

Package ‘kindisperse’

March 31, 2021

Title Simulate and Estimate Close-Kin Dispersal Kernels

Version 0.9.2

Description Functions for simulating and estimating kinship-related dispersal. Based on the methods described in M. Jasper, T.L. Schmidt., N.W. Ahmad, S.P. Sinkins & A.A. Hoffmann (2019) <doi:10.1111/1755-0998.13043> ``A genomic approach to inferring kinship reveals limited intergenerational dispersal in the yellow fever mosquito". Assumes an additive variance model of dispersal in two dimensions, compatible with Wright's neighbourhood area. Simple and composite dispersal simulations are supplied, as well as the functions needed to estimate parent-offspring dispersal for simulated or empirical data, and to undertake sampling design for future field studies of dispersal. For ease of use an integrated Shiny app is also included.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

URL <https://github.com/moshejasper/kindisperse>

RoxygenNote 7.1.1

Suggests testthat, knitr, rmarkdown

Imports ggplot2, readr, shiny, shinythemes, ggrepel, fitdistrplus, LaplacesDemon, here, stats, methods, tibble, grid, magrittr, plotly, dplyr, rlang, tidyselect, stringr

VignetteBuilder knitr

Depends R (>= 2.10)

NeedsCompilation no

Author Moshe-Elijah Jasper [aut, cre]
(<<https://orcid.org/0000-0003-4541-3223>>)

Maintainer Moshe-Elijah Jasper <moshe.jasper@unimelb.edu.au>

Repository CRAN

Date/Publication 2021-03-31 09:10:02 UTC

R topics documented:

| | |
|--------------------------------------|----|
| axials | 3 |
| axials_add | 4 |
| axials_combine | 5 |
| axials_decompose | 5 |
| axials_standard | 6 |
| axials_subtract | 8 |
| axpermute | 9 |
| axpermute_standard | 10 |
| axpermute_subtract | 12 |
| breedsigma<- | 13 |
| check_valid_kinship | 13 |
| check_valid_lifestage | 14 |
| csv_to_kinpair | 14 |
| df_to_kinpair | 15 |
| display_appdata | 15 |
| distances | 16 |
| distances,KinPairData-method | 16 |
| elongate | 18 |
| filtertype | 18 |
| filtertype<- | 19 |
| gravsigma<- | 19 |
| initsigma<- | 20 |
| is.KinPairData | 20 |
| is.KinPairSimulation | 21 |
| kernelshape | 21 |
| kernelshape,KinPairSimulation-method | 22 |
| kerneltype | 26 |
| kerneltype<- | 26 |
| KinPairData | 27 |
| KinPairSimulation | 28 |
| KinPairSimulation_composite | 30 |
| KinPairSimulation_simple | 32 |
| kinpair_to_csv | 33 |
| kinpair_to_tibble | 33 |
| kinpair_to_tsv | 34 |
| kinship | 34 |
| lifestage | 35 |
| mentari | 35 |
| mount_appdata | 36 |
| ovisigma<- | 37 |
| posigma | 37 |
| posigma<- | 38 |
| read_kindata | 38 |
| rebase_dims | 39 |
| reset_appdata | 39 |
| reset_tempdata | 40 |

| | |
|--------------------------------------|----|
| retrieveall_appdata | 40 |
| retrieve_appdata | 41 |
| retrieve_tempdata | 41 |
| run_kindisperse | 42 |
| sample_kindist | 42 |
| simdims | 43 |
| simdims<- | 43 |
| simgraph_data | 44 |
| simgraph_graph | 44 |
| simtype | 46 |
| simtype<- | 46 |
| simulate_kindist_composite | 47 |
| simulate_kindist_simple | 48 |
| tsv_to_kinpair | 49 |
| unmount_appdata | 49 |
| upper | 50 |
| vector_to_kinpair | 51 |
| write_kindata | 51 |

Index 53

| | |
|--------|--|
| axials | <i>Estimate the axial dispersal distance of a kernel</i> |
|--------|--|

Description

Estimate the axial dispersal distance of a kernel

Usage

```
axials(valvect, composite = 1)
```

Arguments

| | |
|-----------|---|
| valvect | A numeric vector of distances between close kin OR an object of class KinPair-Data |
| composite | numeric. The number of separate 'draws' (dispersal events) from the kernel required to produce the final positions of the measured individuals. For example, the displacement of a child from parent at the same lifestage would involve 1 draw and thus be composite = 1. Two full siblings would be two draws (composite = 2) from the FS kernel. Non-symmetric relationships (e.g. AV, 1C) should not be decomposed using this method, nor should any assumptions be made about different kernels (e.g. the 1C relationship would appropriately be given the value 2, but not 4) |

Value

Returns the value of the estimated axial dispersal distance of the kernel producing the dispersal distances measured. (numeric)

Examples

```
po_dists <- c(5, 6, 7.5)
axials(po_dists) # one 'draw' (dispersal event) goes into the parent offspring category
# so composite is left to its default of 1

fs_dists <- c(2, 3, 3)
axials(fs_dists, composite = 2) # two 'draws' (symmetric dispersal events)
# go into the full sibling category so composite is set to 2
```

axials_add

Add axial distributions

Description

Add axial distributions. Useful to construct an overall distribution that results from multiple 'draws' from smaller distributions. E.g. The pathway between first cousins which can be found by adding each of the component distributions of their respective lifespans along with the relevant offspring producing (e.g. oviposition) of the parent.

Usage

```
axials_add(axvals)
```

Arguments

axvals numeric. vector of axial distribution values from different kernels that are to be added.

Value

numeric Returns the axial value that results from adding the input axial values under an additive variance framework.

Examples

```
fullsibs_ax <- 5
parent_offspring_ax <- 25
cousin_ax <- axials_add(c(fullsibs_ax, parent_offspring_ax))
```

| | |
|----------------|--|
| axials_combine | <i>Combine axial distributions to produce a mixed distribution</i> |
|----------------|--|

Description

combines axial distributions to produce a mixed distribution. This is useful in settings where you have two separate distributions (e.g. FS & HS) with their own axial values, but you want to average them appropriately so that they can be compared to e.g. a mixed distribution of full & half cousins which cannot be distinguished via kinship determination methods and thus are best treated as an even mixture of the two categories. Different to adding dispersal events.

Usage

```
axials_combine(axvals)
```

Arguments

| | |
|--------|---|
| axvals | numeric. vector of axial distribution values from different kernels that are to be combined |
|--------|---|

Value

numeric Returns the axial value that results from combining the input axial values under an additive variance framework.

Examples

```
fullax <- axials(c(2, 4, 5), composite = 2)
halfax <- axials(c(6, 5, 7), composite = 2)
sibax <- axials_combine(c(fullax, halfax))
```

| | |
|------------------|---|
| axials_decompose | <i>Decompose an axial distribution into simple components</i> |
|------------------|---|

Description

Decomposes an axial distribution into simple components. Note that this should only be used in the simplest situations.

Usage

```
axials_decompose(ax, n_composites = 2)
```

Arguments

`ax` numeric. The axial value to be decomposed.

`n_composites` numeric. The number of separate 'draws' (dispersal events) from the kernel required to produce the final positions of the measured individuals. For example, the displacement of a child from parent at the same life stage would involve 1 draw and thus be composite = 1. Two full siblings would be two draws (composite = 2) from the FS kernel. Non-symmetric relationships (e.g. AV, 1C) should not be decomposed using this method, nor should any assumptions be made about different kernels (e.g. the 1C relationship would appropriately be given the value 2, but not 4)

Value

Returns the (numeric) axial distribution value of the underlying dispersal kernel from which the composite kernel was (or could be) created.

