Package 'xegaCE'

May 11, 2025

Title Computational Experiments

Version 0.9.0.0

Description The execution of a computational experiment
has three phases, all of which are
supported: (1) The repeated execution of the trials of a treatment
of an experiment is supported by one command dispatcher
which records and summarizes the parameters of a treatment
and its repeated trials in periodic snapshots.
(2) Snapshots are merged, one data set for each treatment.
(3) For report writing, a small language is provided,
for generating descriptive statistics, hypothesis tests,
and a few graphics.

License MIT + file LICENSE

URL https://github.com/ageyerschulz/kSymmetry

Encoding UTF-8

LazyData false

RoxygenNote 7.3.2

Depends R (>= 3.5.0)

Imports grDevices,
graphics,
igraph,
xtable,
smoof,
xegaBNF,
xegaDerivationTrees,
xega

Suggests testthat (>= 3.0.0)

Contents

`beamerBoxPlot` *Produces box-and-whisker plots of treatments of an experiment (eps-file).*

Description

Produces box-and-whisker plots of treatments of an experiment (eps-file).

Usage

```
beamerBoxPlot(
  treatments,
  variable,
  title = "",
  outline = TRUE,
  notch = FALSE,
  horizontal = TRUE,
  varwidth = FALSE,
  name = "",
  inpath = ".",
  outpath = "."
)
```

Arguments

`treatments`	List of filenames of treatments.
`variable`	Name of variable.
`title`	Title of Boxplot. Default: "".
`outline`	Boolean. If FALSE, remove outliers. Default: TRUE.
`notch`	Boolean. If TRUE, notches are drawn. Default: FALSE. Nonoverlapping notches indicate different medians of treatments.
`horizontal`	Boolean. Default: TRUE. Plots are horizontal.
`varwidth`	Boolean. Default: FALSE. If TRUE, width of the plot indicates sample size.
`name`	Name of experiment. Default: "".
`inpath`	Path to treatment files of experiment. Default: ".".
`outpath`	Path to report files.

Value

Name of eps file with output.

See Also

Other beamer Elements: `beamerEndDocument()`, `beamerFigure()`, `beamerMetaTable()`, `beamerProlog()`, `beamerTable()`, `beamerTables()`

Examples

```
cat("Example: TODO.\n")
```

beamerEndDocument *End of document (Latex beamer)*

Description

End of document (Latex beamer)

Usage

```
beamerEndDocument()
```

Value

A text string with latex markup.

See Also

Other beamer Elements: beamerBoxPlot(), beamerFigure(), beamerMetaTable(), beamerProlog(), beamerTable(), beamerTables()

Examples

```
a<-beamerEndDocument()
print(a)
cat(a)
```

beamerFigure *The figure environment for a latex document.*

Description

The figure environment for a latex document.

Usage

```
beamerFigure(
  name,
  fnName,
  caption = "No caption defined",
  label = "",
  width = 0.5,
  angle = -90
)
```

Arguments

name	Name of the experiment.
fnName	File name of the eps file of the graphic.
caption	Caption of the figure.
label	Label of the figure.
width	Proportion of textwidth. Default: 0.6.
angle	Angle. Default: -90.

Value

A text string with latex markup.

See Also

Other beamer Elements: beamerBoxPlot(), beamerEndDocument(), beamerMetaTable(), beamerProlog(), beamerTable(), beamerTables()

Examples

```
a<-beamerFigure(name="Ex A", fnName="test.eps", caption="test", label="ltest",
                width=0.6, angle=-90)
print(a)
cat(a)
```

---

beamerMetaTable *The meta table environment for a latex document.*

Description

The meta table environment for a latex document.

Usage

```
beamerMetaTable(name, fnName, caption = "No caption defined", label = "")
```

Arguments

name	Name of the experiment.
fnName	File name of the exported table element.
caption	Caption of the table.
label	Label of the table.

Value

A text string with latex markup.

See Also

Other beamer Elements: beamerBoxPlot(), beamerEndDocument(), beamerFigure(), beamerProlog(), beamerTable(), beamerTables()

Examples

```
a<-beamerMetaTable(name="Ex A", fnName="test.tex", caption="test", label="ltest")
print(a)
cat(a)
```

beamerProlog *Prolog of minimal latex document for beamer class.*

Description

Contains title, author and purpose of experiment.

Usage

```
beamerProlog(
  name = "Missing",
  title = "Undefined",
  author = "Unknown",
  purpose = "Unknown",
  beamertheme = "default",
  beamercolor = "electricultramarine"
)
```

Arguments

name	Name of experiment. Default: "Missing".
title	Title of the experiment. Default: "Undefined".
author	Author of document. Default: "Unknown".
purpose	Purpose of experiment. A short description. Default: "Unknown"
beamertheme	Beamer theme. Default: "default".
beamercolor	Beamer color. Default: "electricultramarine".

Value

A text string with LaTex markup.

See Also

Other beamer Elements: beamerBoxPlot(), beamerEndDocument(), beamerFigure(), beamerMetaTable(), beamerTable(), beamerTables()

Examples

```
a<-beamerProlog(name="Experiment A.",
                title="First test", author="U.N. Owen",
                purpose="To show the impossible.\n", beamertheme="AnnArbor")
cat(a)
```

beamerSection *Sections of a latex beamer document.*

Description

Sections of a latex beamer document.

Usage

```
beamerSection(name, level = "section", clearpage = FALSE, miniframe = TRUE)
```

Arguments

name	Name of the document element.
level	Latex markup of the document element. Default: "section". Latex supports: "section", "subsection", "subsubsection".
clearpage	Boolean. Default: TRUE. Clearpage before start of document element. Forces output of all floating elements of the previous document element.
miniframe	Boolean. Default: TRUE. If FALSE, miniframes are turned off.

Value

A text string with latex markup.

See Also

Other Latex Elements: latexBoxPlot(), latexEndDocument(), latexFigure(), latexMetaTable(), latexProlog(), latexSection(), latexTable(), latexTables(), latexText()

Examples

```
beamerSection(name="Design", level="section", clearpage=FALSE)
a<-beamerSection(name="Design", level="section", clearpage=TRUE)
print(a)
cat(a)
```

beamerTable *The table environment for a latex document.*

Description

The table environment for a latex document.

Usage

```
beamerTable(
  df,
  caption = "",
  label = "",
  align = "",
  center = "center",
  sanitize = FALSE,
  tableLength = 20
)
```

Arguments

df	Dataframe or matrix.
caption	Caption of the table. Default: "".
label	Label of the table. Default: "".
align	Alignment of columns. Default: "".
center	Center table. Default: "center".
sanitize	Boolean. Default: FALSE.
tableLength	Integer. Default: 25.

Value

A list of text strings with latex markup. Each list element is a table.

See Also

Other beamer Elements: beamerBoxPlot(), beamerEndDocument(), beamerFigure(), beamerMetaTable(), beamerProlog(), beamerTables()

Examples

```
df<-data.frame(Name=c("Alice", "Bob"), Age=c(25, 27))
a<-beamerTable(df, caption="Alice and Bob", label="AB")
print(a)
```

---

beamerTableLength *Beamer table length.*

Description

Beamer table length.

Usage

```
beamerTableLength()
```

Value

Integer. Number of lines in beamer table.

See Also

Other Internal Configuration: `latexTableLength()`

Examples

```
beamerTableLength()
```

`beamerTables` *Write table(s) and meta-table(s) for a latex beamer document.*

Description

Write table(s) and meta-table(s) for a latex beamer document.

Usage

```
beamerTables(
  df,
  name,
  fnName,
  mfnName = "main",
  caption = "",
  label = "",
  align = "",
  center = "center",
  sanitize = FALSE,
  tableLength = 20,
  path = ""
)
```

Arguments

`df`	Dataframe.
`name`	Name of the experiment.
`fnName`	File name of the exported table element.
`mfnName`	File name of the exported meta table element. Default: "main".
`caption`	Caption of the table. Default: "".
`label`	Label of the table. Default: "".
`align`	Alignment of columns. Default: "".
`center`	Center table. Default: "center".
`sanitize`	Boolean. Default: FALSE.
`tableLength`	Integer. Default: 60.
`path`	Path. Default: outpath.

Value

Invisible 0.

See Also

Other beamer Elements: beamerBoxPlot(), beamerEndDocument(), beamerFigure(), beamerMetaTable(), beamerProlog(), beamerTable()

Examples

```
df<-data.frame(Name=c("Alice", "Bob"), Age=c(25, 27))
a<-latexTable(df, caption="Alice and Bob", label="AB")
print(a)
```

---

beamerText *Text.*

Description

Text.

Usage

```
beamerText(..., header = "Comment", block = TRUE)
```

Arguments

...	A list of text strings.
header	Name of the block.
block	Default: TRUE.

Value

A text string.

See Also

Other Console Elements: consoleBoxPlot(), consoleEndDocument(), consoleFigure(), consoleProlog(), consoleSection(), consoleTable(), consoleText(), pause()

Examples

```
a<-beamerText("Here", "we", "are")
print(a)
cat(a)
```

booleanGrammarK *booleanGrammarK*

Description

booleanGrammarK

Usage

```
booleanGrammarK()
```

See Also

Other xegaBNF: `variableNamesLHS()`, `writeBNF()`

buildLatexMain *Build latex main program.*

Description

`buildLatexMain()` reads all latex files starting with "main" from the directory specified by path, catenates them and writes a new latex file "<main>main.tex" to the same directory.

Usage

```
buildLatexMain(name, path = ".")
```

Arguments

`name`	Name of experiment.
`path`	Path of latex report files.

Value

List of imported files. (invisible).

consoleBoxPlot *Produces box-and-whisker plots of treatments of an experiment.*

Description

Produces box-and-whisker plots of treatments of an experiment.

Usage

```
consoleBoxPlot(
  treatments,
  variable,
  title = "",
  outline = TRUE,
  notch = FALSE,
  horizontal = TRUE,
  varwidth = FALSE,
  name = "",
  inpath = ".",
  epsfilename = NULL
)
```

Arguments

treatments	List of filenames of treatments.
variable	Name of variable.
title	Title of Boxplot. Default: "".
outline	Boolean. If FALSE, remove outliers. Default: TRUE.
notch	Boolean. If TRUE, notches are drawn. Default: FALSE. Nonoverlapping notches indicate different medians of treatments.
horizontal	Boolean. Default: TRUE. Plots are horizontal.
varwidth	Boolean. Default: FALSE. If TRUE, width of the plot indicates sample size.
name	Name of experiment. Default: "".
inpath	Path to treatment files of experiment. Default: ".".
epsfilename	File name of epsfile. Default: NULL.

Value

0 (invisible).

See Also

Other Console Elements: beamerText(), consoleEndDocument(), consoleFigure(), consoleProlog(), consoleSection(), consoleTable(), consoleText(), pause()

Examples

```
cat("Example: TODO.\n")
```

consoleEndDocument *End of document (LaTeX)*

Description

End of document (LaTeX)

Usage

```
consoleEndDocument()
```

Value

A text string (invisible).

See Also

Other Console Elements: beamerText(), consoleBoxPlot(), consoleFigure(), consoleProlog(), consoleSection(), consoleTable(), consoleText(), pause()

Examples

```
a<-consoleEndDocument()
print(a)
cat(a)
```

consoleFigure *The figure output on a console.*

Description

The figure output on a console.

Usage

```
consoleFigure(
  name = "",
  fnName = "",
  caption = "No caption defined",
  label = ""
)
```

Arguments

name	Name of the experiment.
fnName	File name of the eps file of the graphic.
caption	Caption of the figure.
label	Label of the figure.

Value

A text string (invisible).

See Also

Other Console Elements: beamerText(), consoleBoxPlot(), consoleEndDocument(), consoleProlog(), consoleSection(), consoleTable(), consoleText(), pause()

Examples

```
consoleFigure(name="Ex A", fnName="test.eps", caption="test", label="ltest")
```

---

consoleProlog *Prolog of experiment.*

Description

Shows name, title, author, and purpose of experiment.

Usage

```
consoleProlog(
  name = "Missing",
  title = "Undefined",
  author = "Unknown",
  purpose = "Unknown"
)
```

Arguments

name	Name of experiment. Default: "Missing".
title	Title of the experiment. Default: "Undefined".
author	Author of experimental report. Default: "Unknown".
purpose	Purpose of experiment. Default: "Unknown".

Value

A text string (invisible).

See Also

Other Console Elements: beamerText(), consoleBoxPlot(), consoleEndDocument(), consoleFigure(), consoleSection(), consoleTable(), consoleText(), pause()

Examples

```
consoleProlog(name="Experiment A.",
    title="First test", author="U.N. Owen",
    purpose=paste0("The experiment shows that algorithm A ",
                   "is faster than algorithm B.\n"))
```

consoleSection *Shows the section of an experiment.*

Description

Shows the section of an experiment.

