Package 'IOHanalyzer'

May 29, 2019

Type Package

Title Data Analysis Part of 'IOHprofiler'

Version 0.1.1

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, withr, shinydashboard, markdown, reshape2, shinyjs, colourpicker, bsplus, DT

LinkingTo Rcpp

SystemRequirements C++11

RoxygenNote 6.1.1

Suggests testthat

Depends R (>= 2.10)

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-05-29 13:40:14 UTC

R topics documented:

`==.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 equal precision and comments; False otherwise

Examples

```
dsl[[1]] == dsl[[2]]
```

`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 Dsitribution Function)*

Description

Area Under Curve (Empirical Cumulative Dsitribution 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(dsl,c(12,14))
AUC(ecdf, 0, 100)
```

---

c.DataSetList *S3 concatenation function for DataSetList*

Description

S3 concatenation function for DataSetList

Usage

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

Arguments

...	The DataSetLists to concatenate

Value

A new DataSetList

Examples

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

cat.DataSet *S3 generic cat operator for DataSet*

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(dsl[[1]])
```

check_format *Check the format of data*

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' or 'IOHprofiler'.

Examples

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

DataSet *Constructor of S3 class 'DataSet'*

Description

DataSet contains the following attributes * funId * DIM * algId * Precision * datafile * instance * maxEvals * finalFunEvals * comment

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_IndexFile(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 = TRUE, 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: • 'IOHProfiler' • 'COCO' • 'TWO_COL' • 'COCO_BIOBJ' • 'NEVERGRAD' 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 *Example DataSetList used in tests / examples*

Description

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

Usage

```
dsl
```

Format

DataSetList

Examples

```
summary(dsl)
```

ECDF *Empirical Cumulative Dsitribution Function of Runtime of a single data set*

Description

Empirical Cumulative Dsitribution 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(dsl,c(12,14))
ECDF(dsl[[1]],c(12,14))
```

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 DataSetLsit

Value

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

Examples

```
get_algId(dsl)
```

---

get_color_scheme *Get colors according to the current colorScheme of the IOHanalyzer*

Description

Get colors according to the current colorScheme of the IOHanalyzer

Usage

```
get_color_scheme(n)
```

Arguments

n	Number of colors to get

Examples

```
get_color_scheme(5)
```

---

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

data	A DataSetList
format_func	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

dsList	The DataSetLsit

Value

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

Examples

```
get_dim(dsl)
```

---

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 *Get all function ids present in a DataSetList*

Description

Get all function ids present in a DataSetList

Usage

```
get_funcId(dsList)
```

Arguments

dsList	The DataSetLsit

Value

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

Examples

```
get_funcId(dsl)
```

get_funvals *Get all function values present in a DataSetList*

Description

Get all function values present in a DataSetList

Usage

```
get_funvals(dsList)
```

Arguments

dsList	The DataSetLsit

Value

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

Examples

```
get_funvals(dsl)
```

get_FV_overview *Get Function Value condensed overview*

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` *Get Funtion Value Samples*

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 *Get Function Value Summary*

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_parId *Get all parameter ids present in a DataSetList*

Description

Get all parameter ids present in a DataSetList

Usage

```
get_parId(dsList)
```

Arguments

dsList	The DataSetLsit

Value

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

Examples

```
get_parId(dsl)
```

---

get_PAR_name *Get the parameter names of the algorithm*

Description

Get the parameter names of the algorithm

Usage

```
get_PAR_name(ds)

## S3 method for class 'DataSet'
get_PAR_name(ds)
```

Arguments

ds	A DataSet object

Value

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

Examples

```
get_PAR_name(dsl[[1]])
```

`get_PAR_sample` *Get Parameter Value Samples*

Description

Get Parameter Value Samples

Usage

```
get_PAR_sample(ds, ftarget, ...)

