Package ‘labsimplex’

February 13, 2020

Version 0.1.2

Date 2018-11-11

Title Implementation of simplex optimization algorithms for laboratory

Author Cristhian Paredes [aut, cre]
Jesús Ágreda [aut],

Maintainer Cristhian Paredes <craparedesca@unal.edu.co>

Description Implementation of simplex algorithms for optimizing experimental conditions. It includes fixed-size algorithm reported by Spendley et al. (1962) <doi:10.1080/00401706.1962.10490033> and variable-size algorithm reported by Nelder and Mead (1965) <doi:10.1093/comjnl/7.4.308>.

License GPL (>= 2)

Encoding UTF-8

Imports scatterplot3d (>= 0.3-41)

RoxygenNote 6.1.1

Roxygen list(wrap = FALSE)

Suggests knitr, rmarkdown, FrF2

VignetteBuilder knitr

NeedsCompilation no

R topics documented:

`labsimplex-package` *labsimplex: Implementation of simplex optimization algorithms for laboratory applications.*

Description

The labsimplex package provides tools to optimize a process by using the simplex algorithms fixed-size reported by Spendley et al. (1962) and variable-size reported by Nelder and Mead (1965).

Details

The package uses a list-like `'smplx'` class object to store the simplex information including all the vertices coordinates and experimental responses.

labsimplex functions

The labsimplex functions allow generate a new `'smplx'` class object, assing responses to the vertices to generate the next one and to visualize different spatial representation of the *n*-dimensional simplex in 2D or 3D proyections. Detailed information can be found by typing `vignette('labsimplex')`.

Author(s)

Cristhian Paredes, <`craparedesca@unal.edu.co`>

Jesús Ágreda, <`jagreda@unal.edu.co`>

References

Nelder, J. A., and R. Mead. 1965. “A Simplex Method for Function Minimization.” The Computer Journal 7 (4): 308–13.

Spendley, W., G. R. Hext, and F. R0. Himsworth. 1962. “Sequential Application of Simplex Designs in Optimization and Evolutionary Operation.” Technometrics 4 (4): 441–61.

`adjustVertex` *Modify given coordinates of given vertices of a simplex*

Description

Changes the coordinates of generated vertices when slightly differences were impossible to avoid at the moment of setting the experiment (e.g. small differences in mass components when preparing a mixture).

Usage

```
adjustVertex(simplex, newcoords, overwrite = FALSE)
```

Arguments

`simplex` — `'smplx'` type object containig simplex information

`newcoords` — List with elements named like the vertices to be modified. Each element must have a vector with the actual (ordered) coordinates used in the experiment. `NA` may be used to indicate coordinates that were unchanged.

`overwrite` — logical argument. If `TRUE` the output simplex will replace the provided one in the `simplex` parameter. Defauklt `overwrite` = `TRUE`

Value

A 'smplx' type object with the modified simplex information.

Author(s)

Cristhian Paredes, <`craparedesca@unal.edu.co`>

Jesús Ágreda, <`jagreda@unal.edu.co`>

Examples

```
simplex <- labsimplex(N = 3, start = c(7, 25, 0.15),
                      stepsize = c(0.2, 5, 0.02))
adjustVertex(simplex = simplex, newcoords = list(Vertex.1 = c(7, NA, NA),
                                                 Vertex.3 = c(7.2, NA, NA)),
             overwrite = TRUE)
```

`generateVertex` *Generates the new vertex of a simplex*

Description

Gives the coordinates for the new vertex that must be performed based on the response values for vertices on the current simplex and considering the optimization criteria.

Usage

```
generateVertex(simplex, qflv = NULL, crit = "max", algor = "fixed",
  overwrite = FALSE)
```

Arguments

`simplex` — `'smplx'` type object containig simplex information

`qflv` — value of the response for the last vertex (or last vertices if it is the first simplex).

`crit` — optimization criteria indicating if the goal is maximize (`"max"`) or minimize (`"min"`) the response. It can also be a numeric value to which the response is supposed to approach.