Usage

```
consoleSection(name = "Missing", level = "section", clearpage = TRUE)
```

Arguments

name	Name of the document element. Default: "Missing".
level	Latex markup of the document element. Default: "section". Latex supports: "section", "subsection", "subsubsection". Used for indentation of output.
clearpage	Boolean. Default: TRUE. Ignored.

Value

A text string (invisible).

See Also

Other Console Elements: beamerText(), consoleBoxPlot(), consoleEndDocument(), consoleFigure(), consoleProlog(), consoleTable(), consoleText(), pause()

Examples

```
consoleSection(name="Design", level="section", clearpage=FALSE)
consoleSection(name="Design", level="subsection", clearpage=TRUE)
```

consoleTable *The table environment for the console.*

Description

The table environment for the console.

Usage

```
consoleTable(df, caption = "", label = "", align = "", center = "center")
```

Arguments

df	Dataframe or matrix.
caption	Caption of the table. Default: "".
label	Label of the table. Default: "".
align	Alignment of columns. Default: "".
center	Center table. Default: "center".

Value

A dataframe (invisible).

See Also

Other Console Elements: beamerText(), consoleBoxPlot(), consoleEndDocument(), consoleFigure(), consoleProlog(), consoleSection(), consoleText(), pause()

Examples

```
df<-data.frame(Name=c("Alice", "Bob"), Age=c(25, 27))
consoleTable(df, caption="Alice and Bob", label="AB")
```

---

consoleText *Text.*

Description

Text.

Usage

```
consoleText(..., header = "")
```

Arguments

...	A list of text strings.
header	Name of the block.

Value

A text string (invisible).

See Also

Other Console Elements: beamerText(), consoleBoxPlot(), consoleEndDocument(), consoleFigure(), consoleProlog(), consoleSection(), consoleTable(), pause()

Examples

```
a<-consoleText("here", "we", "are")
print(a)
cat(a)
```

designOfAllTreatments *Generate a dataframe with all parameters of all treatments of an experiment.*

Description

Generate a dataframe with all parameters of all treatments of an experiment.

Usage

```
designOfAllTreatments(treatments, inpath = ".")
```

Arguments

treatments	List of treatment names of experiments.
inpath	Path for treatment files. Default: ".".

Value

A dataframe with all parameters of all treatments.

See Also

Other Design of Experiment: designOfExperiment(), designOfTreatment(), designSplitParameter(), getTreatment(), pl2df()

Examples

```
cat("TBD!\n")
```

designOfExperiment *Design of Experiment.*

Description

Design of Experiment.

Usage

```
designOfExperiment(treatments, inpath = ".")
```

Arguments

treatments	List of treatment names of experiments.
inpath	Path for treatment files. Default: ".".

Value

A list with two dataframes

- `$common` Dataframe of parameters common to all treatments of an experiment.
- `$different` Dataframe of parameters different in the treatments of an experiment.

See Also

Other Design of Experiment: `designOfAllTreatments()`, `designOfTreatment()`, `designSplitParameter()`, `getTreatment()`, `pl2df()`

Examples

```
cat("TBD!\n")
```

`designOfTreatment` *Return a dataframe with all parameters of a treatment.*

Description

Return a dataframe with all parameters of a treatment.

Usage

```
designOfTreatment(treatmentdf)
```

Arguments

`treatmentdf`	A dataframe with the results of a treatment.

Value

A dataframe with all parameters of a treatment.

See Also

Other Design of Experiment: `designOfAllTreatments()`, `designOfExperiment()`, `designSplitParameter()`, `getTreatment()`, `pl2df()`

Examples

```
cat("TBD!\n")
```

designSplitParameter *Split dataframe in common and different parameters.*

Description

Split dataframe in common and different parameters.

Usage

```
designSplitParameter(DF)
```

Arguments

DF	Dataframe with all parameters of all treatments of an experiment.

Value

A list with two dataframes

- $common Dataframe of parameters common to all treatments of an experiment.
- $different Dataframe of parameters different in the treatments of an experiment.

See Also

Other Design of Experiment: designOfAllTreatments(), designOfExperiment(), designOfTreatment(), getTreatment(), pl2df()

Examples

```
cat("TBD!\n")
```

env5symmetry *A problem environment for the 5-symmetry function.*

Description

A problem environment for the 5-symmetry function.

Usage

```
env5symmetry
```

Format

An object of class list of length 7.

See Also

Other Problem Environment: env5symmetryNN

Examples

```
t3<-"AND(OR(AND(D2,D4),AND(NOT(D2),NOT(D4))),"
t3<-paste0(t3,"OR(AND(D1, D5), AND(NOT(D1), NOT(D5))))")
env5symmetry$f(t1)
env5symmetry$f(t3)
```

---

env5symmetryNN *A problem environment for the 5-symmetry function.*

Description

A problem environment for the 5-symmetry function.

Usage

```
env5symmetryNN
```

Format

An object of class `list` of length 16.

See Also

Other Problem Environment: `env5symmetry`

Examples

```
env5symmetryNN$f(rndParms(env5symmetryNN$nNNparms()))
```

---

envRealFunctions *Package Problem Environments for Real Functions*

Description

Problem environments with real test functions for simple genetic algorithms

Details

The `envRealFunctions` package implements for example the functions F1-F5 of Kenneth De Jong's Thesis.

Problem environments for the package `xega` must be implemented as lists of functions or function factories which return with at least the following functions:

- `<factory name>$name()` is a constant function which returns the name of the environment.
- `<factory name>$bitlength()` is a constant function which returns a vector which returns for each parameter the number of bits used for coding the parameter value.
- `<factory name>$genelength()` is a constant function which returns the number of bits of a gene.

- <factory name>$lb() is a constant function which returns the vector of lower bounds of the parameters.
- <factory name>$ub() is a constant function which returns the vector of upper bounds of the parameters.
- <factory name>$f(parm, gene, lF) specifies the function with the parameter vector parm which should be optimized. The variables gene and lF extend the interface between genetic algorithm and problem environment. Both variables are not used in this package.

The constant functions <factory name>$bitlength(), <factory name>$genelength(), <factory name>$lb(), <factory name>$ub() specify the precision of the parameters as well as the hypercube in which the optimal solution(s) are searched. These four functions provide the information needed for decoding binary genes. For real coded genes, length(<factory name>$bitlength()) indicates the number of parameters.

Three additional functions can be provided:

- <factory name>$terminate(solution) TRUE, if the solution meets an early termination condition.
- <factory name>$describe() prints a description of the function (known mathematical properties and a reference to the origin of the function).
- <factory name>$solution() specifies the global minimum/maximum solution value and the list of minimal/maximal points. The function penvValidate(penv) validates the correct implementation of the function f of the environment.

At the moment, the package contains:

1. A set of 2-D parabola functions in DelayedP.R for testing: With time delays, with random errors, and early termination.
2. De Jong's functions (F1-F5).
3. Multimodal 2-D sine waves (AGSF7d1, F6, F6inv, and F7).
4. Functions A1-A30 from Lavezzi et al. (2022).
5. A wrapper for the single-objective optimization test functions of the smoof package of Bossek (2017).

Copyright

License

MIT

URL

TBD

Installation

TBD

Author(s)

Andreas Geyer-Schulz

References

De Jong, Kenneth A. (1975): *An Analysis of the Behavior of a Class of Genetic Adaptive Systems* University of Michigan, Ann Arbor. URL: https://cs.gmu.edu/~eclab/kdj_thesis.html

Lavezzi, Giovanni and Guye, Kidus and Ciarcia, Marco (2022): Nonlinear Programming Solvers for Unconstrained and Constrained Optimization Problems: a Benchmark Analysis. arXiv, arXiv:2204.05297v1 (11 April 2022).

Bossek, Jakob (2017) smoof: Single- and Multi-Objective Optimization Test Functions. The R Journal, 9(1), 103-113.

`envXORNN231` *Generate a problem environment for a NN for the XOR problem with topology* `c(2, 3, 1)`.

Description

Generate a problem environment for a NN for the XOR problem with topology `c(2, 3, 1)`.

Usage

```
envXORNN231()
```

Value

A problem environment.

Examples

```
require(xega)
cat("\n XOR Topology: c(2, 3, 1). Activation: ReLU \n")
p<-envXORNN231()
t1<-xegaRun(penv=p, algorithm="sga", generations=100, popsize=20,
  evalmethod="Deterministic", max=FALSE, verbose=1)
printNNweights(t1$solution$phenotype, p$topology())
as<-NN(t1$solution$phenotype, p$topology(), data=xorData)
cat("The activation values for the xor data set:\n")
print(as)
cat("Fitness:", t1$solution$fitness, "Above 0.5? \n")
r3<-data.frame(as, (as>0.5), xorData[,3])
colnames(r3)<-c("Activation", "Predicted", "xor")
print(r3)
```

envXORNN2421 *Generate a problem environment for the XOR problem. Topology: c(2, 4, 2, 1)*

Description

Generate a problem environment for the XOR problem. Topology: c(2, 4, 2, 1)

Usage

```
envXORNN2421()
```

Examples

```
require(xega)
cat("\n Topology: c(2, 4, 2, 1). Activation: ReLU \n")
p3<-envXORNN2421()
t3<-xegaRun(penv=p3, max=FALSE, algorithm="sgde", popsize=30, generations=100,
   mutation="MutateGeneDE", scalefactor="Uniform", crossover="UCrossGene",
   genemap="Identity", replication="DE",
   selection="UniformP", mateselection="UniformP", accept="Best")
printNNweights(t3$solution$phenotype, p3$topology())
as<-NN(t3$solution$phenotype, p3$topology(), data=xorData)
cat("Fitness:", t3$solution$fitness, "Above 0.5? \n")
r3<-data.frame(as, (as>0.5), xorData[,3])
colnames(r3)<-c("Activation", "Predicted", "xor")
print(r3)
```

experimentAllXegaParameters *A table of all xega parameters and their values of the a treatment.*

Description

A table of all xega parameters and their values of the a treatment.

Usage

```
experimentAllXegaParameters(
  treatmentName,
  inpath = ".",
  outpath = ".",
  caption = "",
  name = "",
  type = "console"
)
```

Arguments

treatmentName	Name of treatment.
inpath	Path to treatment rds-files of experiment. Default: ".".
outpath	Path to report of experiment. Default: "."
caption	Caption of table. Default: "". Default: ".".
name	Default: "".
type	Output to console (type="console") or to latex files (type="latex" | "beamer")? Default: "console".

Value

Invisible 0.

See Also

Other Report of Experiment: experimentAnalysisOfTreatment(), experimentAppendix(), experimentBoxPlot(), experimentDesign(), experimentEffectReport(), experimentEnd(), experimentGrammarTable(), experimentHypothesisTest(), experimentIsNormal(), experimentMeanMatrix(), experimentNNTables(), experimentPlotPopStats(), experimentSection(), experimentSolutionGenotype(), experimentSolutionTabl experimentStart(), experimentStatistic(), experimentText(), experimentTreatmentParameters(), experimentTreatments(), experimentXegaParameters()

Examples

```
cat("TBD!\n")
```

---

experimentAnalysisOfTreatment

Analysis of a treatment.

Description

Analysis of a treatment.

Usage

```
experimentAnalysisOfTreatment(
  name = "",
  treatmentname = "",
  miniframe = TRUE,
  inpath = ".",
  outpath = ".",
  type = "console"
)
```

Arguments

name	Name of Experiment.
treatmentname	Name of rds-file of treatment
miniframe	Boolean. Default: TRUE. If false, turn off miniframes.
inpath	Path to treatment rds-files of experiment. Default: ".".
outpath	Path to report of experiment. Default: "."
type	Output to console (type="console") or to latex files (type="latex")? Default: "console".

Value

0 (invisible).

See Also

Other Report of Experiment: experimentAllXegaParameters(), experimentAppendix(), experimentBoxPlot(), experimentDesign(), experimentEffectReport(), experimentEnd(), experimentGrammarTable(), experimentHypothesisTest(), experimentIsNormal(), experimentMeanMatrix(), experimentNNTables(), experimentPlotPopStats(), experimentSection(), experimentSolutionGenotype(), experimentSolutionTabl experimentStart(), experimentStatistic(), experimentText(), experimentTreatmentParameters(), experimentTreatments(), experimentXegaParameters()

Examples

```
cat("TBD! \n")
```

---

experimentAppendix *Appendix of an experimental report.*

Description

Three parts:

1. Summary of descriptive statistics of the distribution of the performance variables of all treatments for n trials.
2. Analysis of each treatment.
3. All xega parameters (and their setting for one treatment).

Usage

```
experimentAppendix(
  treatments,
  name = "",
  miniframe = TRUE,
  inpath = ".",
  outpath = ",",
  type = "console"
)
```

Arguments

treatments	List of treatment file names.
name	Name of experiment. Default: "".
miniframe	Boolean. Default: TRUE. If false, turn off miniframes in treatment subsection.
inpath	Path to treatment rds-files of experiment. Default: ".".
outpath	Path to report of experiment. Default: "."
type	Output to console (type="console") or to latex files (type="latex" | "beamer")? Default: "console".