Examples

```
fs_vect <- c(10, 11, 12)
fs_axial_raw <- axials(fs_vect, composite = 1) # composite hasn't corrected for two dispersal events
# inherent to this kin category!
fs_axial_final <- axials_decompose(fs_axial_raw, n_composites = 2)
```

| | |
|------------------------------|--|
| <code>axials_standard</code> | <i>Takes (at least) two vectors of kinship dispersal distances from defined kinship categories, and returns a resulting calculation of the parent-offspring (intergenerational) kinship dispersal kernel Further tests</i> |
|------------------------------|--|

Description

Takes (at least) two vectors of kinship dispersal distances from defined kinship categories, and returns a resulting calculation of the parent-offspring (intergenerational) kinship dispersal kernel Further tests

Usage

```
axials_standard(
  avect,
  bvect,
  acat = NULL,
  bcat = NULL,
  amix = FALSE,
  bmix = FALSE,
  amixcat = NULL,
  bmixcat = NULL,
  acomp = FALSE,
  bcomp = FALSE,
```

```

    acompvect = NULL,
    bcompvect = NULL,
    acompcat = NULL,
    bcompcat = NULL
)

```

Arguments

| | |
|-----------|---|
| avect | vector a of kin dispersal distances for the less closely related kinship category OR object of class KinPairData. |
| bvect | vector b of kin dispersal distances for the more closely related kinship category OR object of class KinPairData. |
| acat | kinship category of kin dispersal vector avect. Must be one of "PO", "FS", "HS", "AV", "GG", "HAV", "GGG", "1C", "1C1", "2C", "GAV", "HGAV", "H1C", "H1C1", "H2C" |
| bcac | kinship category of kin dispersal vector bvect. Must be one of "PO", "FS", "HS", "AV", "GG", "HAV", "GGG", "1C", "1C1", "2C", "GAV", "HGAV", "H1C", "H1C1", "H2C" |
| amix | logical describing whether vector a is a mixture of two kinship categories. Used with amixcat. Default FALSE. |
| bmix | logical describing whether vector b is a mixture of two kinship categories. Used with bmixcat. Default FALSE. |
| amixcat | mixture kinship category of vector a. Must be set if amix == TRUE. Must be one of "PO", "FS", "HS", "AV", "GG", "HAV", "GGG", "1C", "1C1", "2C", "GAV", "HGAV", "H1C", "H1C1", "H2C" |
| bmixcat | mixture kinship category of vector b. Must be set if bmix == TRUE. Must be one of "PO", "FS", "HS", "AV", "GG", "HAV", "GGG", "1C", "1C1", "2C", "GAV", "HGAV", "H1C", "H1C1", "H2C" |
| acomp | logical denoting whether vector a should be composited with an additional kinship category vector. Used with acompvect and acompcat. Default FALSE. |
| bcomp | logical denoting whether vector b should be composited with an additional kinship category vector. Used with bcompvect and bcompcat. Default FALSE. |
| acompvect | vector acomp of kin dispersal distances for compositing with vector a OR object of class KinPairData. Must be set if acomp == TRUE. |
| bcompvect | vector bcomp of kin dispersal distances for compositing with vector b OR object of class KinPairData. Must be set if bcomp == TRUE. |
| acompcat | kinship category of kin dispersal vector acompvect. Must be set if acomp == TRUE. Must be one of "PO", "FS", "HS", "AV", "GG", "HAV", "GGG", "1C", "1C1", "2C", "GAV", "HGAV", "H1C", "H1C1", "H2C" |
| bcompcat | kinship category of kin dispersal vector bcompvect. Must be set if bcomp == TRUE. Must be one of "PO", "FS", "HS", "AV", "GG", "HAV", "GGG", "1C", "1C1", "2C", "GAV", "HGAV", "H1C", "H1C1", "H2C" |

Value

Returns a numeric estimate of PO (intergenerational) dispersal kernel axial distribution.

Examples

```
cous <- rexp(100, 1 / 100)
fullsibs <- rexp(50, 1 / 50)
axials_standard(cous, fullsibs, acat = "1C", bcat = "FS")
```

| | |
|-----------------|-------------------------------------|
| axials_subtract | <i>Subtract axial distributions</i> |
|-----------------|-------------------------------------|

Description

Subtract axial distributions, finding the difference (under an additive variance framework). This is most useful when one distribution subsumes another and includes a unique dispersal event that needs to be extracted. For example, the FS category is subsumed by the 1C category, which can be written 'FS + PO'. In this circumstance, subtracting FS from 1C will yield an estimate of the PO kernel (the basic intergenerational dispersal kernel)

Usage

```
axials_subtract(abig, asmall)
```

Arguments

| | |
|--------|---|
| abig | numeric. The axial dispersal distance of the larger (subsuming) distribution (e.g. 1C). |
| asmall | numeric. The axial dispersal distance of the smaller (subsumed) distribution (e.g. FS). |

Value

numeric Returns an estimate of the axial dispersal distance of those dispersal elements that are unique to the larger dispersal distribution (e.g. PO).

Examples

```
axials_subtract(100, 70)
```

| | |
|-----------|--|
| axpermute | <i>Estimate the axial dispersal distance of a kernel with confidence intervals</i> |
|-----------|--|

Description

Estimate the axial dispersal distance of a kernel with confidence intervals

Usage

```
axpermute(vals, nreps = 1000, nsamp = "std", composite = 1, output = "confs")
```

Arguments

| | |
|-----------|---|
| vals | numeric. Vector of distances between close kin OR object of class KinPairData. |
| nreps | numeric. Number of permutations to run for confidence intervals (default 1000) |
| nsamp | numeric. Number of kin pairs to subsample for each permutation. Either "std" or an integer. If "std" will be computed as equal to the sample size. (default "std") |
| composite | numeric. The number of separate 'draws' (dispersal events) from the kernel required to produce the final positions of the measured individuals. For example, the displacement of a child from parent at the same lifestage would involve 1 draw and thus be composite = 1. Two full siblings would be two draws (composite = 2) from the FS kernel. Non-symmetric relationships (e.g. AV, 1C) should not be decomposed using this method, nor should any assumptions be made about different kernels (e.g. the 1C relationship would appropriately be given the value 2, but not 4) |
| output | character. Denotes what kind of output to return. If 'confs', a vector of 95% confidence intervals. if 'vect', a vector of all permuted axial value results |

Value

If output = 'confs', returns a numeric vector of 95% confidence intervals and mean axial value. if output = 'vect', returns a numeric vector of all permuted axial value results

Examples

```
po_dists <- rexp(100, 1 / 50)
axpermute(po_dists, composite = 1)
```

| | |
|--------------------|---|
| axpermute_standard | <i>Takes (at least) two vectors of kinship dispersal distances from defined kinship categories, and returns a resulting calculation of the parent-offspring (intergenerational) kinship dispersal kernel with bootstrap-based confidence intervals. Further tests</i> |
|--------------------|---|

Description

Takes (at least) two vectors of kinship dispersal distances from defined kinship categories, and returns a resulting calculation of the parent-offspring (intergenerational) kinship dispersal kernel with bootstrap-based confidence intervals. Further tests

Usage

```
axpermute_standard(
  avect = NULL,
  bvect = NULL,
  acat = NULL,
  bcat = NULL,
  nreps = 1000,
  nsamp = "std",
  amix = FALSE,
  bmix = FALSE,
  amixcat = NULL,
  bmixcat = NULL,
  acomp = FALSE,
  bcomp = FALSE,
  acompvect = NULL,
  bcompvect = NULL,
  acompcat = NULL,
  bcompcat = NULL,
  output = "confs"
)
```

Arguments

| | |
|-------|---|
| avect | vector a of kin dispersal distances for the less closely related kinship category OR object of class KinPairData. |
| bvect | vector b of kin dispersal distances for the more closely related kinship category OR object of class KinPairData. |
| acat | kinship category of kin dispersal vector avect. Must be one of "PO", "FS", "HS", "AV", "GG", "HAV", "GGG", "1C", "1C1", "2C", "GAV", "HGAV", "H1C", "H1C1", "H2C" |
| bcat | kinship category of kin dispersal vector bvect. Must be one of "PO", "FS", "HS", "AV", "GG", "HAV", "GGG", "1C", "1C1", "2C", "GAV", "HGAV", "H1C", "H1C1", "H2C" |

| | |
|-----------|---|
| nreps | number of permutations to run for confidence intervals (default 1000) |
| nsamp | number of kin pairs to subsample for each permutation. Either "std" or an integer. If "std" will be computed as equal to the sample size. (default "std") |
| amix | logical describing whether vector a is a mixture of two kinship categories. Used with amixcat. Default FALSE. |
| bmix | logical describing whether vector b is a mixture of two kinship categories. Used with bmixcat. Default FALSE. |
| amixcat | mixture kinship category of vector a. Must be set if amix == TRUE. Must be one of "PO", "FS", "HS", "AV", "GG", "HAV", "GGG", "1C", "1C1", "2C", "GAV", "HGAV", "H1C", "H1C1", "H2C" |
| bmixcat | mixture kinship category of vector b. Must be set if bmix == TRUE. Must be one of "PO", "FS", "HS", "AV", "GG", "HAV", "GGG", "1C", "1C1", "2C", "GAV", "HGAV", "H1C", "H1C1", "H2C" |
| acomp | logical denoting whether vector a should be composited with an additional kinship category vector. Used with acompvect and acompcat. Default FALSE. |
| bcomp | logical denoting whether vector b should be composited with an additional kinship category vector. Used with bcompvect and bcompcat. Default FALSE. |
| acompvect | vector acomp of kin dispersal distances for compositing with vector a OR object of class KinPairData. Must be set if acomp == TRUE. |
| bcompvect | vector bcomp of kin dispersal distances for compositing with vector b OR object of class KinPairData. Must be set if bcomp == TRUE. |
| acompcat | kinship category of kin dispersal vector acompvect. Must be set if acomp == TRUE. Must be one of "PO", "FS", "HS", "AV", "GG", "HAV", "GGG", "1C", "1C1", "2C", "GAV", "HGAV", "H1C", "H1C1", "H2C" |
| bcompcat | kinship category of kin dispersal vector bcompvect. Must be set if bcomp == TRUE. Must be one of "PO", "FS", "HS", "AV", "GG", "HAV", "GGG", "1C", "1C1", "2C", "GAV", "HGAV", "H1C", "H1C1", "H2C" |
| output | string denoting what kind of output to return. If 'confs', a vector of 95% confidence intervals. if 'vect', a vector of all permuted axial value results |