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

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

Arguments

`ds`	A DataSet or DataSetList object
`ftarget`	A Numerical vector. Function values at which parameter values are observed
`...`	Arguments passed to other methods
`parId`	A character vector. Either 'all' or the name of parameters to be retrieved
`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 *Get Parameter Value Summary*

Description

Get Parameter Value Summary

Usage

```
get_PAR_summary(ds, ftarget, ...)

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

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

Arguments

ds	A DataSet or DataSetList object
ftarget	A Numerical vector. Function values at which parameter values are observed
...	Arguments passed to other methods
parId	A character vector. Either 'all' or the name of parameters to be retrieved
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(dsl, 14)
get_PAR_summary(dsl[[1]], 14)
```

get_RT_overview *Get Runtime Value condensed overview*

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(dsl)
get_RT_overview(dsl[[1]])
```

---

get_RT_sample *Get RunTime Sample*

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 *Get RunTime Summary*

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)
```

get_runtimes *Get all runtime values present in a DataSetList*

Description

Get all runtime values present in a DataSetList

Usage

```
get_runtimes(dsList)
```

Arguments

dsList	The DataSetLsit

Value

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

Examples

```
get_runtimes(dsl)
```

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 alligning 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])


runServer()
```

`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

`title`	Title for the plot
`x.title`	X-axis label
`y.title`	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

`df`	The data to subsample
`n`	The amount of samples

Value

A smaller data.frame

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

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

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 DataSetLsit
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 fucntion-values

Examples

```
mean_FVs(dsl)
```

Plot.FV.Aggregated *Plot expected function value-based comparison over multiple functions or dimensions*

Description

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

Usage

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

## S3 method for class 'DataSetList'
Plot.FV.Aggregated(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 logaritmically
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.Aggregated(dsl)
```

Plot.FV.ECDF_AUC *Radarplot of the area under the aggregated ECDF-curve of a DataSetList.*

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

`Plot.FV.ECDF_Per_Target`

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

---

Description

Plot the empirical cumulative distriburtion 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 logaritmically
`scale.reverse`	Whether or not to reverse the x-axis (when using minimization)

Value

A plot of the empirical cumulative distriburtion as a function of the fucntion 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 distriburtion as a function of the function values of a DataSetList.

Description

Plot the aggregated empirical cumulative distriburtion 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

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
scale.xlog	Whether or not to scale the x-axis logaritmically
show.per_target	Whether or not to show the individual ECDF-curves for each runtime
scale.reverse	Whether or not to reverse the x-axis (when using minimization)

Value

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

Examples

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

Plot.FV.Histogram *Plot histograms of the function values of a DataSetList at a certain target runtime*

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

dsList	A DataSetList (should consist of only one function and dimension).
runtime	The target runtime
plot_mode	How to plot the different hisograms 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 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 logaritmically
scale.ylog	Whether or not to scale the y-axis logaritmically
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.PDF — *Plot probablity density function of the function values of a DataSetList at a certain target runtime*

Description

Plot probablity 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 logaritmically

Value

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

Examples

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

Plot.FV.Single_Func — *Plot lineplot of the expected function values of a DataSetList*

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

dsList	A DataSetList (should consist of only one function and dimension).
RTstart	The starting runtime value.
RTstop	The final runtime value.
show.CI	Whether or not to show the standard deviations
show.mean	Whether or not to show the mean runtimes
show.median	Whether or not to show the median runtimes
backend	Which plotting library to use. Can be 'plotly' or 'ggplot2'
scale.xlog	Whether or not to scale the x-axis logaritmically
scale.ylog	Whether or not to scale the y-axis logaritmically
scale.reverse	Wheter 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(dsl, funcId == 1))
```

Plot.Parameters *Plot the parameter values recorded in a DataSetList*

Description

Plot the parameter values recorded in a DataSetList

Usage

```
Plot.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.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

dsList	A DataSetList (should consist of only one function and dimension).
f_min	The starting function value.
f_max	The final function 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 logaritmically
scale.ylog	Whether or not to scale the y-axis logaritmically
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.Parameters(subset(dsl, funcId == 1))
```

Plot.RT.Aggregated *Plot ERT-based comparison over multiple functions or dimensions*

Description

Plot ERT-based comparison over multiple functions or dimensions

Usage

```
Plot.RT.Aggregated(dsList, aggr_on = "funcId", targets = NULL,
  plot_mode = "radar", use_rank = F, scale.ylog = T, maximize = T,
  erts = NULL)

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

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 logaritmically
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.