Value

0 (invisible).

See Also

Other Report of Experiment: experimentAllXegaParameters(), experimentAnalysisOfTreatment(), experimentBoxPlot(), experimentDesign(), experimentEffectReport(), experimentEnd(), experimentGrammarTable(), experimentHypothesisTest(), experimentIsNormal(), experimentMeanMatrix(), experimentNNTables(), experimentPlotPopStats(), experimentSection(), experimentSolutionGenotype(), experimentSolutionTable(), experimentStart(), experimentStatistic(), experimentText(), experimentTreatmentParameters(), experimentTreatments(), experimentXegaParameters()

Examples

```
cat("TBD!\n")
```

---

experimentBoxPlot *Produces box-and-whisker plots of treatments of an experiment.*

Description

See the documentation of base::boxplot.

Usage

```
experimentBoxPlot(
  treatments,
  variable,
  title = "",
  outline = TRUE,
  notch = FALSE,
  horizontal = TRUE,
  varwidth = FALSE,
  name = "",
  inpath = ".",
  outpath = ".",
  type = "console"
)
```

Arguments

treatments	List of filenames of treatments.
variable	Name of variable.
title	Title of Boxplot. Default: "".
outline	Boolean. If FALSE, remove outliers. Default: TRUE.
notch	Boolean. If TRUE, notches are drawn. Default: FALSE. Nonoverlapping notches indicate different medians of treatments.
horizontal	Boolean. Default: TRUE. Plots are horizontal.
varwidth	Boolean. Default: FALSE. If TRUE, width of the plot indicates sample size.
name	Name of experiment. Default: "".
inpath	Path to treatment files of experiment. Default: ".".
outpath	Path to report files.
type	Output to console (type="console") or to latex files (type="latex" | "beamer")? Default: "console".

Value

0 (invisible).

See Also

Other Report of Experiment: experimentAllXegaParameters(), experimentAnalysisOfTreatment(), experimentAppendix(), experimentDesign(), experimentEffectReport(), experimentEnd(), experimentGrammarTable(), experimentHypothesisTest(), experimentIsNormal(), experimentMeanMatrix(), experimentNNTables(), experimentPlotPopStats(), experimentSection(), experimentSolutionGenotype(), experimentSolutionTable(), experimentStart(), experimentStatistic(), experimentText(), experimentTreatmentParameters(), experimentTreatments(), experimentXegaParameters()

Examples

```
cat("Example: TODO.\n")
```

---

experimentDesign *Produces tables of common and different parameters of treatments.*

Description

The table of different parameters of treatments and the table of common parameters of treatments are traditionally called the design of the experiment. The common parameters should make the experiment replicable, The behaviorally relevant different parameters provide the basis of testing hypothesis about the effects caused by these parameters.

Usage

```
experimentDesign(
  treatments,
  inpath = ".",
  outpath = ".",
  caption = "",
  name = "",
  type = "console"
)
```

Arguments

treatments	List of treatment names of experiment.
inpath	Path to treatment rds-files of experiment. Default: ".".
outpath	Path to report of experiment. Default: "."
caption	Caption of table. Default: "". Default: ".".
name	Name of experiment. Default: "".
type	Output to console (type="console") or to latex files (type="latex" | "beamer")? Default: "console".

Value

0 (invisible).

See Also

Other Report of Experiment: experimentAllXegaParameters(), experimentAnalysisOfTreatment(), experimentAppendix(), experimentBoxPlot(), experimentEffectReport(), experimentEnd(), experimentGrammarTable(), experimentHypothesisTest(), experimentIsNormal(), experimentMeanMatrix(), experimentNNTables(), experimentPlotPopStats(), experimentSection(), experimentSolutionGenotype(), experimentSolutionTable(), experimentStart(), experimentStatistic(), experimentText(), experimentTreatmentParameters(), experimentTreatments(), experimentXegaParameters()

Examples

```
cat("TBD! \n")
```

---

experimentEffectReport

Report the effects of two different treatments-

Description

Report the effects of two different treatments-

Usage

```
experimentEffectReport(
  name = "",
  treatmentname1 = "",
  treatmentname2 = "",
  variable = "",
  test = "wilcox.test",
  mu = 0,
  alternative = "two.sided",
  alpha = 0.05,
  coef = 1.5,
  silent = TRUE,
  inpath = ".",
  outpath = ".",
  type = "console"
)
```

Arguments

name	Name of Experiment.
treatmentname1	Name of rds-file of treatment 1 (data vector x).
treatmentname2	Name of rds-file of treatment 2 (data vector y).
variable	Name of variable tested.
test	Name of test. Default: "wilcox.test". Available: "wilcox.test" or "t.test". If both data vectors are normal, "t.test" is used.
mu	Mean of difference of variables of treatments. Default: 0.
alternative	Alternative hypothesis. Default: "two.sided" • "two.sided": `0==(mean(x) - mean(y))`. • "less": `(mean(x) - mean(y))` smaller 0. • "greate": `(mean(x) - mean(y))` greater 0.
alpha	Level of significance. Default: 0.05.
coef	Scaling of window for removing outliers. Default: 1.5 (as in boxplot.stats). 0 means all data are retained. The windox size is given by the length of the box plus/minus coef times the length of the box.
silent	If FALSE, supress some output.
inpath	Path to treatment rds-files of experiment. Default: ".".
outpath	Path to report of experiment. Default: "."
type	Output to console (type="console") or to latex files (type="latex")? Default: "console".

Value

0 (invisible).

See Also

Other Report of Experiment: experimentAllXegaParameters(), experimentAnalysisOfTreatment(), experimentAppendix(), experimentBoxPlot(), experimentDesign(), experimentEnd(), experimentGrammarTab experimentHypothesisTest(), experimentIsNormal(), experimentMeanMatrix(), experimentNNTables(), experimentPlotPopStats(), experimentSection(), experimentSolutionGenotype(), experimentSolutionTabl experimentStart(), experimentStatistic(), experimentText(), experimentTreatmentParameters(), experimentTreatments(), experimentXegaParameters()

Examples

```
cat("TBD\n")
```

`experimentEnd` *End of experiment.*

Description

End of experiment.

Usage

```
experimentEnd(name = "", type = "console", outpath = ".")
```

Arguments

`name`	Name of experiment. Default: "Missing".
`type`	Output to console (type="console") or to latex files (type="latex" | "beamer")? Default: "console".
`outpath`	Path for latex output files. Default: "".

Value

0 (invisible).

See Also

Other Report of Experiment: experimentAllXegaParameters(), experimentAnalysisOfTreatment(), experimentAppendix(), experimentBoxPlot(), experimentDesign(), experimentEffectReport(), experimentGrammarTable(), experimentHypothesisTest(), experimentIsNormal(), experimentMeanMatrix(), experimentNNTables(), experimentPlotPopStats(), experimentSection(), experimentSolutionGenotype(), experimentSolutionTable(), experimentStart(), experimentStatistic(), experimentText(), experimentTreatmentParameters(), experimentTreatments(), experimentXegaParameters()

Examples

```
experimentEnd()
```

`experimentGrammarTable` *A production table of a grammar.*

Description

A production table of a grammar.

Usage

```
experimentGrammarTable(
  treatmentname = "",
  name = "",
  common = TRUE,
  inpath = ".",
  outpath = ".",
  type = "console"
)
```

Arguments

`treatmentname`	Name of rds-file of treatment.
`name`	Name of experiment. Default: "".
`common`	Boolean. Default: TRUE. Used in common parameters or in analysis if treatment part?
`inpath`	Path to treatment rds-files of experiment. Default: ".".
`outpath`	Path to report of experiment. Default: "."
`type`	Output to console (type="console") or to latex files (type="latex" | "beamer")? Default: "console".

Value

0 (invisible).

See Also

Other Report of Experiment: `experimentAllXegaParameters()`, `experimentAnalysisOfTreatment()`, `experimentAppendix()`, `experimentBoxPlot()`, `experimentDesign()`, `experimentEffectReport()`, `experimentEnd()`, `experimentHypothesisTest()`, `experimentIsNormal()`, `experimentMeanMatrix()`, `experimentNNTables()`, `experimentPlotPopStats()`, `experimentSection()`, `experimentSolutionGenotype()`, `experimentSolutionTable()`, `experimentStart()`, `experimentStatistic()`, `experimentText()`, `experimentTreatmentParameters()`, `experimentTreatments()`, `experimentXegaParameters()`

Examples

```
cat("TBD!\n")
```

`experimentHypothesisTest` *Analysis of a treatment.*

Description

Analysis of a treatment.

Usage

```
experimentHypothesisTest(
  name = "",
  treatmentname1 = "",
  treatmentname2 = "",
  variable = "",
  test = "wilcox.test",
  mu = 0,
  alternative = "two.sided",
  alpha = 0.05,
  coef = 1.5,
  inpath = ".",
  outpath = ".",
  type = "console"
)
```

Arguments

`name`	Name of Experiment.
`treatmentname1`	Name of rds-file of treatment 1 (data vector x).
`treatmentname2`	Name of rds-file of treatment 2 (data vector y).
`variable`	Name of variable tested.
`test`	Name of test. Default: "wilcox.test". Available: "wilcox.test" or "t.test".
`mu`	Mean of difference of variables of treatments. Default: 0.
`alternative`	Alternative hypothesis. Default: "two.sided" • "two.sided": `0==(mean(x) - mean(y))`. • "less": `(mean(x) - mean(y))` smaller 0. • "greate": `(mean(x) - mean(y))` greater 0.
`alpha`	Level of significance. Default: 0.05.
`coef`	Scaling of window for removing outliers. Default: 1.5 (as in boxplot.stats). 0 means all data are retained. The windox size is given by the length of the box plus/minus coef times the length of the box.
`inpath`	Path to treatment rds-files of experiment. Default: ".".
`outpath`	Path to report of experiment. Default: "."
`type`	Output to console (type="console") or to latex files (type="latex")? Default: "console".

Value

Boolean (invisible). TRUE: H0: Accepted. FALSE: H0: Rejected.

See Also

Other Report of Experiment: `experimentAllXegaParameters()`, `experimentAnalysisOfTreatment()`, `experimentAppendix()`, `experimentBoxPlot()`, `experimentDesign()`, `experimentEffectReport()`, `experimentEnd()`, `experimentGrammarTable()`, `experimentIsNormal()`, `experimentMeanMatrix()`, `experimentNNTables()`, `experimentPlotPopStats()`, `experimentSection()`, `experimentSolutionGenotype()`, `experimentSolutionTable()`, `experimentStart()`, `experimentStatistic()`, `experimentText()`, `experimentTreatmentParameters()`, `experimentTreatments()`, `experimentXegaParameters()`

Examples

```
cat("TBD\n")
```

---

experimentIsNormal *Testing for normality of a data vector.*

Description

A wrapper for the Shapiro-Wilk test. See `stat::shapiro.test`.

Usage

```
experimentIsNormal(
  treatmentname = "",
  name = "",
  variable = "",
  alpha = 0.05,
  coef = 1.5,
  silent = TRUE,
  inpath = ".",
  outpath = ".",
  type = "console"
)
```

Arguments

`treatmentname`	Name of rds-file of treatment.
`name`	Name of experiment. Default: "".
`variable`	Name of variable. Default: "".
`alpha`	Level of significance. Default: 0.05.
`coef`	Scaling of window for removing outliers. Default: 1.5 (as in boxplot.stats). 0 means all data are retained. The windox size is given by the length of the box plus/minus coef times the length of the box.
`silent`	If TRUE, no output.
`inpath`	Path to treatment rds-files of experiment. Default: ".".

outpath	Path to report of experiment. Default: "."
type	Output to console (type="console") or to latex files (type="latex" | "beamer")? Default: "console".

Value

Boolean (invisible).

See Also

Other Report of Experiment: experimentAllXegaParameters(), experimentAnalysisOfTreatment(), experimentAppendix(), experimentBoxPlot(), experimentDesign(), experimentEffectReport(), experimentEnd(), experimentGrammarTable(), experimentHypothesisTest(), experimentMeanMatrix(), experimentNNTables(), experimentPlotPopStats(), experimentSection(), experimentSolutionGenotype(), experimentSolutionTable(), experimentStart(), experimentStatistic(), experimentText(), experimentTreatmentParameters(), experimentTreatments(), experimentXegaParameters()

Examples

```
cat("TBD!\n")
```

---

experimentMeanMatrix *Produces matrix of mean vectors of statistics of experiment.*

Description

Specify either treatment name or variable name or none. Produces tables with statistical result of experiment.

Usage

```
experimentMeanMatrix(
  treatments,
  tpvec = "",
  variable = NULL,
  statistic = "mean",
  inpath = ".",
  outpath = ".",
  caption = "",
  name = "",
  type = "console"
)
```

Arguments

treatments	List of treatment names of experiment.
tpvec	Vector of treatment patterns.
variable	Name of variable. Default: NULL.
statistic	Name of statistic. Default: "mean".

inpath	Path to treatment rds-files of experiment. Default: ".".
outpath	Path to report of experiment. Default: "."
caption	Caption of table. Default: "". Default: ".".
name	Name of experiment. Default: "".
type	Output to console (type="console") or to latex files (type="latex" | "beamer")? Default: "console".

