function, 16
- get_algId, 16
- get_color_scheme, 17
- get_color_scheme_dt, 17
- get_default_ECDF_targets, 18
- get_dim, 18
- get_ECDF_targets, 19
- get_ERT, 19
- get_funcId, 20
- get_funvals, 21

- get_FV_overview, 21
- get_FV_sample, 22
- get_FV_summary, 23
- get_line_style, 23
- get_maxRT, 24
- get_overview, 25
- get_PAR_name, 26
- get_PAR_sample, 27
- get_PAR_summary, 28
- get_parId, 25
- get_RT_overview, 29
- get_RT_sample, 29
- get_RT_summary, 30
- get_runtimes, 31
- get_target_dt, 32
- glicko2_ranking, 32

- IOH_plot_ly_default, 34
- IOHanalyzer, 33
- IOHanalyzer-package (IOHanalyzer), 33

- limit.data, 34

- max_ERTs, 35
- mean_FVs, 35

- pairwise.test, 36
- Plot.FV.Aggregated, 37
- Plot.FV.ECDF_AUC, 38
- Plot.FV.ECDF_Per_Target, 38
- Plot.FV.ECDF_Single_Func, 39
- Plot.FV.Histogram, 40
- Plot.FV.Multi_Func, 41
- Plot.FV.Parameters, 42
- Plot.FV.PDF, 43
- Plot.FV.Single_Func, 43
- Plot.RT.Aggregated, 44
- Plot.RT.ECDF_AUC, 45
- Plot.RT.ECDF_Multi_Func, 46
- Plot.RT.ECDF_Per_Target, 47

Plot.RT.ECDF_Single_Func, [48](#)
Plot.RT.Histogram, [49](#)
Plot.RT.Multi_Func, [49](#)
Plot.RT.Parameters, [50](#)
Plot.RT.PMF, [51](#)
Plot.RT.Single_Func, [52](#)
Plot.Stats.Glicko2_Candlestick, [53](#)
Plot.Stats.Significance_Graph, [54](#)
Plot.Stats.Significance_Heatmap, [54](#)
plot_general_data, [55](#)
print.DataSet, [56](#)
print.DataSetList, [57](#)

read_index_file, [57](#)
runServer, [58](#)

save_plotly, [58](#)
scan_index_file, [59](#)
seq_FV, [59](#)
seq_RT, [60](#)
set_color_scheme, [61](#)
SP, [62](#)
subset.DataSetList, [62](#)
summary.DataSet, [63](#)
summary.DataSetList, [63](#)