Value

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

Examples

```
Plot.RT.Aggregated(dsl)
```

Plot.RT.ECDF_AUC *Radarplot of the area under the aggregated ECDF-curve of a DataSetList.*

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

Plot.RT.ECDF_Multi_Func

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

Description

Plot the aggregated empirical cumulative distriburtion 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

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

Value

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

Examples

```
Plot.RT.ECDF_Multi_Func(dsl)
```

Plot.RT.ECDF_Per_Target

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

Description

Plot the empirical cumulative distriburtion 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

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

Value

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

Examples

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

Plot.RT.ECDF_Single_Func

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

Description

Plot the aggregated empirical cumulative distriburtion 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 logaritmically

Value

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

Examples

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

Plot.RT.Histogram *Plot histograms of the runtimes of a DataSetList at a certain target function value*

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 hisograms 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 *Plot ERT-plots for multiple functions or dimensions*

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 logaritmically
scale.ylog	Whether or not to scale the y-axis logaritmically
scale.reverse	Wheter 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.PMF` *Plot probablity mass function of the runtimes of a DataSetList at a certain target function value*

Description

Plot probablity 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

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

Value

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

Examples

```
Plot.RT.PMF(subset(dsl, 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)

## 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)
```

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 logaritmically
scale.ylog	Whether or not to scale the y-axis logaritmically
scale.reverse	Wheter or not to reverse the x-axis (when using minimization)

Value

A plot of ERT-values of the DataSetList

Examples

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

---

print.DataSet *S3 generic print operator for DataSet*

Description

S3 generic print operator for DataSet

Usage

```
## S3 method for class 'DataSet'
print(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

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

Description

Read .info files and extract information

Usage

```
read_IndexFile(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_IndexFile(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

```
runServer()
```

save_plotly *Save plotly figure in multiple format*

Description

NOTE: This function requires orca to be installed, and for pdf and eps formats inkscape is also needed.

Usage

```
save_plotly(p, file, format = "svg", ...)
```

Arguments

p	plotly object. The plot to be saved
file	String. The name of the figure file
format	String. The format of the figure: 'svg', 'pdf', 'eps', 'png' are supported
...	Additional arguments for orca

Examples

```
p <- Plot.RT.Single_Func(dsl[1])
save_plotly(p, 'example_file.png', format = 'png')
```

scan_IndexFile *Scan *.info files for IOHProfiler or COCO*

Description

Scan *.info files for IOHProfiler or COCO

Usage

```
scan_IndexFile(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_IndexFile(path)
```

---

seq_FV *Function for generating sequences of function values*

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 *Set the colorScheme of the IOHanalyzer plots*

Description

Set the colorScheme of the IOHanalyzer plots

Usage

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

Arguments

schemename	Three default colorschemes are implemented: • Default • Variant 1 • Variant 2 And it is also possible to select "Custom", which allows uploading of a custom set of colors
path	The path to the file containing the colors to use. Only used if schemename is "Custom"

Examples

```
set_color_scheme("Default")
```

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
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(dsl[[1]]$RT, max(dsl[[1]]$RT))
```

subset.DataSetList *Filter a DataSetList by some criterium*

Description

Filter a DataSetList by some criterium

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 *S3 generic summary operator for DataSet*

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` *S3 summary function for DataSetList*

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(dsl)
```

`[.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

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

Index