Value

0 (invisible).

See Also

Other Report of Experiment: experimentAllXegaParameters(), experimentAnalysisOfTreatment(), experimentAppendix(), experimentBoxPlot(), experimentDesign(), experimentEffectReport(), experimentEnd(), experimentGrammarTable(), experimentHypothesisTest(), experimentIsNormal(), experimentNNTables(), experimentPlotPopStats(), experimentSection(), experimentSolutionGenotype(), experimentSolutionTable(), experimentStart(), experimentStatistic(), experimentText(), experimentTreatmentParameters(), experimentTreatments(), experimentXegaParameters()

Examples

```
cat("TBD! \n")
```

---

experimentNNTables *The solution of a feedforward NN.*

Description

The solution of a feedforward NN.

Usage

```
experimentNNTables(
  treatmentname = "",
  name = "",
  inpath = ".",
  outpath = ".",
  type = "console",
  n = 1
)
```

Arguments

treatmentname	Name of rds-file of treatment.
name	Name of experiment. Default: "".
inpath	Path to treatment rds-files of experiment. Default: ".".
outpath	Path to report of experiment. Default: "."
type	Output to console (type="console") or to latex files (type="latex" | "beamer")? Default: "console".
n	Number of solutions. Default: 5.

Value

0 (invisible).

See Also

Other Report of Experiment: experimentAllXegaParameters(), experimentAnalysisOfTreatment(), experimentAppendix(), experimentBoxPlot(), experimentDesign(), experimentEffectReport(), experimentEnd(), experimentGrammarTable(), experimentHypothesisTest(), experimentIsNormal(), experimentMeanMatrix(), experimentPlotPopStats(), experimentSection(), experimentSolutionGenotype(), experimentSolutionTable(), experimentStart(), experimentStatistic(), experimentText(), experimentTreatmentParameters(), experimentTreatments(), experimentXegaParameters()

Examples

```
cat("TBD!\n")
```

---

experimentPlotPopStats

Produces a plot of population statistics of xegaRun.

Description

See the documentation xegaPlotPopStats.

Usage

```
experimentPlotPopStats(
  treatmentname,
  name = "",
  inpath = ".",
  outpath = ".",
  type = "console"
)
```

Arguments

treatmentname	Name of rds-file of treatment.
name	Name of experiment. Default: "".
inpath	Path to treatment files of experiment. Default: ".".
outpath	Path to report files.
type	Output to console (type="console") or to latex files (type="latex" | "beamer")? Default: "console".

Value

0 (invisible).

See Also

Other Report of Experiment: experimentAllXegaParameters(), experimentAnalysisOfTreatment(), experimentAppendix(), experimentBoxPlot(), experimentDesign(), experimentEffectReport(), experimentEnd(), experimentGrammarTable(), experimentHypothesisTest(), experimentIsNormal(), experimentMeanMatrix(), experimentNNTables(), experimentSection(), experimentSolutionGenotype(), experimentSolutionTable(), experimentStart(), experimentStatistic(), experimentText(), experimentTreatmentParameters(), experimentTreatments(), experimentXegaParameters()

Examples

```
cat("Example: TODO.\n")
```

`experimentSection` *Sections of an experiment*

Description

Sections of an experiment

Usage

```
experimentSection(
  name = "Missing experiment",
  secname = "Missing section",
  level = "section",
  clearpage = TRUE,
  type = "console",
  miniframe = TRUE,
  outpath = "."
)
```

Arguments

`name`	Name of experiment. (
`secname`	Name of section.
`level`	Latex markup of the document element. Default: "section". Latex supports: "section", "subsection", "subsubsection". Used for indentation of output.
`clearpage`	Boolean. Default: TRUE. Ignored.
`type`	Output to console (type="console") or to latex files (type="latex" | "beamer")? Default: "console".
`miniframe`	Boolean. Default: TRUE. If false, turns of miniframes for this section.
`outpath`	Path for latex output files. Default: "".

Value

0 (invisible).

See Also

Other Report of Experiment: experimentAllXegaParameters(), experimentAnalysisOfTreatment(), experimentAppendix(), experimentBoxPlot(), experimentDesign(), experimentEffectReport(), experimentEnd(), experimentGrammarTable(), experimentHypothesisTest(), experimentIsNormal(), experimentMeanMatrix(), experimentNNTables(), experimentPlotPopStats(), experimentSolutionGenotype(), experimentSolutionTable(), experimentStart(), experimentStatistic(), experimentText(), experimentTreatmentParameters(), experimentTreatments(), experimentXegaParameters()

Examples

```
experimentSection(name="Experiment A",
                  secname="Design", level="section", clearpage=FALSE)
```

---

experimentSolutionGenotype

Plots the derivation tree of a solution.

Description

For algorithm=="sgp", the genotype of a solution is a complete derivation tree.

Usage

```
experimentSolutionGenotype(
  treatmentname,
  name = "",
  inpath = ".",
  outpath = ".",
  type = "console"
)
```

Arguments

treatmentname	Name of rds-file of treatment.
name	Name of experiment. Default: "".
inpath	Path to treatment files of experiment. Default: ".".
outpath	Path to report files.
type	Output to console (type="console") or to latex files (type="latex" | "beamer")? Default: "console".

Value

0 (invisible).

See Also

Other Report of Experiment: experimentAllXegaParameters(), experimentAnalysisOfTreatment(), experimentAppendix(), experimentBoxPlot(), experimentDesign(), experimentEffectReport(), experimentEnd(), experimentGrammarTable(), experimentHypothesisTest(), experimentIsNormal(), experimentMeanMatrix(), experimentNNTables(), experimentPlotPopStats(), experimentSection(), experimentSolutionTable(), experimentStart(), experimentStatistic(), experimentText(), experimentTreatmentParameters(), experimentTreatments(), experimentXegaParameters()

Examples

```
cat("Example: TODO.\n")
```

---

experimentSolutionTable

A solution table.

Description

A solution table.

Usage

```
experimentSolutionTable(
  treatmentname = "",
  name = "",
  inpath = ".",
  outpath = ".",
  type = "console",
  n = 5
)
```

Arguments

treatmentname	Name of rds-file of treatment.
name	Name of experiment. Default: "".
inpath	Path to treatment rds-files of experiment. Default: ".".
outpath	Path to report of experiment. Default: "."
type	Output to console (type="console") or to latex files (type="latex" | "beamer")? Default: "console".
n	Number of solutions. Default: 5.

Value

0 (invisible).

See Also

Other Report of Experiment: experimentAllXegaParameters(), experimentAnalysisOfTreatment(), experimentAppendix(), experimentBoxPlot(), experimentDesign(), experimentEffectReport(), experimentEnd(), experimentGrammarTable(), experimentHypothesisTest(), experimentIsNormal(), experimentMeanMatrix(), experimentNNTables(), experimentPlotPopStats(), experimentSection(), experimentSolutionGenotype(), experimentStart(), experimentStatistic(), experimentText(), experimentTreatmentParameters(), experimentTreatments(), experimentXegaParameters()

Examples

```
cat("TBD!\n")
```

`experimentStart` *Start of experiment.*

Description

Start of experiment.

Usage

```
experimentStart(
  name = "Missing",
  title = "Undefined",
  author = "Unknown",
  purpose = "Unkown",
  beamertheme = "Berlin",
  beamercolor = "electricultramarine",
  type = "console",
  outpath = "."
)
```

Arguments

`name`	Name of experiment. Default: "Missing".
`title`	Title of the experiment. Default: "Undefined".
`author`	Author of document. Default: "Unknown".
`purpose`	Purpose of experiment. Default: "Unknown".
`beamertheme`	Beamer layout. Default: "Berlin". Choose from "default", "AnnArbor", "Antibes", "Bergen", "Berkeley", "Berlin", "Boadilla", "CambridgeUS", "Copenhagen", "Darmstadt", "Dresden", "Frankfurt", "Goettingen", "Hannover", "Ilmenau", "JuanLesPins", "Luebeck", "Madrid", "Malmoe", "Marburg", "Montpellier", "PaloAlto", "Pittsburgh", "Rochester", "Singapore", "Szeged", "Warsaw".
`beamercolor`	Color of beamer structure elements. Setting `beamercolor="default"` uses the standard color scheme of the beamer theme. Default: "electricultramarine". • Yellow(s): "goldenyellow", "electricyellow", "icterine", "flavescent", "lemon". • Orange(s): "amber", "cadmiumorange", "internationalorange".

- Red(s): "ferrarired", "fireenginered", "cadmiumred".
- Blue(s): "ao", "babyblueeyes", "bleudefrance", "blue", "cobalt", "darkmidnightblue", "brandeisblue", "deepskyblue", "iris", "navyblue", "ultramarine", "electricultramarine".
- Green(s): "darkpastelgreen", "cadmiumgreen".

`type` Output to console (type="console") or to latex files (type="latex" | "beamer")? Default: "console".

`outpath` Path for latex output files. Default: "".

Value

0 (invisible).

See Also

Other Report of Experiment: `experimentAllXegaParameters()`, `experimentAnalysisOfTreatment()`, `experimentAppendix()`, `experimentBoxPlot()`, `experimentDesign()`, `experimentEffectReport()`, `experimentEnd()`, `experimentGrammarTable()`, `experimentHypothesisTest()`, `experimentIsNormal()`, `experimentMeanMatrix()`, `experimentNNTables()`, `experimentPlotPopStats()`, `experimentSection()`, `experimentSolutionGenotype()`, `experimentSolutionTable()`, `experimentStatistic()`, `experimentText()`, `experimentTreatmentParameters()`, `experimentTreatments()`, `experimentXegaParameters()`

Examples

```
experimentStart(name="Experiment A.", title="First test", author="U.N. Owen")
```

---

experimentStatistic *Produces tables of statistics of experiment.*

Description

Specify either treatment name or variable name or none. Produces tables with statistical result of experiment.

Usage

```
experimentStatistic(
  treatments,
  treatment = NULL,
  variable = NULL,
  inpath = ".",
  outpath = ".",
  caption = "",
  name = "",
  type = "console"
)
```

Arguments

treatments	List of treatment names of experiment.
treatment	Name of treatment. Default: NULL.
variable	Name of variable. Default: NULL.
inpath	Path to treatment rds-files of experiment. Default: ".".
outpath	Path to report of experiment. Default: "."
caption	Caption of table. Default: "". Default: ".".
name	Name of experiment. Default: "".
type	Output to console (type="console") or to latex files (type="latex" | "beamer")? Default: "console".

Value

0 (invisible).

See Also

Other Report of Experiment: experimentAllXegaParameters(), experimentAnalysisOfTreatment(), experimentAppendix(), experimentBoxPlot(), experimentDesign(), experimentEffectReport(), experimentEnd(), experimentGrammarTable(), experimentHypothesisTest(), experimentIsNormal(), experimentMeanMatrix(), experimentNNTables(), experimentPlotPopStats(), experimentSection(), experimentSolutionGenotype(), experimentSolutionTable(), experimentStart(), experimentText(), experimentTreatmentParameters(), experimentTreatments(), experimentXegaParameters()

Examples

```
cat("TBD! \n")
```

---

experimentText *Description (a block of text).*

Description

Description (a block of text).

Usage

```
experimentText(
  ...,
  header = "",
  name = "",
  block = TRUE,
  type = "console",
  outpath = "."
)
```

Arguments

...	A list of comma separated strings.
header	Name of the block.
name	Default: "".
block	Default: TRUE: frame(block(text)) else FALSE: frame(text).
type	Output to console (type="console") or to latex files (type="latex" | "beamer")? Default: "console".
outpath	Path for latex output files. Default: "".

Value

0 (invisible).

See Also

Other Report of Experiment: experimentAllXegaParameters(), experimentAnalysisOfTreatment(), experimentAppendix(), experimentBoxPlot(), experimentDesign(), experimentEffectReport(), experimentEnd(), experimentGrammarTable(), experimentHypothesisTest(), experimentIsNormal(), experimentMeanMatrix(), experimentNNTables(), experimentPlotPopStats(), experimentSection(), experimentSolutionGenotype(), experimentSolutionTable(), experimentStart(), experimentStatistic(), experimentTreatmentParameters(), experimentTreatments(), experimentXegaParameters()

Examples

```
experimentText(
"This text describes my experiment.",
"However, it is really neat.")
```

---

experimentTreatmentParameters

A table of the parameters and their values of a treatment.

Description

A table of the parameters and their values of a treatment.

Usage

```
experimentTreatmentParameters(
  treatmentName,
  inpath = ".",
  outpath = ".",
  caption = "",
  name = "",
  type = "console"
)
```

Arguments

treatmentName	Name of treatment.
inpath	Path to treatment rds-files of experiment. Default: ".".
outpath	Path to report of experiment. Default: "."
caption	Caption of table. Default: "". Default: ".".
name	Default: "".
type	Output to console (type="console") or to latex files (type="latex" | "beamer")? Default: "console".