Value

If output = 'confs' returns vector of 95% confidence intervals (with mean). If output = 'vect' returns vector of individual axial estimates from each permutation

Examples

```
cous <- rexp(100, 1 / 100)
fullsibs <- rexp(50, 1 / 50)
axpermute_standard(cous, fullsibs, acat = "1C", bcat = "FS")
```

axpermute_subtract *Subtract axial distributions with confidence intervals*

Description

Find the difference between two different empirical axial distributions with confidence intervals. This is most useful when one distribution subsumes another and includes a unique dispersal event that needs to be extracted. For example, the FS category is subsumed by the 1C category, which can be written 'FS + PO'. In this circumstance, subtracting FS from 1C will yield an estimate of the PO kernel (the basic intergenerational dispersal kernel)

Usage

```
axpermute_subtract(
  bigvals,
  smallvals,
  nreps = 1000,
  nsamp = "std",
  composite = 2,
  output = "confs"
)
```

Arguments

| | |
|-----------|---|
| bigvals | numeric. Vector of distance distributions of the larger (subsuming) distribution (e.g. 1C) OR object of class KinPairData. |
| smallvals | numeric. Vector of distance distributions of the smaller (subsumed) distribution (e.g. FS) OR object of class KinPairData. |
| nreps | numeric. Number of permutations to perform when generating confidence intervals. |
| nsamp | numeric. number of kin pairs to subsample for each permutation. Either "std" or an integer. If "std" will be computed as equal to the sample size. (default "std") |
| composite | numeric. The number of separate 'draws' (dispersal events) from the kernel required to produce the final positions of the measured individuals. For example, the displacement of a child from parent at the same lifestage would involve 1 draw and thus be composite = 1. Two full siblings would be two draws (composite = 2) from the FS kernel. Non-symmetric relationships (e.g. AV, 1C) should not be decomposed using this method, nor should any assumptions be made about different kernels (e.g. the 1C relationship would appropriately be given the value 2, but not 4) |
| output | character. What kind of output to return. Either 'confs' (default -> confidence intervals) or 'vect' -> vector of axial distances |

Value

If output = 'confs' returns numeric vector of 95% confidence intervals and mean axial value. If output = 'vect' returns numeric vector of individual axial estimates from each permutation

Examples

```

firstcous <- rexp(100, 1 / 80)
fullsibs <- rexp(100, 1 / 50)
apermute_subtract(firstcous, fullsibs)

```

| | |
|--------------|--------------|
| breedsigma<- | <i>Title</i> |
|--------------|--------------|

Description

Title

Usage

```
breedsigma(x) <- value
```

Arguments

| | |
|-------|-----------------------------|
| x | object with relevant method |
| value | new value to assign |

Value

returns a modified object of the relevant class

| | |
|---------------------|---|
| check_valid_kinship | <i>Check if vector of kinship categories contains all valid entries</i> |
|---------------------|---|

Description

Check if vector of kinship categories contains all valid entries

Usage

```
check_valid_kinship(vect)
```

Arguments

| | |
|------|------------------------------|
| vect | vector of kinship categories |
|------|------------------------------|

Value

TRUE if valid. Error otherwise.

`check_valid_lifestage` *Check if vector of lifestages contains all valid entries*

Description

Check if vector of lifestages contains all valid entries

Usage

```
check_valid_lifestage(vect)
```

Arguments

`vect` vector of lifestages

Value

TRUE if valid. Error otherwise

`csv_to_kinpair` *Reads .csv and converts to KinPairData object*

Description

Reads .csv and converts to KinPairData object

Usage

```
csv_to_kinpair(file, kinship = NULL, lifestage = NULL, ...)
```

Arguments

`file` The file path to read from

`kinship` character. kin category to assign or extract from data. one of PO, FS, HS, AV, GG, HAV, GGG, 1C, 1C1, 2C, GAV, HGAV, H1C , H1C1 or H2C

`lifestage` character. lifestage to assign or extract from data. one of 'unknown', 'immature' or 'ovipositional'.

`...` additional arguments to pass to `read_csv`

Value

returns an object of class KinPairData

| | |
|---------------|---|
| df_to_kinpair | <i>Convert dataframe or tibble to KinPairData class</i> |
|---------------|---|

Description

Convert dataframe or tibble to KinPairData class

Usage

```
df_to_kinpair(data, kinship = NULL, lifestage = NULL)
```

Arguments

| | |
|-----------|--|
| data | data.frame or tibble of kin distances - can contain \$distance (kin distances), \$kinship (kin cats) & \$lifestage columns |
| kinship | character. kin category to assign or extract from data. one of PO, FS, HS, AV, GG, HAV, GGG, 1C, 1C1, 2C, GAV, HGAV, H1C , H1C1 or H2C |
| lifestage | character. lifestage to assign or extract from data. one of 'unknown', 'immature' or 'ovipositional'. |

Value

returns valid KinPairData object

Examples

```
mydata <- tibble::tibble(
  distance = 1:10, lifestage = "immature",
  kinship = c("FS", "FS", "FS", "FS", "FS", "FS", "HS", "HS", "HS", "HS")
)
df_to_kinpair(mydata, kinship = "FS")
```

| | |
|-----------------|--|
| display_appdata | <i>Show printout of named items stored in appdata.</i> |
|-----------------|--|

Description

Show printout of named items stored in appdata.

Usage

```
display_appdata()
```

Value

No return value, called for side effects

Examples

```

mount_appdata(KinPairData(), "my_kindata")
mount_appdata(simulate_kindist_simple(nsim = 10), "my_simdata")

display_appdata()

```

| | |
|-----------|---|
| distances | <i>Access distances (generic for KinPairData class)</i> |
|-----------|---|

Description

Access distances (generic for KinPairData class)

Usage

```
distances(x)
```

Arguments

x Object of Class KinPairData

Value

Returns a numeric vector of kin separation distances

| | |
|-------------------------------|--|
| distances, KinPairData-method | <i>Constructor method for KinPairData objects.</i> |
|-------------------------------|--|

Description

Constructor method for KinPairData objects.

Usage

```

## S4 method for signature 'KinPairData'
distances(x)

## S4 method for signature 'KinPairData'
kinship(x)

## S4 replacement method for signature 'KinPairData'
kinship(x) <- value

## S4 method for signature 'KinPairData'
lifestage(x)

```



```
## S4 replacement method for signature 'KinPairData'
lifestage(x) <- value

## S4 method for signature 'KinPairData'
show(object)

## S4 method for signature 'KinPairData'
initialize(.Object, data = NULL, kinship = NULL, lifestage = NULL, ...)
```

Arguments

| | |
|-------------|--|
| x | object of class KinPairData |
| value | value to assign to slot |
| object | an object of class KinpairData |
| .Object | the KinPairData object to be constructed |
| data | data about kinship to be used to construct object (tibble, data.frame, or numeric vector of distances) |
| kinship | character. Kinship category value for object. - one of PO, FS, HS, AV, HAV, GG, 1C, H1C, GAV, HGAV, 1C1, H1C1, GGG, 2C, and H2C. |
| lifestage | character. Lifestage value for object. - one of 'immature', 'ovipositional' or 'unknown' |
| ... | additional argument to pass to downstream functions in future |
| KinPairData | object of class KinPairData |

Value

numeric vector of kin separation distances
 character kinship of KinPairData object
 modified object of class KinPairData
 character lifestage of KinPairData object
 modified object of class KinPairData
 No return value, called for side effects
 Returns an object of class KinPairData

Methods (by generic)

- distances: access distances
- kinship: access kin category
- kinship<-: assign kin category
- lifestage: access lifestage
- lifestage<-: assign lifestage
- show: standard print method
- initialize: initialize method

elongate *Change the shape (aspect ratio) of a rectangle while preserving area*

Description

Change the shape (aspect ratio) of a rectangle while preserving area

Usage

```
elongate(dims, aspect = 1)
```

Arguments

dims Original rectangle dimensions - either single number (length of side of square) or length 2 numeric vector (lengths of sides x and y of rectangle)

aspect Aspect ratio of side lengths x & y (i.e. x/y) in the new rectangle

Value

Returns a numeric vector containing the side lengths c(x, y) of a transformed rectangle with preserved area

