Package 'fitscape'

March 1, 2022

Type Package

Title Classes for Fitness Landscapes and Seascapes

Version 0.1.0

Description Convenient classes to model fitness landscapes and fitness seascapes. A low-level package with which most users will not interact but upon which other packages modeling fitness landscapes and fitness seascapes will depend.

License MIT + file LICENSE

Encoding UTF-8

URL https://github.com/rrrlw/fitscape

BugReports https://github.com/rrrlw/fitscape/issues

RoxygenNote 7.1.1

Imports stats

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Raoul Wadhwa [aut, cre] (<https://orcid.org/0000-0003-0503-9580>), Jacob Scott [aut] (<https://orcid.org/0000-0003-2971-7673>)

Maintainer Raoul Wadhwa <raoulwadhwa@gmail.com>

Repository CRAN

Date/Publication 2022-03-01 09:10:01 UTC

R topics documented:

dims *Get Dimensions of Fitness Landscape*

Description

Get Dimensions of Fitness Landscape

Usage

```
dims(x)
```

Arguments

x	FitLandDF object

Value

integer vector analogous to ‘base::dim‘

Examples

```
# create flat fitness landscape with dimensions 3x3x3
values <- array(0, dim = rep(3, 3))
my_landscape <- FitLandDF(values)

# print dimensions
dims(my_landscape)
```

extract_df *Extract Data Frame Representation of Fitness Landscape*

Description

Extract Data Frame Representation of Fitness Landscape

Usage

```
extract_df(x)
```

Arguments

x	FitLandDF object

Value

data frame representation of fitness landscape

Examples

```
# create fitness landscape
values <- array(1:27, dim = rep(3, 3))
my_landscape <- FitLandDF(values)

# extact data frame representation
my_df <- extract_df(my_landscape)
```

`FitLandDF` *Create New FitLandDF Instance*

Description

Create New FitLandDF Instance

Usage

```
FitLandDF(scape_data, dims = dim(scape_data))
```

Arguments

`scape_data`	either data.frame or array object
`dims`	integer vector containing dimensions

Value

FitLandDF object

Examples

```
# create a flat fitness landscape with 3 binary (values 1 and 2) dimensions
values <- array(2, dim = rep(2, 3))

my_landscape <- FitLandDF(values)

# create a 2x2 fitness landscape that's highest when both dimensions are at 1
vals <- 1:2
df <- expand.grid(vals, vals)
df$Landscape_value <- c(1, 2, 3, 6)

my_landscape <- FitLandDF(df, dims = c(2L, 2L))
```

isFitLandDF *Confirm Object is Valid Instance of FitLandDF*

Description

Confirm Object is Valid Instance of FitLandDF

Usage

```
is.FitLandDF(x)

is_FitLandDF(x)
```

Arguments

x	object whose class is in question

Value

'logical'; 'TRUE' if 'x' is an instance of FitLandDF, 'FALSE' otherwise

minmax *Get Highest and Lowest Fitness Values from Fitness Landscape*

Description

Get Highest and Lowest Fitness Values from Fitness Landscape

Usage

```
min_fit(x)

max_fit(x)
```

Arguments

x	FitLandDF object

Value

minimum or maximum fitness value in this landscape

Examples

```
# create fitness landscape with min value 1 and max value 27
values <- array(1:27, dim = rep(3, 3))
my_landscape <- FitLandDF(values)

# calculate maximum fitness value
max_fit(my_landscape)

# calculate minimum fitness value
min_fit(my_landscape)
```

---

sdvar *Get Standard Deviation/Variance of Values in Fitness Landscape*

Description

Get Standard Deviation/Variance of Values in Fitness Landscape

Usage

```
variance(x, ...)

sdev(x, ...)
```

Arguments

x	FitLandDF object
...	additional parameters (e.g. 'na.rm')

Value

variance or standard deviation of values in fitness landscape

Examples

```
# create fitness landscape with non-zero variance and standard deviation
values <- array(1:27, dim = rep(3, 3))
my_landscape <- FitLandDF(values)

# calculate variance
variance(my_landscape)

# calculate standard deviation
sdev(my_landscape)
```

Index