Value

Invisible 0.

See Also

Other Report of Experiment: experimentAllXegaParameters(), experimentAnalysisOfTreatment(), experimentAppendix(), experimentBoxPlot(), experimentDesign(), experimentEffectReport(), experimentEnd(), experimentGrammarTable(), experimentHypothesisTest(), experimentIsNormal(), experimentMeanMatrix(), experimentNNTables(), experimentPlotPopStats(), experimentSection(), experimentSolutionGenotype(), experimentSolutionTable(), experimentStart(), experimentStatistic(), experimentText(), experimentTreatments(), experimentXegaParameters()

Examples

```
cat("TBD!\n")
```

---

experimentTreatments *Get list of files with treatment results.*

Description

Get list of files with treatment results.

Usage

```
experimentTreatments(pattern = "*\\.rds", inpath = ".")
```

Arguments

pattern	A regular expression for selecting filenames of treatments. Assumption: One rds file per treatment. Default: "*\.rds".
inpath	Path for treatment files. Default: ".".

Value

List of filenames with treatments.

See Also

Other Report of Experiment: experimentAllXegaParameters(), experimentAnalysisOfTreatment(), experimentAppendix(), experimentBoxPlot(), experimentDesign(), experimentEffectReport(), experimentEnd(), experimentGrammarTable(), experimentHypothesisTest(), experimentIsNormal(), experimentMeanMatrix(), experimentNNTables(), experimentPlotPopStats(), experimentSection(), experimentSolutionGenotype(), experimentSolutionTable(), experimentStart(), experimentStatistic(), experimentText(), experimentTreatmentParameters(), experimentXegaParameters()

Examples

```
experimentTreatments()
```

---

experimentXegaParameters

A table of the xega parameters and their values of a treatment.

Description

A table of the xega parameters and their values of a treatment.

Usage

```
experimentXegaParameters(
  treatmentName,
  inpath = ".",
  outpath = ".",
  caption = "",
  name = "",
  type = "console"
)
```

Arguments

treatmentName	Name of treatment.
inpath	Path to treatment rds-files of experiment. Default: ".".
outpath	Path to report of experiment. Default: "."
caption	Caption of table. Default: "". Default: ".".
name	Default: "".
type	Output to console (type="console") or to latex files (type="latex" | "beamer")? Default: "console".

Value

Invisible 0.

See Also

Other Report of Experiment: experimentAllXegaParameters(), experimentAnalysisOfTreatment(), experimentAppendix(), experimentBoxPlot(), experimentDesign(), experimentEffectReport(), experimentEnd(), experimentGrammarTable(), experimentHypothesisTest(), experimentIsNormal(), experimentMeanMatrix(), experimentNNTables(), experimentPlotPopStats(), experimentSection(), experimentSolutionGenotype(), experimentSolutionTable(), experimentStart(), experimentStatistic(), experimentText(), experimentTreatmentParameters(), experimentTreatments()

Examples

```
cat("TBD!\n")
```

---

fullFileNames *A list of paths to treatment rds-files.*

Description

A list of paths to treatment rds-files.

Usage

```
fullFileNames(treatments, inpath = ".")
```

Arguments

treatments	List of treatment names of experiments.
inpath	Path for treatment files. Default: ".".

Value

A list of paths to treatment rds-files.

See Also

Other File I/O: newFileName(), writeText()

Examples

```
cat("TBD!\n")
```

getTreatment *Get treatment.*

Description

Get treatment.

Usage

```
getTreatment(treatmentname = "", inpath = ".")
```

Arguments

treatmentname	Name of the rds-file of a treatment.
inpath	Path for treatment files. Default: ".".

Value

A treatment.

See Also

Other Design of Experiment: designOfAllTreatments(), designOfExperiment(), designOfTreatment(), designSplitParameter(), pl2df()

Examples

```
cat("TBD!\n")
```

Identity *Identity function*

Description

Identity function

Usage

```
Identity(x)
```

Arguments

x	A number.

Value

x.

iterationReporter — *Reporter of mean treatment performance data (fitness, generations) after k trials.*

Description

If `!(verbose==0)` print data on console.

Usage

```
iterationReporter(
  experimentName,
  treatmentName,
  trials,
  i,
  resultDF,
  verbose = 0
)
```

Arguments

`experimentName`	Name of experiment.
`treatmentName`	Name of treatement.
`trials`	Number of trails.
`i`	Number of iteration.
`resultDF`	Data frame of results.
`verbose`	Progress report. If not `0`, output performance data to console. Default: 0 - Not output to console.

Value

Invisible zero.

See Also

Other Wrapper (Internal): `treatmentReporter()`

kSym — *Wrapper for xegaRun for k-symmetry experiments.*

Description

Tentative. Restricts the parameter space of xegaRun. Not yet flexible at all.

Usage

```
kSym(
  expReplay = 0,
  treatmentname = "t0",
  experimentname = "e0",
  replay = 0,
  k = 2,
  trials = 1,
  tVerbose = 1,
  grammarfn = "AndOrNot.txt",
  algorithm = "sgp",
  popsize = 200,
  generations = 5e+05,
  initgene = "InitGene",
  selection = "SUS",
  mateselection = "SUS",
  replication = "Kid2",
  crossover = "Cross2Gene",
  mutation = "MutateGene",
  accept = "All",
  reportEvalErrors = "FALSE",
  crossrate = 0.2,
  mutrate = 0.4,
  ivmutrate = "Const",
  mutrate2 = 0.8,
  ivcrossrate = "Const",
  crossrate2 = 0.4,
  scalefactor = "Uniform",
  executionModel = "Sequential",
  verbose = 0,
  Gpath = kSymmetryGrammarPath,
  outpath = ".",
  semantics = "byValue",
  batch = FALSE,
  everyk = 10
)
```

Arguments

`expReplay`	Integer. If `expReplay>0`, then use `expReplay` as the seed of the random number generator and store it for the exact repetition of this experiment. Default: 0.
`treatmentname`	Default: "t0".
`experimentname`	Default: "e0".
`replay`	Default: 0.
`k`	Defautl: 2.
`trials`	Default: 1.
`tVerbose`	Default: 1.
`grammarfn`	Grammar file name.
`algorithm`	Default: "sgp".
`popsize`	Default: 200.

generations	Default: 500000.
initgene	Default: "InitGene".
selection	Default: "SUS".
mateselection	Default: "SUS".
replication	Default: "Kid2".
crossover	Default: "Cross2Gene".
mutation	Default: "MutateGene".
accept	Default: "All".
reportEvalErrors	Default: "FALSE".
crossrate	Default: 0.2.
mutrate	Default: 0.4.
ivmutrate	Default: "Const".
mutrate2	Default: 0.4.
ivcrossrate	Default: "Const".
crossrate2	Default: 0.2.
scalefactor	Default: "Uniform".
executionModel	Default: "Sequential",
verbose	Default: 0.
Gpath	Default: kSymmetryGrammarPath.
outpath	Default: ".".
semantics	Default: "byValue".
batch	Default: FALSE
everyk	Integer. Save results after every k trials. Default:10.

Details

Repeats xegaRun several times.

- As a side effect writes files with the experimental data into the directory `outpath` which must exist.
- The treatmentname may not contain numbers, because the files generated `everyk` trials have serial numbers with 3 decimal places at the end.

Value

A named list with the following elements

- `$lastGAResult`: The return object of `xegaRun()`.
- `$resultDF`: A dataframe with the following columns:
  - `$Fit`: The best fitness value of a trial.
  - `$Solution`: The best solution (the phenotype of the gene).
  - `$Seconds`: Execution time of the trial in seconds.
  - `$Generations`: The number of generations until termination.
  - `$Evaluations`: The number of fitness evaluations.
- `$tArgs`: The arguments for the treatment.
- `$xegaArgs`: The arguments passed to xega.
- `$filename`: The name of the last file written.

See Also

Other Experiment: kSymNN(), treatmentRun()

Examples

```
tmpPath<-tempdir()
gfn<-newFileName(fn="BooleanK", ftype="txt", path=tmpPath)
g<-booleanGrammarK()
writeBNF(g, gfn)
r<-kSym(treatmentname="t0", experimentname="e0", k=2, trials=1,
      grammarfn=gfn, algorithm="sgp", popsize=20,
      generations= 20, crossrate=0.2, mutrate=0.4,
      ivmutrate="Const", mutrate2=0.4,
      ivcrossrate="Const", crossrate2=0.2,
      executionModel="Sequential", verbose=1,
      Gpath="", outpath=tmpPath, semantics="byValue")
print(r$resultDF)
```

---

kSymmetryGrammarPath *The path to the grammars of the experiment.*

Description

For convenience. Should not be on cran.

Usage

```
kSymmetryGrammarPath
```

Format

An object of class character of length 1.

---

kSymNN *Wrapper for xegaRun for k-symmetry NN experiments.*

Description

Tentative. Restricts the parameter space of xegaRun. Not yet flexible at all.

Usage

```
kSymNN(
  expReplay = 0,
  treatmentname = "t0",
  experimentname = "e0",
  replay = 0,
  k = 2,
  topology = c(2, 3, 1),
```

```
  trials = 1,
  tVerbose = 1,
  algorithm = "sga",
  popsize = 200,
  generations = 5e+05,
  initgene = "InitGene",
  selection = "SUS",
  mateselection = "SUS",
  replication = "Kid2",
  crossover = "Cross2Gene",
  mutation = "MutateGene",
  accept = "All",
  reportEvalErrors = "FALSE",
  crossrate = 0.2,
  mutrate = 0.4,
  ivmutrate = "Const",
  mutrate2 = 0.8,
  ivcrossrate = "Const",
  crossrate2 = 0.4,
  scalefactor = "Uniform",
  executionModel = "Sequential",
  verbose = 0,
  outpath = ".",
  early = TRUE,
  semantics = "byValue",
  batch = FALSE,
  everyk = 10
)
```

Arguments

`expReplay`	Integer. If `expReplay>0`, then use `expReplay` as the seed of the random number generator and store it for the exact repetition of this experiment. Default: 0.
`treatmentname`	Default: "t0".
`experimentname`	Default: "e0".
`replay`	Default: 0.
`k`	Defautl: 2.
`topology`	Defautl: `c(2, 3, 1)`.
`trials`	Default: 1.
`tVerbose`	Default: 1.
`algorithm`	Default: "sga".
`popsize`	Default: 200.
`generations`	Default: 500000.
`initgene`	Default: "InitGene".
`selection`	Default: "SUS".
`mateselection`	Default: "SUS".
`replication`	Default: "Kid2".
`crossover`	Default: "Cross2Gene".

mutation	Default: "MutateGene".
accept	Default: "All".
reportEvalErrors	Default: "FALSE".
crossrate	Default: 0.2.
mutrate	Default: 0.4.
ivmutrate	Default: "Const".
mutrate2	Default: 0.4.
ivcrossrate	Default: "Const".
crossrate2	Default: 0.2.
scalefactor	Default: "Uniform".
executionModel	Default: "Sequential",
verbose	Default: 0.
outpath	Default: ".".
early	Early termination. Default: TRUE.
semantics	Default: "byValue".
batch	Default: FALSE.
everyk	Integer. Save results after every k trials. Default:10.

Details

Repeats xegaRun several times.

- As a side effect writes files with the experimental data into the directory `outpath` which must exist.
- The treatmentname may not contain numbers, because the files generated `everyk` trials have serial numbers with 3 decimal places at the end.

Value

A named list with the following elements

- `$lastGAResult`: The return object of `xegaRun()`.
- `$resultDF`: A dataframe with the following columns:
  - `$Fit`: The best fitness value of a trial.
  - `$Solution`: The best solution (the phenotype of the gene).
  - `$Seconds`: Execution time of the trial in seconds.
  - `$Generations`: The number of generations until termination.
  - `$Evaluations`: The number of fitness evaluations.
- `$tArgs`: The arguments for the treatment.
- `$xegaArgs`: The arguments passed to xega.
- `$filename`: The name of the last file written.

See Also

Other Experiment: `kSym()`, `treatmentRun()`

Examples

```
tmpPath<-tempdir()
r1<-kSymNN(treatmentname="t0", experimentname="e0",
      k=2, topology=c(2, 3, 1), trials=1,
      algorithm="sga", popsize=20,
      generations= 100, crossrate=0.2, mutrate=0.4,
      ivmutrate="Const", mutrate2=0.4,
      ivcrossrate="Const", crossrate2=0.2,
      executionModel="Sequential", verbose=1,
      outpath=tmpPath, semantics="byValue")
r1$lastGAResult$GAenv$penv$errorTable(r1$lastGAResult$solution$phenotype)
r2<-kSymNN(treatmentname="t1", experimentname="e0",
      k=3, topology=c(3, 6, 3, 1), trials=1,
      algorithm="sgde", popsize=100,
      generations= 200, crossrate=0.2, mutrate=0.4,
      ivmutrate="Const", mutrate2=0.4,
      ivcrossrate="Const", crossrate2=0.2,
      mutation="MutateGeneDE", scalefactor="Uniform",
      crossover="UCrossGene", replication="DE",
      selection="UniformP", mateselection="UniformP", accept="Best",
      executionModel="Sequential", verbose=1,
      outpath=tmpPath, semantics="byValue")
p<-r2$lastGAResult$GAenv$penv
s<-r2$lastGAResult$solution$phenotype
cbind((p$kSymmetryTable(p$k())[,1:3]), p$errorTable(s))
```

`latexBoxPlot` *Produces box-and-whisker plots of treatments of an experiment (eps-file).*

Description

Produces box-and-whisker plots of treatments of an experiment (eps-file).