`algor` algorithm to be followed in the vertex generation. `"fixed"` for a fixed-size simplex following Spendley (1962) algorithm or `"variable"` for a variable-size simplex following Nelder and Mead (1965) algorithm

`overwrite` logical argument. If `TRUE` the output simplex will replace the provided one in the `simplex` parameter. Defauklt `overwrite = TRUE`

Details

When minimization is the criteria, the algorithm will tend to approach zero. If negative responses are possible and the most negative value is desired, a very large negative number must be provided in `crit` parameter.

Value

A 'smplx' type object with the new simplex information including the conditions for the new experiment to be permormed.

Author(s)

Cristhian Paredes, <`craparedesca@unal.edu.co`>

Jesús Ágreda, <`jagreda@unal.edu.co`>

Examples

```
simplex3D <- labsimplex(N = 3)
simplex3D <- generateVertex(simplex = simplex3D, qflv = rnorm(4))
## Not run:
## Optional form:
## After obtaining the response for the last vertex generated:
NV <- rnorm(1)
generateVertex(simplex = simplex3D, qflv = NV, overwrite = TRUE)

## End(Not run)
```

`labsimplex` *Generates simplex object with coordinates of starting vertices.*

Description

Generates a `smplx` class object containing the coordinates of the N+1 vertices of a simplex in a N-dimensional space. Addionally can check if faces for given coordinates of a used defined simplex are in different hyperplanes.

Usage

```
labsimplex(N, start = NULL, centroid = NULL, stepsize = NULL,
  usrdef = NULL, var.name = NULL)
```

Arguments

`N` number of dimentions (variables) in the space

`start` numeric vector of size `N` with initial coordinates for the first vertex

`centroid` coordinates of centroid for initial simplex

`stepsize` numeric vector of size `N` with the step size for each coordinate

`usrdef` `(N+1)xN` matrix containig in (N+1) rows the N coordinates for each vertex

`var.name` vector containing the names for the variables

Details

The only non-optional parameter is `N` that relates the simplex dimensionality. All other parameters allows tunning the properties of generated simplex or define one using defined coordinates. Once the simplex is genereated, the experiments under the conditions indicated for each variable at each vertex must be carried and the response obtained. Those responses are assigned to the `smplx` object at the moment of generating the new vertex (see `generateVertex`).

If the initial simplex is generated using the function, it will be the result of modifications made to a regular simplex centered at the origin. This regular simplex coordinates are generated following the general algorithm for the cartesian coordinates for regular n-dimensional simplex. The algorithm considerates that all vertices must be equal distanced from simplex centroid and all angles subtended between any two vertices and the centroid of a simplex are equal (with a value of arccos(-1/N)).

Value

A `smplx` type object with the brand new simplex information.

Author(s)

Cristhian Paredes, <`craparedesca@unal.edu.co`>

Jesús Ágreda, <`jagreda@unal.edu.co`>

References

Nelder, J. A., and R. Mead. 1965. “A Simplex Method for Function Minimization.” The Computer Journal 7 (4): 308–13.

Spendley, W., G. R. Hext, and F. R0. Himsworth. 1962. “Sequential Application of Simplex Designs in Optimization and Evolutionary Operation.” Technometrics 4 (4): 441–61.

Examples

```
  labsimplex(N = 3)
  labsimplex(N = 3, centroid = c(66, 1, 12), stepsize = c(10, 0.1, 2),
             var.name = c('potential', 'pH', 'T'))

  labsimplex(N = 3, usrdef = rbind(c(2, 0, 0), c(-0.5, 1, 0),
                                   c(-0.5, -0.7, 1), c(-0.5, -0.4, -0.6)))
## Not run:
## A user defined coordinates may define faces that rely on same hyperplane:
  labsimplex(N = 3, usrdef = rbind(c(2, 0, 0), c(-0.5, 0, 0), c(0, 0, 0),
                                   c(-0.5, -0.4, -0.6)))

## End(Not run)
```

`plot.smplx` *Makes a two dimentional plot of a simplex object.*

Description

The function generates a 2D plot for a `smplx` class object having dimensionality of at least 2. When dimensionality is higher than 2, the function plots a 2D proyection of selected dimensions.