Examples

```
elongate(10, 100)
elongate(c(5, 125), 4)
```

filtertype *Access filter type of KinPairSimulation object*

Description

Access filter type of KinPairSimulation object

Usage

```
filtertype(x)
```

Arguments

x object of class KinPairSimulation

Value

character filter status of simulation

filtertype<- *Title*

Description

Title

Usage

`filtertype(x) <- value`

Arguments

x object of class KinPairSimulation
value new value to assign

Value

returns a modified object of the relevant class

gravsigma<- *Title*

Description

Title

Usage

`gravsigma(x) <- value`

Arguments

x object of class KinPairSimulation
value new value to assign

Value

returns a modified object of the relevant class

`initsigma<-` *Title*

Description

Title

Usage

```
initsigma(x) <- value
```

Arguments

`x` object of class `KinPairSimulation`
`value` new value to assign

Value

returns a modified object of the relevant class

`is.KinPairData` *Check if object is of class KinPairData*

Description

Check if object is of class `KinPairData`

Usage

```
is.KinPairData(x)
```

Arguments

`x` object to be checked

Value

Returns TRUE if of class `KinPairData`, FALSE if not.

`is.KinPairSimulation` *Check if object is of class KinPairSimulation*

Description

Check if object is of class KinPairSimulation

Usage

`is.KinPairSimulation(x)`

Arguments

x object to be checked

Value

Returns TRUE if of class KinPairSimulation, FALSE if not

`kernelshape` *kindisperse - access kerneltype of KinPairSimulation object*

Description

kindisperse - access kerneltype of KinPairSimulation object

Usage

`kernelshape(x)`

Arguments

x object of class KinPairSimulation

Value

character the shape parameter used in kernel simulation (if kerneltype is vgamma)

kernelshape, KinPairSimulation-method
)

Description

)

Usage

```
## S4 method for signature 'KinPairSimulation'  
kernelshape(x)  
  
## S4 method for signature 'KinPairSimulation'  
simtype(x)  
  
## S4 method for signature 'KinPairSimulation'  
kerneltype(x)  
  
## S4 method for signature 'KinPairSimulation'  
posigma(x)  
  
## S4 method for signature 'KinPairSimulation'  
initsigma(x)  
  
## S4 method for signature 'KinPairSimulation'  
breedsigma(x)  
  
## S4 method for signature 'KinPairSimulation'  
gravsigma(x)  
  
## S4 method for signature 'KinPairSimulation'  
ovisigma(x)  
  
## S4 method for signature 'KinPairSimulation'  
sindims(x)  
  
## S4 method for signature 'KinPairSimulation'  
filtertype(x)  
  
## S4 method for signature 'KinPairSimulation'  
upper(x)  
  
## S4 method for signature 'KinPairSimulation'  
lower(x)  
  
## S4 method for signature 'KinPairSimulation'
```

```
spacing(x)

## S4 method for signature 'KinPairSimulation'
samplenum(x)

## S4 method for signature 'KinPairSimulation'
sampledims(x)

## S4 replacement method for signature 'KinPairSimulation'
upper(x) <- value

## S4 replacement method for signature 'KinPairSimulation'
lower(x) <- value

## S4 replacement method for signature 'KinPairSimulation'
spacing(x) <- value

## S4 replacement method for signature 'KinPairSimulation'
samplenum(x) <- value

## S4 replacement method for signature 'KinPairSimulation'
sampledims(x) <- value

## S4 method for signature 'KinPairSimulation'
show(object)

## S4 method for signature 'KinPairSimulation'
initialize(
  .Object,
  data = NULL,
  kinship = NULL,
  lifestage = NULL,
  simtype = NULL,
  kerneltype = NULL,
  kernelshape = NULL,
  posigma = NULL,
  initsigma = NULL,
  breedsigma = NULL,
  gravsigma = NULL,
  ovisigma = NULL,
  simdims = NULL,
  call = NULL,
  filtertype = NULL,
  upper = NULL,
  lower = NULL,
  spacing = NULL,
  samplenum = NULL,
  sampledims = NULL
```

)

Arguments

| | |
|-------------------|---|
| x | object of class KinPairSimulation |
| value | value for parameter to be adjusted to |
| object | object of class KinPairSimulation |
| .Object | object to be constructed into KinPairSimulation class |
| data | tbl_df. tibble of simulation values |
| kinship | character - one of PO, FS, HS, AV, HAV, GG, 1C, H1C, GAV, HGAV, 1C1, H1C1, GGG, 2C, and H2C. |
| lifestage | character - one of 'unknown', 'immature' or 'ovipositional' |
| simtype | character - simulation type |
| kerneltype | character. - 'Gaussian', 'Laplace' or 'vgamma' (variance-gamma) |
| kernelshape | numeric. - value of kernel shape of simulation (if using kernel with shape parameter e.g. vgamma) |
| posigma | numeric - overall value of dispersal sigma (for simple kernel) |
| initsigma | numeric. - value of pre-breeding dispersal sigma (for composite kernel) |
| breedsigma | numeric. - value of breeding dispersal sigma (for composite kernel) |
| gravsigma | numeric. - value of post-breeding dispersal sigma (for composite kernel) |
| ovisigma | numeric. - value of oviposition dispersal sigma (for composite kernel) |
| simdims | numeric. - dimensions of sampling area (assumes one side of square) |
| call | call. Call to create object |
| filtertype | character. whether the initial sim has been further filtered |
| upper | numeric. - FILTER: upper threshold used |
| lower | numeric. - FILTER: lower threshold used |
| spacing | numeric. - FILTER: spacing used |
| samplenum | numeric. - FILTER: sample number used |
| sampldims | numeric. - FILTER: sample dimensions used |
| KinPairSimulation | an object of class KinPairSimulation |

Value

character the shape parameter used in kernel simulation (if kerneltype is vgamma)
 character the kind of simulation stored in the object (simple or composite)
 character the type of statistical kernel used to run the simulation (Gaussian, Laplace, vgamma)
 numeric posigma value of simple simulation
 numeric initsigma value of composite simulation
 numeric breedsigma value of composite simulation

numeric gravsigma value of composite simulation
numeric ovisigma value of composite simulation
numeric vector simulation dimensions of KinPairSimulation object
character filter status of KinPairSimulation object
numeric upper value of sampled KinPairSimulation object
numeric lower value of sampled KinPairSimulation object
numeric trap spacing value of sampled KinPairSimulation object
numeric number of kin dyads in KinPairSimulation object
numeric vector sampling dimensions of KinPairSimulation object
returns a modified object of class KinPairSimulation
returns a modified object of class KinPairSimulation
returns a modified object of class KinPairSimulation
returns a modified object of class KinPairSimulation
No return value, called for side effects
Returns an object of class KinPairSimulation

Methods (by generic)

- kernelshape: access kernelshape
- simtype: access simulation type
- kerneltype: access kerneltype
- posigma: access sigma
- initsigma: access initsigma
- breedsigma: access breedsigma
- gravsigma: access gravsigma
- ovisigma: access ovisigma
- simdims: access simdims
- filtertype: access filtertype
- upper: access upper filter distance
- lower: access lower filter distance
- spacing: access spacing
- samplenum: access sampled samplenum
- sampledims: access sampled dimensions
- upper<-: assign and filter by upper distance (uses sample_kindist())
- lower<-: assign and filter by lower distance (uses sample_kindist())
- spacing<-: assign kin spacing (uses sample_kindist())
- samplenum<-: assign and downsample to samplenum (uses sample_kindist())
- sampledims<-: assign and filter by sample dimensions (uses sample_kindist())
- show: print method
- initialize: initialisation method

| | |
|------------|--|
| kerneltype | <i>kindisperse - access kerneltype of KinPairSimulation object</i> |
|------------|--|

Description

kindisperse - access kerneltype of KinPairSimulation object

Usage

```
kerneltype(x)
```

Arguments

x object of class KinPairSimulation

Value

character the type of statistical kernel used to run the simulation (Gaussian, Laplace, vgamma)

| | |
|--------------|--------------|
| kerneltype<- | <i>Title</i> |
|--------------|--------------|

Description

Title

Usage

```
kerneltype(x) <- value
```

Arguments

x object of class KinPairSimulation
value new value to assign

Value

returns a modified object of the relevant class with altered kerneltype parameter

| | |
|-------------|-----------------------------------|
| KinPairData | <i>Formal class "KinPairData"</i> |
|-------------|-----------------------------------|

Description

The class KinPairData is a formal (S4) class for storing kinship and lifespan dispersal information concerning kin pairs.