Usage

```
latexBoxPlot(
  treatments,
  variable,
  title = "",
  outline = TRUE,
  notch = FALSE,
  horizontal = TRUE,
  varwidth = FALSE,
  name = "",
  inpath = ".",
  outpath = "."
)
```

Arguments

treatments	List of filenames of treatments.
variable	Name of variable.
title	Title of Boxplot. Default: "".
outline	Boolean. If FALSE, remove outliers. Default: TRUE.
notch	Boolean. If TRUE, notches are drawn. Default: FALSE. Nonoverlapping notches indicate different medians of treatments.
horizontal	Boolean. Default: TRUE. Plots are horizontal.
varwidth	Boolean. Default: FALSE. If TRUE, width of the plot indicates sample size.
name	Name of experiment. Default: "".
inpath	Path to treatment files of experiment. Default: ".".
outpath	Path to report files.

Value

Name of eps file with output.

See Also

Other Latex Elements: beamerSection(), latexEndDocument(), latexFigure(), latexMetaTable(), latexProlog(), latexSection(), latexTable(), latexTables(), latexText()

Examples

```
cat("Example: TODO.\n")
```

---

latexEndDocument *End of document (LaTeX)*

Description

End of document (LaTeX)

Usage

```
latexEndDocument()
```

Value

A text string with latex markup.

See Also

Other Latex Elements: beamerSection(), latexBoxPlot(), latexFigure(), latexMetaTable(), latexProlog(), latexSection(), latexTable(), latexTables(), latexText()

Examples

```
a<-latexEndDocument()
print(a)
cat(a)
```

latexFigure *The figure environment for a latex document.*

Description

The figure environment for a latex document.

Usage

```
latexFigure(
  name,
  fnName,
  caption = "No caption defined",
  label = "",
  width = 0.6,
  angle = -90
)
```

Arguments

name	Name of the experiment.
fnName	File name of the eps file of the graphic.
caption	Caption of the figure.
label	Label of the figure.
width	Proportion of textwidth. Default: 0.6.
angle	Angle. Default: -90.

Value

A text string with latex markup.

See Also

Other Latex Elements: beamerSection(), latexBoxPlot(), latexEndDocument(), latexMetaTable(), latexProlog(), latexSection(), latexTable(), latexTables(), latexText()

Examples

```
a<-latexFigure(name="Ex A", fnName="test.eps", caption="test", label="ltest",
               width=0.6, angle=-90)
print(a)
cat(a)
```

latexMetaTable *The meta table environment for a latex document.*

Description

The meta table environment for a latex document.

Usage

```
latexMetaTable(name, fnName, caption = "No caption defined", label = "")
```

Arguments

name	Name of the experiment.
fnName	File name of the exported table element.
caption	Caption of the table.
label	Label of the table.

Value

A text string with latex markup.

See Also

Other Latex Elements: beamerSection(), latexBoxPlot(), latexEndDocument(), latexFigure(), latexProlog(), latexSection(), latexTable(), latexTables(), latexText()

Examples

```
a<-latexMetaTable(name="Ex A", fnName="test.tex", caption="test", label="ltest")
print(a)
cat(a)
```

latexProlog *Prolog of minimal latex document.*

Description

Contains title, author and purpose of experiment.

Usage

```
latexProlog(
  name = "Missing",
  title = "Undefined",
  author = "Unknown",
  purpose = "Unknown"
)
```

Arguments

name	Name of experiment. Default: "Missing".
title	Title of the experiment. Default: "Undefined".
author	Author of document. Default: "Unknown".
purpose	Purpose of experiment. A short description. Default: "Unknown"

Value

A text string with LaTex markup.

See Also

Other Latex Elements: beamerSection(), latexBoxPlot(), latexEndDocument(), latexFigure(), latexMetaTable(), latexSection(), latexTable(), latexTables(), latexText()

Examples

```
a<-latexProlog(name="Experiment A.",
               title="First test", author="U.N. Owen",
               purpose="To show the impossible.\n")
cat(a)
```

---

latexSection *Sections of a latex document.*

Description

Sections of a latex document.

Usage

```
latexSection(name, level = "section", clearpage = TRUE)
```

Arguments

name	Name of the document element.
level	Latex markup of the document element. Default: "section". Latex supports: "section", "subsection", "subsubsection".
clearpage	Boolean. Default: TRUE. Clearpage before start of document element. Forces output of all floating elements of the previous document element.

Value

A text string with latex markup.

See Also

Other Latex Elements: beamerSection(), latexBoxPlot(), latexEndDocument(), latexFigure(), latexMetaTable(), latexProlog(), latexTable(), latexTables(), latexText()

Examples

```
latexSection(name="Design", level="section", clearpage=FALSE)
a<-latexSection(name="Design", level="section", clearpage=TRUE)
print(a)
cat(a)
```

`latexTable` *The table environment for a latex document.*

Description

The table environment for a latex document.

Usage

```
latexTable(
  df,
  caption = "",
  label = "",
  align = "",
  center = "center",
  sanitize = FALSE,
  tableLength = 60
)
```

Arguments

`df`	Dataframe or matrix.
`caption`	Caption of the table. Default: "".
`label`	Label of the table. Default: "".
`align`	Alignment of columns. Default: "".
`center`	Center table. Default: "center".
`sanitize`	Boolean. Default: FALSE.
`tableLength`	Integer. Default: 60.

Value

A text string with latex markup.

See Also

Other Latex Elements: `beamerSection()`, `latexBoxPlot()`, `latexEndDocument()`, `latexFigure()`, `latexMetaTable()`, `latexProlog()`, `latexSection()`, `latexTables()`, `latexText()`

Examples

```
df<-data.frame(Name=c("Alice", "Bob"), Age=c(25, 27))
a<-latexTable(df, caption="Alice and Bob", label="AB")
print(a)
```

latexTableLength *latex table length.*

Description

latex table length.

Usage

```
latexTableLength()
```

Value

Integer. Number of lines in beamer table.

See Also

Other Internal Configuration: `beamerTableLength()`

Examples

```
latexTableLength()
```

latexTables *Write table(s) and meta-table(s) for a latex document.*

Description

Write table(s) and meta-table(s) for a latex document.

Usage

```
latexTables(
  df,
  name,
  fnName,
  mfnName = "main",
  caption = "",
  label = "",
  align = "",
  center = "center",
  sanitize = FALSE,
  tableLength = 60,
  path = ""
)
```

Arguments

df	Dataframe.
name	Name of the experiment.
fnName	File name of the exported table element.
mfnName	File name of the exported meta table element. Default: "main".
caption	Caption of the table. Default: "".
label	Label of the table. Default: "".
align	Alignment of columns. Default: "".
center	Center table. Default: "center".
sanitize	Boolean. Default: FALSE.
tableLength	Integer. Default: 60.
path	Path. Default: outpath.

Value

Invisible 0.

See Also

Other Latex Elements: beamerSection(), latexBoxPlot(), latexEndDocument(), latexFigure(), latexMetaTable(), latexProlog(), latexSection(), latexTable(), latexText()

Examples

```
df<-data.frame(Name=c("Alice", "Bob"), Age=c(25, 27))
a<-latexTable(df, caption="Alice and Bob", label="AB")
print(a)
```

---

latexText *Text.*

Description

Text.

Usage

```
latexText(..., header = "")
```

Arguments

...	A list of text strings.
header	Name of the text.

Value

A text string.

See Also

Other Latex Elements: beamerSection(), latexBoxPlot(), latexEndDocument(), latexFigure(), latexMetaTable(), latexProlog(), latexSection(), latexTable(), latexTables()

Examples

```
a<-latexText("Here", "we",  "are")
print(a)
cat(a)
```

---

mergeTreatments *Merge treatment files of an experiment.*

Description

inpath is supposed to contain all rds-files of all treatments of an experiment. The filenames of a treatment are supposed to have the format "<treatmentname><serial>.rds". Serial is a number with three decimal places. mergeTreatments() reads all rds-files of a treatment, builds a new dataframe, and writes the result to a rds-file with the name "merge<treatmentname>.rds" in outpath.

Usage

```
mergeTreatments(inpath = ".", outpath = ".")
```

Arguments

inpath	Path of treatment files.
outpath	Path for merged files.

Value

Invisible zero.

Examples

```
## dir.create(tempfile())
## dir.create(tempfile())
## tmpPath<-tempdir()
## subdirs<-list.files(path=tmpPath)
## dataTmpPath<-paste0(tmpPath, .Platform$file.sep, subdirs[1])
## mergeTmpPath<-paste0(tmpPath, .Platform$file.sep, subdirs[2])
tmpPath<-tempdir()
dataTmpPath<-tmpPath
mergeTmpPath<-tmpPath
unlink(paste0(tmpPath, "/*"))
list.files(path=dataTmpPath)
sym4replay19<-kSym(treatmentname="BoolT1SGPreplayXIX",
              experimentname="Ex", k=2, trials=3,
              grammarfn="AndOrNotTuned1.txt", algorithm="sgp",
              popsize=20, generations=10,
              executionModel="Sequential", verbose=0,
              semantics="byValue", replay=19,
              Gpath=kSymmetryGrammarPath, outpath=dataTmpPath, everyk=1)
```

```
sym4replay0<-kSym(treatmentname="BoolT1SGPreplayZ",
              experimentname="Ex", k=2, trials=3,
              grammarfn="AndOrNotTuned1.txt", algorithm="sgp",
              popsize=20, generations=10,
              executionModel="Sequential", verbose=0,
              semantics="byValue", replay=0,
              Gpath=kSymmetryGrammarPath, outpath=dataTmpPath, everyk=1)
list.files(path=dataTmpPath)
mergeTreatments(inpath=dataTmpPath, outpath=mergeTmpPath)
list.files(path=dataTmpPath)
list.files(path=mergeTmpPath)
```

`nandpairs` *Returns a the set of production rules with symmetric variable pairs (Nand)*

Description

Symmetric variable pairs are the pairs of variables with the same truth value in the k-symmetry problem. E.g. In a 4-symmetry problem with variables V1-V4, the 2 symmetric variable pairs are (V1, V4) and (V2, V3).

Usage

```
nandpairs(varNT, varSym, k)
```

Arguments

`varNT`	The non-terminal symbol on the left-hand side of a BNF rule.
`varSym`	The symbol of the variable name. E.g. "V".
`k`	The number of bits.

Details

For the BNF notation, see the R-package `xegaBNF` The function is called by the preprocessor `xegaBNF::preBNF` and is expected to return a part of a BNF grammar specification. Used in the grammar NandTuned2.txt.

The pattern used needs to be improved.

Value

A text string with the production rules in BNF.

See Also

Other Grammar Tuning: `sym2pairs()`, `sympairs()`

Examples

```
nandpairs(varNT="<sympairs>", varSym="V", k=4)
```

`newEnvKsymmetry` *Constructor for a k-symmetry problem environment.*

Description

Constructor for a k-symmetry problem environment.

Usage

```
newEnvKsymmetry(k = 5)
```

Arguments

`k`	Integer. Number of bits.

Value

A problem environment for the k-symmetry problem. A closure with the following elements:

1. $name() The name of the problem environment.
2. $k() The number of bit of the problem.
3. $BuildTest(expr) Returns the fitness function for the boolean expression `expr`.
4. $dec2bin(x) Converts an integer into a boolean vector.
5. $kSymmetry(v) Tests if the boolean vector `v` is symmetric. Returns `0` or `1`.
6. $f(expr, gene, lF) The fitness function of the k-symmetry problem. Tests `2^k` boolean vectors for the expression `expr`. Returns the number of test errors.
7. $globalOptimum() The minimal number of errors is `0`.

See Also

Other Problem Environment Factory: `newEnvKsymmetryNN()`

Examples

```
env3symmetry<-newEnvKsymmetry(k=3)
env3symmetry$f("OR(AND(D1, D3), NOT(OR(D1, D3)))")
env3symmetry$f("OR(AND(D1, D3), AND(NOT(D1), NOT(D3)))")
env3symmetry$f("AND(AND(NOT(D1), NOT(D3)), AND(NOT(D2), NOT(D1)))")
```

`newEnvKsymmetryNN` *Constructor for a k-symmetry feedforward NN problem environment.*

Description

Constructor for a k-symmetry feedforward NN problem environment.