Usage

```
## S3 method for class 'smplx'
plot(x, sel.dim = NULL, all.ver = TRUE,
  all.lin = TRUE, expand = TRUE, exp.fac = 1.5, ...)
```

Arguments

`x` `smplx` class object containing the vertices coordinates.

`sel.dim` numeric or char vector for variables to be considered when simplex dimensionality is higher than 2. If `numeric` form it must contain dimensions ordinal number. If `char`, it must contain desired dimensions names.

`all.ver` logical. Should all vertex be plotted? If `FALSE` draws only vertices corresponding to current simplex.

`all.lin` logical. Should all lines be drawn? If `FALSE` draws only last simplex.

`expand` logical. Should the plot scales be expanded?

`exp.fac` expansion factor used when `expand = TRUE`.

`...` other graphical parameters used in plot()

Details

For 3D representations of simplex with dimensionality higher than 2 use `plotSimplex3D`

Value

2D proyection of the simplex coordinates

Author(s)

Cristhian Paredes, `<craparedesca@unal.edu.co>`

Jesús Ágreda, `<jagreda@unal.edu.co>`

Examples

```
plot(x = labsimplex(N = 2))
plot(x = labsimplex(N = 2), expand = FALSE)

plot(x = labsimplex(N = 8))
plot(x = labsimplex(N = 8), sel.dim = c(3, 4))

## Simulation of the real proccess where a simplex is made and evaluated
set.seed(12)
simplex2D <- labsimplex(N = 2)
plot(x = simplex2D)
generateVertex(simplex = simplex2D, qflv = rnorm(3), overwrite = TRUE)
plot(x = simplex2D)
generateVertex(simplex = simplex2D, qflv = rnorm(1), overwrite = TRUE)
plot(x = simplex2D)
generateVertex(simplex = simplex2D, qflv = rnorm(1), overwrite = TRUE)
plot(x = simplex2D)
```

---

plotSimplex3D *Makes a three dimentional plot of a simplex object.*

Description

The function generates a 3D plot for a `smplx` object having dimensionality of at least 3. When dimensionality is higher than 3, the function plots a 3D proyection of selected dimensions.

Usage

```
plotSimplex3D(simplex, sel.dim = NULL, all.ver = TRUE,
  all.lin = TRUE, main = NULL, angle = 30, ...)
```

Arguments

`simplex`	`'smplx'` type object containig simplex information
`sel.dim`	numeric or char vector for variables to be considered when simplex dimensionality is higher than 3. If `numeric` form it must contain dimensions ordinal number. If `char`, it must contain dimensions names.
`all.ver`	logical. Should all vertex be plotted? If `FALSE` draws only vertices corresponding to current simplex.
`all.lin`	logical. Should all lines be drawn? If `FALSE` draws only last simplex.
`main`	title for the plot.
`angle`	angle for perspective between x and y axis.
`...`	other arguments passed to scatterplot3d::scatterplot3d

Value

3D proyection of the simplex coordinates.

Author(s)

Cristhian Paredes, <craparedesca@unal.edu.co>

Jesús Ágreda, <jagreda@unal.edu.co>

Examples

```
plotSimplex3D(simplex = labsimplex(N = 3))

plotSimplex3D(simplex = labsimplex(N = 8))
plotSimplex3D(simplex = labsimplex(N = 8), sel.dim = c(4, 6, 8))

## Simulation of the real proccess where a simplex is made and measured,
## plotted nad the new vertex is measuresd after it is generated.
set.seed(12)
simplex3D <- labsimplex(N = 3)
plotSimplex3D(simplex = simplex3D)
generateVertex(simplex = simplex3D, qflv = rnorm(4), overwrite = TRUE)
plotSimplex3D(simplex = simplex3D)
generateVertex(simplex = simplex3D, qflv = rnorm(1), overwrite = TRUE)
plotSimplex3D(simplex = simplex3D)
generateVertex(simplex = simplex3D, qflv = rnorm(1), overwrite = TRUE)
plotSimplex3D(simplex = simplex3D)
```

plotSimplexResponse *Plots the response versus the vertex number of a simplex optimization.*

Description

The function generates a plot for a smplx class object whose vertices must have a response value assigned. The response is plotted against the vertex number. The only vertex allowed to not having response assigned is the last one.