Usage

```
KinPairData(data = NULL, kinship = NULL, lifestage = NULL)
```

```
KinPairData(data = NULL, kinship = NULL, lifestage = NULL)
```

Arguments

| | |
|-----------|--|
| data | tbl_df. Tibble of kinpair distances |
| kinship | character. - one of PO, FS, HS, AV, HAV, GG, 1C, H1C, GAV, HGAV, 1C1, H1C1, GGG, 2C, H2C & UN. |
| lifestage | character. - one of 'unknown', 'immature' or 'ovipositional' |

Value

returns object of class KinPairData

returns an object of class KinPairData

Slots

kinship character.

lifestage character.

tab tbl_df.

Examples

```
KinPairData()
```

KinPairSimulation *KinPairSimulation Class*

Description

KinPairSimulation Class

Usage

```
KinPairSimulation(  
  data = NULL,  
  kinship = NULL,  
  lifestage = NULL,  
  simtype = NULL,  
  kerneltype = NULL,  
  posigma = NULL,  
  initsigma = NULL,  
  breedsigma = NULL,  
  gravsigma = NULL,  
  ovisigma = NULL,  
  simdims = NULL,  
  kernelshape = NULL,  
  call = NULL,  
  filtertype = NULL,  
  upper = NULL,  
  lower = NULL,  
  spacing = NULL,  
  samplenum = NULL,  
  sampledims = NULL  
)
```

```
KinPairSimulation(  
  data = NULL,  
  kinship = NULL,  
  lifestage = NULL,  
  simtype = NULL,  
  kerneltype = NULL,  
  posigma = NULL,  
  initsigma = NULL,  
  breedsigma = NULL,  
  gravsigma = NULL,  
  ovisigma = NULL,  
  simdims = NULL,  
  kernelshape = NULL,  
  call = NULL,  
  filtertype = NULL,  
  upper = NULL,
```

```

    lower = NULL,
    spacing = NULL,
    samplenum = NULL,
    sampledims = NULL
  )

```

Arguments

| | |
|-------------|---|
| data | tbl_df. tibble of simulation values |
| kinship | character - one of PO, FS, HS, AV, HAV, GG, 1C, H1C, GAV, HGAV, 1C1, H1C1, GGG, 2C, and H2C. |
| lifestage | character - one of 'unknown', 'immature' or 'ovipositional' |
| simtype | character - simulation type |
| kerneltype | character. - 'Gaussian', 'Laplace' or 'vgamma' (variance-gamma) |
| posigma | numeric - overall value of dispersal sigma (for simple kernel) |
| initsigma | numeric. - value of pre-breeding dispersal sigma (for composite kernel) |
| breedsigma | numeric. - value of breeding dispersal sigma (for composite kernel) |
| gravsigma | numeric. - value of post-breeding dispersal sigma (for composite kernel) |
| ovisigma | numeric. - value of oviposition dispersal sigma (for composite kernel) |
| simdims | numeric. - dimensions of sampling area (assumes one side of square) |
| kernelshape | numeric. - value of kernel shape of simulation (if using kernel with shape parameter e.g. vgamma) |
| call | call. Call to create object |
| filtertype | character. whether the initial sim has been further filtered |
| upper | numeric. - FILTER: upper threshold used |
| lower | numeric. - FILTER: lower threshold used |
| spacing | numeric. - FILTER: spacing used |
| samplenum | numeric. - FILTER: sample number used |
| sampledims | numeric. - FILTER: sample dimensions used |

Value

returns object of class KinPairSimulation
 returns an object of class KinPairSimulation.

Slots

kinship character - one of PO, FS, HS, AV, HAV, GG, 1C, H1C, GAV, HGAV, 1C1, H1C1, GGG, 2C, and H2C.
 simtype character.
 kerneltype character. - 'Gaussian', 'Laplace' or 'vgamma' (variance-gamma)
 posigma numeric. - overall value of dispersal sigma (for simple kernel)

initsigma numeric. - value of pre-breeding dispersal sigma (for composite kernel)
 breedsigma numeric. - value of breeding dispersal sigma (for composite kernel)
 gravsigma numeric. - value of post-breeding dispersal sigma (for composite kernel)
 ovisigma numeric. - value of oviposition dispersal sigma (for composite kernel)
 simdims numeric. - dimensions of sampling area (assumes 1 side of square)
 lifestage character. - lifestage at sampling - either 'immature' or 'ovipositional'
 kernelshape numeric. - shape parameter if vgamma kerneltype
 call call. - call to create initial simulation
 tab tbl_df. - tibble of simulation values
 filtertype character. - whether the initial sim has been further filtered
 upper numeric. - FILTER: upper threshold used
 lower numeric. - FILTER: lower threshold used
 spacing numeric. - FILTER: spacing used
 samplenum numeric. - FILTER: sample number used
 sampledims numeric. - FILTER: dimensions used

Examples

```
KinPairSimulation()
```

```
KinPairSimulation_composite
  Constructor for KinPairSimulation Class (composite)
```

Description

Constructor for KinPairSimulation Class (composite)

Usage

```

KinPairSimulation_composite(
  data = NULL,
  kinship = NULL,
  kerneltype = NULL,
  initsigma = NULL,
  breedsigma = NULL,
  gravsigma = NULL,
  ovisigma = NULL,
  simdims = NULL,
  lifestage = NULL,
  kernelshape = NULL,
  call = NULL
)

```

Arguments

| | |
|-------------|--|
| data | tibble of pairwise kin classes & distances. Ideally contains fields id1 & id2 (chr) an distance (dbl) optionally includes coords (x1, y1, x2, y2), lifestage (ls1 & ls2), kinship (chr) and sims (dbl) |
| kinship | character. Code for kinship category of simulation. one of PO, FS, HS, AV, GG, HAV, GGG, 1C, 1C1, 2C, GAV, HGAV, H1C or H2C |
| kerneltype | character. Statistical model for simulated dispersal kernel. Currently either "Gaussian", "Laplace" or "vgamma" (variance-gamma). |
| initsigma | numeric. Axial sigma of prebreeding ('juvenile') dispersal kernel (axial standard deviation). |
| breedsigma | numeric. Axial sigma of breeding dispersal kernel (axial standard deviation). |
| gravsigma | numeric. Axial sigma of post-breeding ('gravid') dispersal kernel (axial standard deviation). |
| ovisigma | numeric. Axial sigma of oviposition dispersal kernel (axial standard deviation). |
| sindims | numeric. Length of side of simulated area square. |
| lifestage | character. Simulated lifestage of sampling. Either "immature" (sampled at hatching) or "ovipositional" (sampled as an adult during oviposition - essentially one lifespan later than 'immature') |
| kernelshape | numeric. Value of shape parameter for simulated kernel if kernel requires one (e.g. vgamma kernel). |
| call | call object. Use to pass the system call that led to the generation of this class. (via sys.call) |

Value

Returns a KinPairSimulation Class object with simtype set to 'composite' and relevant fields included.

Examples

```
kindata <- tibble::tibble(
  id1 = c("a", "b", "c"), id2 = c("x", "y", "z"),
  distance = c(50, 45, 65), kinship = c("1C", "1C", "1C")
)
KinPairSimulation_composite(kindata,
  kinship = "1C", kerneltype = "Gaussian",
  initsigma = 15, breedsigma = 25, gravsigma = 20, ovisigma = 10, lifestage = "immature"
)
```

 KinPairSimulation_simple

Constructor for KinPairSimulation Class (simple)

Description

Constructor for KinPairSimulation Class (simple)

Usage

```
KinPairSimulation_simple(
  data = NULL,
  kinship = NULL,
  kerneltype = NULL,
  posigma = NULL,
  simdims = NULL,
  lifestage = NULL,
  kernelshape = NULL,
  call = NULL
)
```

Arguments

| | |
|-------------|--|
| data | tibble of pairwise kin classes & distances. Ideally contains fields id1 & id2 (chr) an distance (dbl) optionally includes coords (x1, y1, x2, y2), lifestage (ls1 & ls2), kinship (chr) and sims (dbl) |
| kinship | character. Code for kinship category of simulation. one of PO, FS, HS, AV, GG, HAV, GGG, 1C, 1C1, 2C, GAV, HGAV, H1C or H2C |
| kerneltype | character. Statistical model for simulated dispersal kernel. Currently either "Gaussian", "Laplace" or "vgamma" (variance-gamma). |
| posigma | numeric. Axial sigma of dispersal kernel (axial standard deviation). |
| simdims | numeric. Length of side of simulated area square. |
| lifestage | character. Simulated lifestage of sampling. Either "immature" (sampled at hatching) or "ovipositional" (sampled as an adult during oviposition - essentially one lifespan later than 'immature') |
| kernelshape | numeric. Value of shape parameter for simulated kernel if kernel requires one (e.g. vgamma kernel). |
| call | call object. Use to pass the system call that led to the generation of this class. (via sys.call) |

Value

Returns a KinPairSimulation Class object with simtype set to 'simple' and relevant fields included.