Usage

```
newEnvKsymmetryNN(k = 5, topology = c(5, 20, 1))
```

Arguments

`k`	Integer. Number of bits.
`topology`	Vector of integers. `topology[1]` defines the number of neurons on the input layer. `topology[length(topology)]` defines the number of neurons on the output layer.

Value

A problem environment for the k-symmetry problem. A closure with at least (TODO) the following elements:

1. $name() The name of the problem environment.
2. $k() The number of bit of the problem.
3. $BuildTest(expr) Returns the fitness function for the boolean expression `expr`.
4. $dec2bin(x) Converts an integer into a boolean vector.
5. $kSymmetry(v) Tests if the boolean vector `v` is symmetric. Returns `0` or `1`.
6. $f(expr, gene, lF) The fitness function of the k-symmetry problem. Tests `2^k` boolean vectors for the expression `expr`. Returns the number of test errors.
7. $globalOptimum() The minimal number of errors is `0`.

See Also

Other Problem Environment Factory: `newEnvKsymmetry()`

Examples

```
e3NN<-newEnvKsymmetryNN(k=3, topology=c(3, 9, 1))
e3NN$f(rndParms(e3NN$nNNparms()))
```

`newFileName` *Generate a new filename with a version number.*

Description

The filename is built by catenating the following elements:

1. `path` The path of the file.
2. `expname` is the name of the experiment. May be empty.
3. `fn` is the filename.
4. `i` is a number. Is formatted to the 3 positions (with leading `0`s).
5. `type` indicates the filetype (e.g. tex or rds).

If a filename exits, `i` is incremented by `1`. **Warning:** The operation is not atomic.

Usage

```
newFileName(fn, ftype = "tex", expname = "", path = ".")
```

Arguments

`fn`	A filename.
`ftype`	A filetype. Default: "tex".
`expname`	Name of an Experiment. Default: "".
`path`	A filepath. Default: "".

Value

A filename

See Also

Other File I/O: `fullFileNames()`, `writeText()`

Examples

```
path<-tempdir()
a<-newFileName(fn="test", ftype="rds", path=path)
saveRDS(a, file=a)
cat("Filename 1:",a, "\n")
b<-newFileName(fn="test", ftype="rds", path=path)
cat("Filename 2:",b, "\n")
saveRDS(b, file=b)
c<-newFileName(fn="test", ftype="rds", path=path)
cat("Filename 3:",c, "\n")
```

newSmoofWrapper *Factory for wrapping smoof functions.*

Description

A minimal wrapper for smoof functions. This is an R-example for the facade architectural pattern.

Usage

```
newSmoofWrapper(smoofFN)
```

Arguments

smoofFN	smoof function (usually a call to the generator of a smoof function.

Details

The R-package smoof provides a large number of benchmark and test functions for single and multiobjective optimization. See https://cran.r-project.org/web/packages/smoof/index.html

Value

A list of functions (`$name()`, `$bitlength()`, `$genelength()`, `$lb()`, `$ub()`, `$f(parm, gene=0, lF=0)`, ...) For details, see the interface description in the package description.

References

Bossek, Jakob (2017) smoof: Single- and Multi-Objective Optimization Test Functions. The R Journal, 9(1), 103-113.

Examples

```
require(smoof)
t1<-newSmoofWrapper(makeSchwefelFunction(10))
```

NN *A feedforward neural network with topology* top.

Description

A feedforward neural network with topology top.

Usage

```
NN(parm, topology, AH = ReLU, AO = ReLU, data)
```

Arguments

parm	A parameter vector of length numberOfNNParms(top).
topology	The topology of the neural network.
AH	Activation function hidden layers. Default: ReLU.
AO	Activation function output layer. Default: ReLU.
data	An input data matrix. The column number must match top[1]. E.g. for xor.

Value

A vector of activation values of the output layer.

See Also

Other Feedforward Neural Network: P2NN(), ReLU(), RsquareNN(), numberOfNNParms(), printNNweights(), rndParms()

Examples

```
NN(rndParms(numberOfNNParms(c(2, 3, 1))), c(2, 3, 1), data=xorData)
```

---

numberOfNNParms *Computes the number of parameters of a NN for a given topology vector.*

Description

Computes the number of parameters of a NN for a given topology vector.

Usage

```
numberOfNNParms(topology)
```

Arguments

topology	An integer vector. The i-th element defined the number of nodes of the i-th layer. The first layer is the input layer. The last layer is the output layer.

Value

The number of parameters.

See Also

Other Feedforward Neural Network: NN(), P2NN(), ReLU(), RsquareNN(), printNNweights(), rndParms()

Examples

```
numberOfNNParms(c(2, 3, 1))
```

P2NN *Convert the parameter vector of a NN into a list of weight matrices.*

Description

The dimensions of the weight matrices are defined by the vector `topology`.

Usage

```
P2NN(parm, topology)
```

Arguments

`parm`	A parameter vector of length `numberOfNNParms(topology)`.
`topology`	The topology of the neural network.

Value

A list of weight matrices.

See Also

Other Feedforward Neural Network: `NN()`, `ReLU()`, `RsquareNN()`, `numberOfNNParms()`, `printNNweights()`, `rndParms()`

Examples

```
topology<-c(1, 3, 2, 1)
parms<-rndParms(numberOfNNParms(topology))
P2NN(parms, topology)
```

pause *Wait for input.*

Description

Wait for input.

Usage

```
pause()
```

Value

Keyboard input of user (invisible).

See Also

Other Console Elements: `beamerText()`, `consoleBoxPlot()`, `consoleEndDocument()`, `consoleFigure()`, `consoleProlog()`, `consoleSection()`, `consoleTable()`, `consoleText()`

Examples

```
pause()
a<-pause()
print(a)
```

`pl2df` *Convert a named parameter list into a dataframe.*

Description

Three parameters need special handling:

1. `$tRNG`: Convert to one string.
2. `$penv`: Replace penv by name of penv.
3. `$grammar`: Replace grammar by its name.

Usage

```
pl2df(pl)
```

Arguments

`pl`	Parameter list.

Value

A dataframe with one column: The parameter values. The rownames are the (raw) parameter names.

See Also

Other Design of Experiment: `designOfAllTreatments()`, `designOfExperiment()`, `designOfTreatment()`, `designSplitParameter()`, `getTreatment()`

Examples

```
cat("TBD!\n")
```

printNNweights *Print the parameter vector of a NN with topology c(2, 3, 1)*

Description

TODO: Add row and column names.

Usage

```
printNNweights(parm, topology)
```

Arguments

parm	A parameter vector.
topology	A topology vector.

Value

Invisible zero.

See Also

Other Feedforward Neural Network: NN(), P2NN(), ReLU(), RsquareNN(), numberOfNNParms(), rndParms()

Examples

```
printNNweights(parm=rndParms(numberOfNNParms(c(2, 3, 1))), topology=c(2, 3, 1))
```

ReLU *Activation function of a rectified linear unit (RELU).*

Description

Activation function of a rectified linear unit (RELU).

Usage

```
ReLU(z)
```

Arguments

z	A (real) matrix or vector.

Value

In r is identical to z except for the negative elements of z. These are set to 0.

See Also

Other Feedforward Neural Network: NN(), P2NN(), RsquareNN(), numberOfNNParms(), printNNweights(), rndParms()

Examples

```
a<-rndParms(numberOfNNParms(c(2, 3, 1)))
a
ReLU(a)
```

`rmOutliers` *Removes outliers of data vector.*

Description

An outlier is a datapoint outside `coef` times of the box size. See `boxplot.stats`.

Usage

```
rmOutliers(x, coef = 1.5)
```

Arguments

`x`	A numeric vector.
`coef`	Scaling of threshold for outliers. Default: 1.5.

Value

A numeric vector.

Examples

```
x<-rnorm(30, mean=0, sd=2)
rmOutliers(x, coef=0)
rmOutliers(x, coef=0.5)
rmOutliers(x, coef=1.0)
rmOutliers(x)
```

rndParms *Generates a random vector of length n in an n-dimensional hypercube.*

Description

Generates a random vector of length n in an n-dimensional hypercube.

Usage

```
rndParms(n, lb = -1, ub = 1)
```

Arguments

n	Number of elements.
lb	Lower bound.
ub	Upper bound.

Value

A random vector of length n with elements in [lb, ub].

See Also

Other Feedforward Neural Network: NN(), P2NN(), ReLU(), RsquareNN(), numberOfNNParms(), printNNweights()

Examples

```
rndParms(numberOfNNParms(c(2, 3, 1)))
```

RsquareNN *R-square.*

Description

R-square.

Usage

```
RsquareNN(predicted, actual)
```

Arguments

predicted	Predicted output vector.
actual	Actual output vector.

Value

The sum of squared errors.

See Also

Other Feedforward Neural Network: NN(), P2NN(), ReLU(), numberOfNNParms(), printNNweights(), rndParms()

Examples

```
RsquareNN(c(1, 2, 3), c(0, 3, 3))
```

`selectFromExperiment` *Select a subset of the descriptive statistics of an experiment.*

Description

Specify either treatment name or variable name or none. Returns the appropriate subset of records of the dataframe of all descriptive statistics of the experiment.

Usage

```
selectFromExperiment(
  treatments,
  treatment = NULL,
  variable = NULL,
  inpath = "."
)
```

Arguments

`treatments`	List of treatment names of experiment.
`treatment`	Name of treatment. Default: NULL.
`variable`	Name of variable. Defautl: NULL.
`inpath`	Path to treatement rds-files of experiment. Default: ".".

Value

A dataframe with the following columns:

1. `$Treatment`: Name of treatment.
2. `$Trials`: Number of trials.
3. `$Variable`: Name of Variable.
4. `$min`: Minimum.
5. `$mean`: Mean.
6. `$sd`: Standard deviation.
7. `$max`: Minimum.

See Also

Other Statistics of Experiment: summaryExperiment()

Examples

```
cat("TBD! \n")
```

`summaryExperiment` *Dataframe of descriptive statistics for experiment.*

Description

Dataframe of descriptive statistics for experiment.

Usage

```
summaryExperiment(treatments, byVars = TRUE, inpath = ".")
```

Arguments

`treatments`	List of treatment names of experiment.
`byVars`	Boolean. If TRUE, Sort by variables. Default: TRUE.
`inpath`	Path to treatement rds-files of experiment. Default: ".".

Value

A dataframe with the following columns:

1. `$Treatment`: Name of treatment.
2. `$Trials`: Number of trials.
3. `$Variable`: Name of Variable.
4. `$min`: Minimum.
5. `$mean`: Mean.
6. `$sd`: Standard deviation.
7. `$max`: Minimum.

See Also

Other Statistics of Experiment: `selectFromExperiment()`

Examples

```
cat("TBD! \n")
```

summaryTreatment *Dataframe of descriptive statistics of all variables of a treatment.*

Description

Dataframe of descriptive statistics of all variables of a treatment.

Usage

```
summaryTreatment(name, df)
```

Arguments

name	Name of treatment.
df	Experimental results of treatment. Each record is the result of one trial.

Value

A dataframe with the following columns:

1. $Treatment: Name of treatment.
2. $Trials: Number of trials.
3. $Variable: Name of Variable.
4. $min: Minimum.
5. $mean: Mean.
6. $sd: Standard deviation.
7. $max: Minimum.

See Also

Other Descriptive Statistics (Internal): summaryVariable()

Examples

```
df<-data.frame(Fitness=sample(100, 10), Seconds=sample(50, 10),
               Generations=sample(30, 10), Evaluations=sample(3000, 10))
summaryTreatment("MC-GP", df)
```

summaryVariable *Summarize a performance variable of one treatment of an experiment.*

Description

Summarize a performance variable of one treatment of an experiment.

Usage

```
summaryVariable(name, variable, column)
```

Arguments

name	Name of treatment.
variable	Name of variable.
column	Vector of values of variable.

Value

A dataframe with the following columns:

1. $Treatment: Name of treatment.
2. $Trials: Number of trials.
3. $Variable: Name of Variable.
4. $min: Minimum.
5. $mean: Mean.
6. $sd: Standard deviation.
7. $max: Minimum.

See Also

Other Descriptive Statistics (Internal): summaryTreatment()

Examples

```
summaryVariable("MC-GP", "Seconds", sample(100, 10))
```

`sym2pairs` *Returns a the set of production rules with symmetric variable pairs and their negation.*

Description

Symmetric variable pairs are the pairs of variables with the same truth value in the k-symmetry problem. E.g. In a 4-symmetry problem with variables V1-V4, the 2 symmetric variable pairs are (V1, V4) and (V2, V3).

Usage

```
sym2pairs(varNT, varSym, k)
```

Arguments

`varNT`	The non-terminal symbol on the left-hand side of a BNF rule.
`varSym`	The symbol of the variable name. E.g. "V".
`k`	The number of bits.

Details

For the BNF notation, see the R-package `xegaBNF` The function is called by the preprocessor `xegaBNF::preBNF` and is expected to return a part of a BNF grammar specification. Used in the grammar AndOrNotTuned2.txt.