Usage

```
plotSimplexResponse(x, ...)
```

Arguments

x	smplx class object containig the coordinates of the vertices and their response values.
...	other graphical parameters used in plot()

Details

If the simplex object being ploted was obtained using a variable size algorithm, some experimental points could be disregarded and will be shown with a red mark indicating that the vertex was not used in new vertices calculations. Those points are also ignored by the line that links the data in the scaterplot and pretends to show a tendency to better values as more vertices are evaluated.

Value

Plot of response against vertex number.

Author(s)

Cristhian Paredes, <craparedesca@unal.edu.co>

Jesús Ágreda, <jagreda@unal.edu.co>

Examples

```
set.seed(1)
# Generate a simplex and add some responses to the vertices
simplex <- labsimplex(N = 3)
generateVertex(simplex, qflv = rnorm(4), algor = 'variable',
  overwrite = TRUE)
generateVertex(simplex, qflv = 2.75, algor = 'variable', overwrite = TRUE)
generateVertex(simplex, qflv = 0.8,  algor = 'variable', overwrite = TRUE)
generateVertex(simplex, qflv = 2.94, algor = 'variable', overwrite = TRUE)
generateVertex(simplex, qflv = 3.14, algor = 'variable', overwrite = TRUE)

# Plot the response versus the vertex function
plotSimplexResponse(simplex)
```

`print.smplx` *Prints given simplex (a* `smplx` *class object)*

Description

Prints given simplex information.

Usage

```
## S3 method for class 'smplx'
print(x, extended = FALSE, conventions = TRUE, ...)
```

Arguments

`x`	simplex object to be printed
`extended`	logical, if `TRUE`, the object is printed as a list containing all hiden elements
`conventions`	logical, if `TRUE` (default), the conventions used are printed
`...`	other arguments passed to print

Author(s)

Cristhian Paredes, <craparedesca@unal.edu.co>

Jesús Ágreda, <jagreda@unal.edu.co>

`simplexExport` *Exports the information contained on a* `smplx` *class object.*

Description

Creates a `.smplx` file (text file) that stores the information contained in the simplex (a `smplx` class object, see `labsimplex`). It allows to continue with the optimization proccess after experiment described by a vertex has been carried and a quality function value has been obtained.

Usage

```
simplexExport(simplex, filename = NULL, direc = NULL)
```

Arguments

`simplex` `smplx` class object containing the simplex to be stored.

`filename` string with the name (without extention) of the file that will be created. If not provided, the name of the simplex object will be used.

`direc` directory in which the file will be saved. If not provided, the current directory will be used.

Value

A `.smplx` file containing all the information required to continue with the optimization process once all vertex experiments have been carried.

Author(s)

Cristhian Paredes, <`craparedesca@unal.edu.co`>

Jesús Ágreda, <`jagreda@unal.edu.co`>

Examples

```
## Not run:
  simplex <- labsimplex(N = 5, qual.fun = rnorm(6, 2, 1))
  simplexExport(simplex = simplex)

## End(Not run)
```

simplexImport *Imports the information contained in a* `.smplx` *file.*

Description

Imports the information contained in a `.smplx` file and creates a `'smplx'` class object.

Usage

```
simplexImport(filename, auth.ld = TRUE, name = NULL)
```

Arguments

`filename`	string with the name of the file (with extension) to be imported. The file must be generated using `simplexExport`. It may include path if the file is not in the current directory.
`auth.ld`	logical. Should the imported simplex object be directly loaded on the Environment? Default to TRUE.
`name`	name for the simplex object to be created if `auth.ld` = `FALSE`. When not provided, the ID in the file will be used.

Value

A `smplx` class object with the complete information of the simplex

Author(s)

Cristhian Paredes, <`craparedesca@unal.edu.co`>

Jesús Ágreda, <`jagreda@unal.edu.co`>

Examples

```
## Not run:
  simplexExport(labsimplex(N = 4), name = "simplex4D")
  simplexImport("simplex4D.smplx")

## End(Not run)
```

Index