Examples

```
kindata <- tibble::tibble(
  id1 = c("a", "b", "c"), id2 = c("x", "y", "z"),
  distance = c(50, 45, 65), kinship = c("1C", "1C", "1C")
)
KinPairSimulation_simple(kindata,
  kinship = "1C", kerneltype = "Gaussian",
  posigma = 38, lifestage = "immature"
)
```

| | |
|----------------|--|
| kinpair_to_csv | <i>Write KinPairData object in .csv format (strips extra data)</i> |
|----------------|--|

Description

Write KinPairData object in .csv format (strips extra data)

Usage

```
kinpair_to_csv(x, file, ...)
```

Arguments

| | |
|------|--|
| x | Object of class KinPairData or KinPairSimulation |
| file | The file path to write to |
| ... | Additional arguments to pass to write_csv |

Value

Invisibly returns the initial object

| | |
|-------------------|---|
| kinpair_to_tibble | <i>Extract KinPairData class object to tibble</i> |
|-------------------|---|

Description

Extract KinPairData class object to tibble

Usage

```
kinpair_to_tibble(x)
```

Arguments

| | |
|---|-----------------------------|
| x | object of class KinPairData |
|---|-----------------------------|

Value

tibble (class tbl_df)

| | |
|----------------|--|
| kinpair_to_tsv | <i>Write KinPairData object in .tsv format (strips extra data)</i> |
|----------------|--|

Description

Write KinPairData object in .tsv format (strips extra data)

Usage

```
kinpair_to_tsv(x, file, ...)
```

Arguments

| | |
|------|--|
| x | Object of class KinPairData or KinPairSimulation |
| file | The file path to write to |
| ... | Additional arguments to pass to write_tsv |

Value

Invisibly returns the initial object

| | |
|---------|--|
| kinship | <i>Access or assign kin category (generic for KinPairData class)</i> |
|---------|--|

Description

Access or assign kin category (generic for KinPairData class)

Usage

```
kinship(x)
```

```
kinship(x) <- value
```

Arguments

| | |
|-------|-----------------------------|
| x | object with relevant method |
| value | new value to assign |

Value

character. Kinship category of object
returns modified object

| | |
|-----------|---|
| lifestage | <i>Access or assign lifestage (generic for KinPairData class)</i> |
|-----------|---|

Description

Access or assign lifestage (generic for KinPairData class)

Usage

```
lifestage(x)
```

```
lifestage(x) <- value
```

Arguments

| | |
|-------|-----------------------------|
| x | object with relevant method |
| value | new value to assign |

Value

character life stage of object
returns modified object with altered lifestage

| | |
|---------|---|
| mentari | <i>Position & kinship information of Aedes aegypti from Mentari Court, Malaysia</i> |
|---------|---|

Description

A data file containing the positions & kinship values of 98 *Ae. aegypti* larval kin pairs collected between September 19 & October 10, 2017 in Mentari Court (Petaling Jaya), Malaysia.

Usage

```
mentari
```

Format

A data frame with 98 rows and 10 variables

id1 id of first individual of kinpair

id2 id of second individual of kinpair

kinship kinship category of the pairing

distance geographical distance between kinpair

x1 relative x coordinate of first individual in metres
y1 relative y coordinate of first individual in metres
x2 relative x coordinate of second individual in metres
y2 relative y coordinate of second individual in metres
lifestage lifestage at time of sampling of kinpair
k_loiselle calculated Loiselle's k value for kinpair

Details

162 individuals were sourced as larvae from ovitraps placed in eight apartment buildings (in floors three or four for each), collected over three weeks. Entire larval bodies were extracted and sequenced using the double-digest restriction-site-associated DNA sequencing protocol for *Ae. aegypti* (doi: [10.1186/1471216415275](https://doi.org/10.1186/1471216415275)). After sequencing & genotyping, Loiselle's k was used as an initial estimate of genetic kinship. The program ML-Relate (doi: [10.1111/j.14718286.2006.01256.x](https://doi.org/10.1111/j.14718286.2006.01256.x)) was then used to estimate the pedigree kinships for the FS and HS categories. Following simulation work described in doi: [10.1111/17550998.13043](https://doi.org/10.1111/17550998.13043) the 1C category was assigned to all remaining unassigned individuals with a Loiselle's k of less than 0.06.

Value

returns an object of class `tbl_df`

Source

doi: [10.1111/17550998.13043](https://doi.org/10.1111/17550998.13043)

mount_appdata

Mount KinPairData Objects for use in kindisperse app

Description

Mount KinPairData Objects for use in kindisperse app

Usage

```
mount_appdata(x, nm)
```

Arguments

x An object of class `KinPairData` or `KinPairSimulation`
nm character. A name to store the object as

Value

invisibly returns `x`.

Examples

```
mount_appdata(KinPairData(), "mydata")
```

| | |
|------------|--------------|
| ovisigma<- | <i>Title</i> |
|------------|--------------|

Description

Title

Usage

```
ovisigma(x) <- value
```

Arguments

| | |
|-------|-----------------------------------|
| x | object of class KinPairSimulation |
| value | new value to assign |

Value

returns a modified object of the relevant class

| | |
|---------|---|
| posigma | <i>kindisperse - access sigmas of KinPairSimulation objects</i> |
|---------|---|

Description

kindisperse - access sigmas of KinPairSimulation objects

Usage

```
posigma(x)  
initsigma(x)  
breedsigma(x)  
gravsigma(x)  
ovisigma(x)
```

Arguments

| | |
|---|-----------------------------------|
| x | object of class KinPairSimulation |
|---|-----------------------------------|

Value

numeric posigma value of simple simulation
 numeric initsigma value of composite simulation
 numeric breedsigma value of composite simulation
 numeric gravsigma value of composite simulation
 numeric ovisigma value of composite simulation

| | |
|-----------|--------------|
| posigma<- | <i>Title</i> |
|-----------|--------------|

Description

Title

Usage

```
posigma(x) <- value
```

Arguments

| | |
|-------|-----------------------------------|
| x | object of class KinPairSimulation |
| value | new value to assign |

Value

returns a modified object of the relevant class

| | |
|--------------|---|
| read_kindata | <i>Reads .kindata filetype back to KinPairData or KinPairSimulation object.</i> |
|--------------|---|

Description

Reads .kindata filetype back to KinPairData or KinPairSimulation object.

Usage

```
read_kindata(file)
```

Arguments

| | |
|------|---|
| file | Character giving path reference to file with extension .kinpair |
|------|---|

Value

Returns either KinPairData or KinPairSimulation object.

| | |
|-------------|---|
| rebase_dims | <i>Change the dimensions of a KinPairSimulation Object and shift kinpairs so at least one individual is within the area</i> |
|-------------|---|

Description

Change the dimensions of a KinPairSimulation Object and shift kinpairs so at least one individual is within the area

Usage

```
rebase_dims(kindist, dims)
```

Arguments

| | |
|---------|--|
| kindist | KinPairSimulation - KinPairSimulation Class Object |
| dims | New site dimensions - either single number (length of side of square) or length 2 vector (lengths of sides x and y of rectangle) |

Value

returns a rebased object of class KinPairSimulation with adjusted simulation dimensions

Examples

```
simobject <- simulate_kindist_simple()
rebase_dims(simobject, c(1, 100))
rebase_dims(simobject, 15)
```

| | |
|---------------|----------------------------------|
| reset_appdata | <i>Reset kindisperse appdata</i> |
|---------------|----------------------------------|

Description

Reset kindisperse appdata

Usage

```
reset_appdata()
```

Value

No return value, called for side effects

Examples

```
reset_appdata()
```

| | |
|----------------|--|
| reset_tempdata | <i>Reset app tempdata (internal mem)</i> |
|----------------|--|

Description

Reset app tempdata (internal mem)

Usage

```
reset_tempdata()
```

Value

No return value, called for side effects

Examples

```
reset_tempdata()
```

| | |
|---------------------|--|
| retrieveall_appdata | <i>Retrieve all KinPairData objects from appdata (as list)</i> |
|---------------------|--|

Description

Retrieve all KinPairData objects from appdata (as list)

Usage

```
retrieveall_appdata()
```

Value

Returns a list of objects stored in the appdata environment

Examples

```
mount_appdata(KinPairData(), "k1")  
mount_appdata(KinPairSimulation(), "s1")  
retrieveall_appdata()
```

| | |
|------------------|--|
| retrieve_appdata | <i>Retrieve KinPairData object from appdata (single)</i> |
|------------------|--|