Combines a symmetric pairs with its negation. This further reduces the search space size for grammar-based genetic programming.

Value

A text string with the production rules in BNF.

See Also

Other Grammar Tuning: `nandpairs()`, `sympairs()`

Examples

```
sym2pairs(varNT="<sympairs>", varSym="V", k=4)
```

sympairs — *Returns a the set of production rules with symmetric variable pairs.*

Description

Symmetric variable pairs are the pairs of variables with the same truth value in the k-symmetry problem. E.g. In a 4-symmetry problem with variables V1-V4, the 2 symmetric variable pairs are (V1, V4) and (V2, V3).

Usage

```
sympairs(varNT, varSym, k)
```

Arguments

varNT	The non-terminal symbol on the left-hand side of a BNF rule.
varSym	The symbol of the variable name. E.g. "V".
k	The number of bits.

Details

For the BNF notation, see the R-package xegaBNF The function is called by the preprocessor xegaBNF::preBNF and is expected to return a part of a BNF grammar specification. Used in the grammar AndOrNotTuned2.txt.

Value

A text string with the production rules in BNF.

See Also

Other Grammar Tuning: nandpairs(), sym2pairs()

Examples

```
sympairs(varNT="<sympairs>", varSym="V", k=4)
```

t1 — *A test case for the 5-symmetry problem.*

Description

A test case for the 5-symmetry problem.

Usage

```
t1
```

Format

An object of class character of length 1.

tformat *Format a number. .*

Description

Format a number. .

Usage

```
tformat(x)
```

Arguments

x	A real.

Value

A string.

Examples

```
tformat(1/3)
```

treatmentReporter *Reporter of and observer treatment performance and solution data.*

Description

Returns a dataframe with the performance data and the phenotype of the solution of a single treatment. If `!(verbose==0)` print data on console.

Usage

```
treatmentReporter(experimentName, treatmentName, s, i, verbose = 0)
```

Arguments

experimentName	Name of experiment.
treatmentName	Name of treatment.
s	A solution object of xegaRun (with elements $popStat, $fit, $solution, $evalFail, $GAconfig, $GAenv, and $timer.
i	Number of iteration.
verbose	If not 0, output treatment data to console. Progress report. Default: 0 - Not output to console.

Value

A dataframe with a single row and the following columns:

- `$Fit`: The phenotype value of the solution.
- `$Solution`: The solution (its phenotype).
- `$Seconds`: Execution time of `xegaRun()` in seconds.
- `$Generations`: Number of generations until termination.
- `$Evaluations`: Number of evaluations until termination.

See Also

Other Wrapper (Internal): `iterationReporter()`

`treatmentRun` *Runs a treatment with xegaRun repeatedly* `trials` *times.*

Description

Tentative. Not yet flexible at all.

Usage

```
treatmentRun(
  experimentName = "e0",
  treatmentName = "t0",
  tReplay = 0,
  trials = 1,
  everyK = 10,
  outpath = ".",
  tVerbose = 0,
  ...
)
```

Arguments

`experimentName`	Name of experiment. Default: "e0".
`treatmentName`	Name of treatment. Default: "t0".
`tReplay`	Integer. Seed of Random Number. Default: 0 (No seeding)
`trials`	Default: 1.
`everyK`	Save results of trials after `everyK` trials (Default: 10).
`outpath`	Path for treatment results. Deafult: ".".
`tVerbose`	Screen output. Default: 0 (no output). • `tVerbose=0`: No output. • `tVerbose=1`: Mean performance after i trials. • `tVerbose>1`: Mean performance after i trials and performance of i-th trial.
`...`	Arguments for xegaRun of this treatment.

Details

Repeats xegaRun several times.

- As a side effect writes files with the experimental data into the directory `outpath` which must exist.
- The treatmentname may not contain numbers, because the files generated `everyk` trials have serial numbers with 3 decimal places at the end.

Value

A named list with the following elements

- `$lastGAResult`: The return object of `xegaRun()`.
- `$resultDF`: A dataframe with the following columns:
  - `$Fit`: The best fitness value of a trial.
  - `$Solution`: The best solution (the phenotype of the gene).
  - `$Seconds`: Execution time of the trial in seconds.
  - `$Generations`: The number of generations until termination.
  - `$Evaluations`: The number of fitness evaluations.
- `$tArgs`: Arguments of a treatmentRun: `$tRNG`, `$tReplay`, `$treatmentName`, `$trials`, `$everyK`, `$outPath`, `$batchPath`, `$tVerbose`.
- `$xegaArgs`: Arguments of `xega::xegaRun`.
- `$filename`: The name of the last file written.

See Also

Other Experiment: `kSym()`, `kSymNN()`

Examples

```
require(xegaBNF)
tmpPath<-tempdir()
gBNF<-compileBNF(preBNF(booleanGrammarK, list(k=3)))
kSymEnv<-newEnvKsymmetry(k=3)
r<-treatmentRun(experimentName="ExpA", treatmentName="t0", tReplay=0,
      trials=2, everyK=1, outpath=tmpPath, tVerbose=1,
      penv=kSymEnv, grammar=gBNF, algorithm="sgp", max=FALSE, popsize=20,
      generations= 5, crossrate=0.2, mutrate=0.4,
      ivmutrate="Const", mutrate2=0.4,
      ivcrossrate="Const", crossrate2=0.2,
      executionModel="Sequential", verbose=1,
      semantics="byValue")
print(r$resultDF)
```

variableNamesLHS *Generates a list of alternative variable names.*

Description

Generates a list of alternative variable names.

Usage

```
variableNamesLHS(varSym, k)
```

Arguments

varSym	Variable symbol. E.g. "D"-
k	Number of bits.

Value

Text. Contains the LHS of a production rule with a list of k variable names in EBNF.

See Also

Other xegaBNF: booleanGrammarK(), writeBNF()

writeBNF *Write a grammar file.*

Description

Write a grammar file.

Usage

```
writeBNF(g, fn = NULL, eol = "\n")
```

Arguments

g	A grammar object.
fn	A filename.
eol	End of line symbol.

See Also

Other xegaBNF: booleanGrammarK(), variableNamesLHS()

writeText *Write a text file.*

Description

Write a text file.

Usage

```
writeText(text, fn = NULL)
```

Arguments

text	A text string.
fn	File name. Default: NULL.

Value

A text string (invisible)

See Also

Other File I/O: fullFileNames(), newFileName()

Examples

```
path<-tempdir()
a<-newFileName(fn="test", ftype="txt", path=path)
b<-writeText(a, fn=a)
```

xegaCE *Package xegaCE*

Description

xegaCE Support for compuational experiments with xega.

Organization of a Computational Experiment

The following is a guideline of how to organize the files of a computational experiment:

1. Use a separate directory for each computational experiment. E.g. ~/experiment.
2. The directory ~/experiment contains:
   - A README file informing about the experiment.
   - The R-file Experiment.R with the specification of of the treatments of the experiment.
   - The R-file Merge.R which merges multiple treatment files.
   - The R-file Report.R which generates the report about the result(s) of the experiment.
   - The directory ./data for the rds-files with the experimental results.
   - The directory ./merge for the merged rds-files of the treatments of an experiment.
   - The directory ./report for the latex elements of the report.

The Process of a Computational Experiment

A computational experiment has the following steps:

- **I. The Experiment.** A computational experiments consists of the repeated execution of each of its treatments. **The function** `kSym()`**.** implements the set of potential treatments of the experiment. A treatment is a call of `kSym()` with a set of parameters specified (definition of the treatment). A definition of an experiment is an R-script which contains all treatments of the experiment. The directory for the experimental data must exist and is specified by the `outpath` parameter of `kSym()`. The parameter `trials` specifies the number of repetitions of a treatment. The parameter `everyk` specifies the number of repetitions between saves of experimental results. A call of `kSym()` produces at least one rds-file with the experimental results of a treatment (if `everyk>trials`). Repeated runs of the R-script defining a computational experiment produce new rds-files and never overwrite rds-files.
- **II. Merging Results of the Experiment.** The function `mergeTreatments` reads all treatment files from the directory specified by `inpath`, merges multiple experimental result files from this directory and writes a single file for each treatment to the directory specified by `outpath`. Files in this directory are overwritten.
- **III. Report of the Experiment.** The report of the experiment is specified in an R-script. The report elements read the rds-files of the treatments from the merge directory and write the latex code to the report directory.

Elements of the Report of a Computational Experiment

- Document structure: Preamble, sections, and end of document.
  - experimentStart(name, title, author, purpose)
  - experimentSection()
  - experimentEnd()
- Parts of experiment: Design of experiment, treatments and their analysis.
  - experimentDesign()
  - experimentTreatments()
  - experimentAnalysisOfTreatment()
  - experimentEffectReport()
- Tests of Properties of Random Variables
  - experimentIsNormal() shapiro.test()
- Tests of Hypothesis:
  - experimentHypothesisTests() t.test(), wilcox.test()
- Tables:
  - experimentGrammarTable()
  - experimentSolutionTable()
  - experimentStatistic()
- Figures:
  - experimentSolutionGenotype()
  - experimentPlotPopStats()
  - experimentBoxPlot()

k-Symmetry Boolean Functions

1. newEnvKsymmetry, a constructor (a function facotry) for problem environments for k-symmetry functions. Implemented in file envKsymmetry.R.
2. Some functions for supporting grammar tuning. Implemented in file grammarTuning.R.
3. File I/O. Implemented in file fileIO.R.
4. A wrapper function for xega for running a single treatment of a computational experiment repeatedly. Implemented in file kSymmetryWrapper.R.
5. A merge function for combining the result of multiple runs of treatments of a computational experiment.
6. A set of functions for documenting a computational experiment.

Additional Problem Environments

Problem environments for the package xega must be implemented as lists of functions or function factories which return with at least the following functions:

- <factory name>$name() is a constant function which returns the name of the environment.
- <factory name>$bitlength() is a constant function which returns a vector which returns for each parameter the number of bits used for coding the parameter value.
- <factory name>$genelength() is a constant function which returns the number of bits of a gene.
- <factory name>$lb() is a constant function which returns the vector of lower bounds of the parameters.
- <factory name>$ub() is a constant function which returns the vector of upper bounds of the parameters.
- <factory name>$f(parm, gene, lF) specifies the function with the parameter vector parm which should be optimized. The variables gene and lF extend the interface between genetic algorithm and problem environment. Both variables are not used in this package.

The constant functions <factory name>$bitlength(), <factory name>$genelength(), <factory name>$lb(), <factory name>$ub() specify the precision of the parameters as well as the hypercube in which the optimal solution(s) are searched. These four functions provide the information needed for decoding binary genes. For real coded genes, length(<factory name>$bitlength()) indicates the number of parameters.

Three additional functions can be provided:

- <factory name>$terminate(solution) TRUE, if the solution meets an early termination condition.
- <factory name>$describe() prints a description of the function (known mathematical properties and a reference to the origin of the function).
- <factory name>$solution() specifies the global minimum/maximum solution value and the list of minimal/maximal points. The function penvValidate(penv) validates the correct implementation of the function f of the environment.

At the moment, the package contains:

1. A wrapper for the single-objective optimization test functions of the smoof package of Bossek (2017).

Copyright

(c) 2025 Andreas Geyer-Schulz

License

MIT

URL

https://github.com/ageyerschulz/xegaCE

Installation

From CRAN by `install.packages('xegaCE')`

Author(s)

Andreas Geyer-Schulz

References

Bossek, Jakob (2017) smoof: Single- and Multi-Objective Optimization Test Functions. The R Journal, 9(1), 103-113.

See Also

Useful links:

- https://github.com/ageyerschulz/kSymmetry

`xegaPlotPopStats` *Plot population fitness statistics*

Description

`xegaPlotPopStats` plots the population fitness statistics (min, Q1, mean, median, Q3, max) of a run of a genetic algorithm.

For an introduction to graphics in R, see Paul Murrell (2006) R-Graphics, Chapman & Hall/CRC, Boca Raton

Usage

```
xegaPlotPopStats(
  xegaResult,
  epsfile = FALSE,
  filename = NULL,
  description = ""
)
```

Arguments

xegaResult	the result of RunSGA (or RunSGP).
epsfile	If epsfile is FALSE, then the plot is shown on an X11-device else an epsfile with the name of the problem environment is printed.
filename	If specified, overrides filename of epsfile.
description	Additional text.

Value

the return code of device.off()

References

Murrell, Paul (2006): *R-Graphics*. Chapman & Hall/CRC, Boca Raton.

TODO: Automatic file names for printing several runs ...

Examples

```
library(xega)
a<-xegaRun(Parabola2D)
xegaPlotPopStats(a)
## Not run: xegaPlotPopStats(a, epsfile=TRUE)
```

---

xorData *Data for the xor problem.*

Description

Data for the xor problem.

Usage

```
xorData
```

Format

An object of class matrix (inherits from array) with 4 rows and 3 columns.

Index