Description

Retrieve KinPairData object from appdata (single)

Usage

```
retrieve_appdata(nm)
```

Arguments

nm character. Name of item as stored in appdata

Value

Returns KinPairData object accessible by name nm

Examples

```
mount_appdata(KinPairData(), "mydata")  
retrieve_appdata("mydata")
```

| | |
|-------------------|--|
| retrieve_tempdata | <i>Retrieve all tempdata (internal mem) from app (as list)</i> |
|-------------------|--|

Description

Retrieve all tempdata (internal mem) from app (as list)

Usage

```
retrieve_tempdata()
```

Value

A list of all KinPairData objects in kindisperse app's tempdata

Examples

```
retrieve_tempdata()
```

| | |
|-----------------|----------------------------|
| run_kindisperse | <i>Run kindisperse app</i> |
|-----------------|----------------------------|

Description

Run kindisperse app

Usage

```
run_kindisperse()
```

Value

returns a shiny app instance of kindisperse

| | |
|----------------|--|
| sample_kindist | <i>Subsample and filter a KinPairSimulation Object</i> |
|----------------|--|

Description

This function takes a pre-existing KinPairSimulation Object with distance and coordinate data and filters it to simulate various in-field sampling schemes.

Usage

```
sample_kindist(
  kindist,
  upper = NULL,
  lower = NULL,
  spacing = NULL,
  n = NULL,
  dims = NULL
)
```

Arguments

| | |
|---------|---|
| kindist | KinPairSimulation - KinPairSimulation Class Object |
| upper | numeric - upper cutoff for kin pair distances |
| lower | numeric - lower cutoff for kin pair distances |
| spacing | numeric - spacing between traps (assume 1D layout) |
| n | numeric - number of individuals to keep after filtering (if possible) |
| dims | dimensions to sample within (works with the KinPairSimulation spatial & dimension information). Either 'num' (square) or 'c(num1, num2)' (rectangle). |

Value

returns an object of class 'KinPairSimulation' containing simulation and filtering details and a tibble (tab) of filtered simulation values

Examples

```
simobject <- simulate_kindist_simple(nsims = 100000, sigma = 100, kinship = "P0")
sample_kindist(simobject, upper = 200, lower = 50, spacing = 15, n = 100)
```

| | |
|----------------------|---|
| <code>simdims</code> | <i>Access simulation dimensions of KinPairSimulation object</i> |
|----------------------|---|

Description

Access simulation dimensions of KinPairSimulation object

Usage

```
simdims(x)
```

Arguments

`x` object of class KinPairSimulation

Value

numeric vector dimensions of simulated object

| | |
|---------------------------|--------------|
| <code>simdims<-</code> | <i>Title</i> |
|---------------------------|--------------|

Description

Title

Usage

```
simdims(x) <- value
```

Arguments

`x` object of class KinPairSimulation
`value` new value to assign

Value

returns a modified object of the relevant class

| | |
|---------------|--|
| simgraph_data | <i>Simple kin dispersal simulation for graphical display. (returns the data side as a tibble).</i> |
|---------------|--|

Description

Simple kin dispersal simulation for graphical display. (returns the data side as a tibble).

Usage

```
simgraph_data(nsims = 1000, posigma = 50, dims = 250, kinship = "2C")
```

Arguments

| | |
|---------|---|
| nsims | Integer. The number of kin dispersal families to simulate. |
| posigma | Integer. The axial deviation of the (simple) parent-offspring dispersal kernel governing this simulation. |
| dims | Integer. Lays out the length of the sides of a square within which parent individuals are seeded. |
| kinship | Character. Lists the kin category the simulation is reconstructing. One of "PO", "FS", "HS", "AV", "GG", "HAV", "GGG", "1C", "1C1", "2C", "GAV" (no half-categories included) |

Value

Returns a tibble containing the coordinates of the f0 to f2 generations, as well as coordinates and distances relative to the 'focus' kinship categories. (kindist, kinmid, k1 & k2)

Examples

```
simgraph_data(nsims = 100, dims = 1000, kinship = "GAV")
```

| | |
|----------------|---|
| simgraph_graph | <i>Simple kin dispersal simulation for graphical display. (graphs the pre-existing simulation).</i> |
|----------------|---|

Description

Simple kin dispersal simulation for graphical display. (graphs the pre-existing simulation).

Usage

```

simgraph_graph(
  result,
  nsims = 10,
  labls = TRUE,
  steps = TRUE,
  moves = TRUE,
  shadows = TRUE,
  kinship = NULL,
  show_area = TRUE,
  centred = FALSE,
  pinwheel = FALSE,
  scattered = FALSE,
  lengths = TRUE,
  lengthlabs = TRUE,
  histogram = FALSE,
  binwidth = posigma/5,
  freqpoly = FALSE
)

```

Arguments

| | |
|------------|--|
| result | simulation supplied from simgraph_data() function (tibble) |
| nsims | number of families to graph |
| labls | Logical. Displays labels. |
| steps | Logical. Whether or not to show any details of dispersal movement |
| moves | Logical. Whether or not to show (curved) lines denoting dispersal movement |
| shadows | Logical. Whether or not to show (dashed) shadows tracing dispersal movement. |
| kinship | Character. Lists the kin category the simulation is reconstructing. One of "PO", "FS", "HS", "AV", "GG", "HAV", "GGG", "1C", "1C1", "2C", "GAV" (no half-categoris included) |
| show_area | Logical. Whether or not to show the parental seed area as defined in data\$dims |
| centred | Logical. Whether or not to centre the coordinates on one individual. |
| pinwheel | Logical. Whether the final graph should be of the pinwheel form. |
| scattered | Logical. Whether the final graph should be of the scatter form. |
| lengths | Logical. Whether or not to show a dashed line connecting the 'focus' kin to illustrate overall distance of dispersal. |
| lengthlabs | Logical. Whether to show labels denoting distance of dispersal between focus kin. |
| histogram | Logical. Whether the final graph should be of the histogram form. |
| binwidth | Numeric. Binwidth for histogram or freqpoly. |
| freqpoly | Logical. Whether the final graph should be of the freqpoly form. |

Value

Returns a ggplot object for graphing.

Examples

```
simdata <- simgraph_data()
simgraph_graph(simdata)
```

| | |
|---------|---|
| simtype | <i>kindisperse - access simtype of KinPairSimulation object</i> |
|---------|---|

Description

kindisperse - access simtype of KinPairSimulation object

Usage

```
simtype(x)
```

Arguments

x object of class KinPairSimulation

Value

character the kind of simulation stored in the object (simple or composite)

| | |
|-----------|--------------|
| simtype<- | <i>Title</i> |
|-----------|--------------|

Description

Title

Usage

```
simtype(x) <- value
```

Arguments

x object of class KinPairSimulation
value new value to assign

Value

returns a modified object of relevant class

 simulate_kindist_composite

Simulate kin dispersal distance pairs with composite sigmas

Description

Simulate kin dispersal distance pairs with composite sigmas

Usage

```
simulate_kindist_composite(
  nsims = 100,
  initsigma = 100,
  breedsigma = 50,
  gravsigma = 50,
  ovisigma = 25,
  dims = 100,
  method = "Gaussian",
  kinship = "FS",
  lifestage = "immature",
  shape = 0.5
)
```

Arguments

| | |
|------------|---|
| nsims | (integer) - number of pairs to simulate |
| initsigma | (numeric) - size of pre-breeding (axial) sigma |
| breedsigma | (numeric) - size of breeding (axial) sigma |
| gravsigma | (numeric) - size of post-breeding (axial) sigma |
| ovisigma | (numeric) - size of oviposition (axial) sigma |
| dims | (numeric) - length of sides of (square) simulated site area |
| method | (character) - kernel shape to use: either 'Gaussian', 'Laplace' or 'vgamma' (variance-gamma) |
| kinship | (character)- kin category to simulate: one of PO, FS, HS, AV, GG, HAV, GGG, 1C, 1C1, 2C, GAV, HGAV, H1C H1C1 or H2C |
| lifestage | (lifestage) lifestage at sample collection: either 'immature' or 'ovipositional' |
| shape | (numeric) - value of shape parameter to use with 'vgamma' method. Default 0.5 |

Value

returns an object of class KinPairSimulation containing simulation details and a tibble (tab) of simulation values

Examples

```
simulate_kindist_composite(nsims = 100)
simulate_kindist_composite(
  nsims = 10000, initsigma = 20, breedsigma = 30, gravsigma = 30,
  ovisigma = 12, dims = 500, method = "Laplace", kinship = "1C", lifestage = "immature"
)
```

```
simulate_kindist_simple
```

Simulate kin dispersal distance pairs with simple sigma

Description

Simulate kin dispersal distance pairs with simple sigma

Usage

```
simulate_kindist_simple(
  nsims = 100,
  sigma = 125,
  dims = 100,
  method = "Gaussian",
  kinship = "PO",
  lifestage = "immature",
  shape = 0.5
)
```

Arguments

| | |
|-----------|--|
| nsims | (integer) - number of pairs to simulate |
| sigma | (numeric) - size of simple (axial) sigma |
| dims | (numeric) - length of sides of (square) simulated site area |
| method | (character) - kernel shape to use: either 'Gaussian', 'Laplace' or 'vgamma' (variance-gamma) |
| kinship | (character)- kin category to simulate: one of PO, FS, HS, AV, GG, HAV, GGG, 1C, 1C1, 2C, GAV, HGAV, H1C or H2C |
| lifestage | (lifestage) lifestage at sample collection: either 'immature' or 'ovipositional' |
| shape | (numeric) value of shape parameter to use with 'vgamma' method. Default 0.5 |

Value

returns an object of class `KinPairSimulation` containing simulation details and a `tibble`

Examples

```
test <- simulate_kindist_simple(nsims = 10, sigma = 50, dims = 1000, method = "Laplace")
simulate_kindist_simple(nsims = 10000, sigma = 75, kinship = "PO", lifestage = "ovipositional")
```

| | |
|----------------|--|
| tsv_to_kinpair | <i>Reads .tsv and converts to KinPairData object</i> |
|----------------|--|

Description

Reads .tsv and converts to KinPairData object

Usage

```
tsv_to_kinpair(file, kinship = NULL, lifestage = NULL, ...)
```

Arguments

| | |
|-----------|--|
| file | The file path to read from |
| kinship | character. kin category to assign or extract from data. one of PO, FS, HS, AV, GG, HAV, GGG, 1C, 1C1, 2C, GAV, HGAV, H1C , H1C1 or H2C |
| lifestage | character. lifestage to assign or extract from data. one of 'unknown', 'immature' or 'ovipositional'. |
| ... | additional arguments to pass to read_tsv |

Value

Returns object of class KinPairData

| | |
|-----------------|---|
| unmount_appdata | <i>Unmount a KinPairData Object (clear slot from appdata environment)</i> |
|-----------------|---|

Description

Unmount a KinPairData Object (clear slot from appdata environment)

Usage

```
unmount_appdata(nms)
```

Arguments

| | |
|-----|--|
| nms | A character vector of names of objects to unmount from the appdata environment |
|-----|--|

Value

No return value, called for side effects

Examples

```
mount_appdata(KinPairData(), "mydata")  
  
unmount_appdata("mydata")
```

upper

Access & filter by filter parameters of KinPairSimulation Object

Description

Access & filter by filter parameters of KinPairSimulation Object

Usage

```
upper(x)  
  
upper(x) <- value  
  
lower(x)  
  
lower(x) <- value  
  
spacing(x)  
  
spacing(x) <- value  
  
samplenum(x)  
  
samplenum(x) <- value  
  
sampledims(x)  
  
sampledims(x) <- value
```

Arguments

| | |
|-------|-----------------------------------|
| x | object of class KinPairSimulation |
| value | new value to assign |

Value

numeric upper value of sampled object
returns a modified object of the relevant class
numeric lower value of sampled object
returns a modified object of the relevant class
numeric trap spacing of sampled object

returns a modified object of the relevant class
 numeric number of kin dyads in sampled object
 returns a modified object of the relevant class
 numeric vector dimensions of sampled object
 returns a modified object of the relevant class

vector_to_kinpair *Convert vector of kin separation distances to KinPairData class*

Description

Convert vector of kin separation distances to KinPairData class

Usage

```
vector_to_kinpair(vect, kinship = NULL, lifestage = NULL)
```

Arguments

vect vector of kinpair distances
 kinship character or character vector containing kinship categories of kinpairs
 lifestage character or character vector containing lifestages of kinpairs

Value

returns valid KinPairData object.

Examples

```
vector_to_kinpair(1:10, "FS", "immature")
```

write_kindata *Write KinPairData or KinPairSimulation object in .kindata format (no information lost)*

Description

Write KinPairData or KinPairSimulation object in .kindata format (no information lost)

Usage

```
write_kindata(x, file)
```

Arguments

- `x` Object of class `KinPairData` or `KinPairSimulation`
- `file` The file path to write to. If it doesn't end in `'.kindata'`, this will be added.

Value

Invisibly returns the initial object

Index

* **datasets**
 mentari, 35

axials, 3
axials_add, 4
axials_combine, 5
axials_decompose, 5
axials_standard, 6
axials_subtract, 8
axpermute, 9
axpermute_standard, 10
axpermute_subtract, 12

breedsigma (posigma), 37
breedsigma, KinPairSimulation-method
 (kernelshape, KinPairSimulation-method),
 22
breedsigma<-, 13

check_valid_kinship, 13
check_valid_lifestage, 14
csv_to_kinpair, 14

df_to_kinpair, 15
display_appdata, 15
distances, 16
distances, KinPairData-method, 16

elongate, 18

filtertype, 18
filtertype, KinPairSimulation-method
 (kernelshape, KinPairSimulation-method),
 22
filtertype<-, 19

gravsigma (posigma), 37
gravsigma, KinPairSimulation-method
 (kernelshape, KinPairSimulation-method),
 22
gravsigma<-, 19

initialize, KinPairData-method
 (distances, KinPairData-method),
 16
initialize, KinPairSimulation-method
 (kernelshape, KinPairSimulation-method),
 22
initsigma (posigma), 37
initsigma, KinPairSimulation-method
 (kernelshape, KinPairSimulation-method),
 22
initsigma<-, 20
is.KinPairData, 20
is.KinPairSimulation, 21

kernelshape, 21
kernelshape, KinPairSimulation-method,
 22
kerneltype, 26
kerneltype, KinPairSimulation-method
 (kernelshape, KinPairSimulation-method),
 22
kerneltype<-, 26
kinpair_to_csv, 33
kinpair_to_tibble, 33
kinpair_to_tsv, 34
KinPairData, 27
KinPairSimulation, 28
KinPairSimulation_composite, 30
KinPairSimulation_simple, 32
kinship, 34
kinship, KinPairData-method
 (distances, KinPairData-method),
 16
kinship<- (kinship), 34
kinship<-, KinPairData-method
 (distances, KinPairData-method),
 16

lifestage, 35

lifestage, KinPairData-method
 (distances, KinPairData-method),
 16
 lifestage<- (lifestage), 35
 lifestage<- , KinPairData-method
 (distances, KinPairData-method),
 16
 lower (upper), 50
 lower, KinPairSimulation-method
 (kernelshape, KinPairSimulation-method),
 22
 lower<- (upper), 50
 lower<- , KinPairSimulation-method
 (kernelshape, KinPairSimulation-method),
 22
 mentari, 35
 mount_appdata, 36

 ovisigma (posigma), 37
 ovisigma, KinPairSimulation-method
 (kernelshape, KinPairSimulation-method),
 22
 ovisigma<- , 37

 posigma, 37
 posigma, KinPairSimulation-method
 (kernelshape, KinPairSimulation-method),
 22
 posigma<- , 38

 read_kindata, 38
 rebase_dims, 39
 reset_appdata, 39
 reset_tempdata, 40
 retrieve_appdata, 41
 retrieve_tempdata, 41
 retrieveall_appdata, 40
 run_kindisperse, 42

 sample_kindist, 42
 sampledims (upper), 50
 sampledims, KinPairSimulation-method
 (kernelshape, KinPairSimulation-method),
 22
 sampledims<- (upper), 50
 sampledims<- , KinPairSimulation-method
 (kernelshape, KinPairSimulation-method),
 22

 samplenum (upper), 50
 samplenum, KinPairSimulation-method
 (kernelshape, KinPairSimulation-method),
 22
 samplenum<- (upper), 50
 samplenum<- , KinPairSimulation-method
 (kernelshape, KinPairSimulation-method),
 22
 show, KinPairData-method
 (distances, KinPairData-method),
 16
 show, KinPairSimulation-method
 (kernelshape, KinPairSimulation-method),
 22
 simdims, 43
 simdims, KinPairSimulation-method
 (kernelshape, KinPairSimulation-method),
 22
 simdims<- , 43
 simgraph_data, 44
 simgraph_graph, 44
 simtype, 46
 simtype, KinPairSimulation-method
 (kernelshape, KinPairSimulation-method),
 22
 simtype<- , 46
 simulate_kindist_composite, 47
 simulate_kindist_simple, 48
 spacing (upper), 50
 spacing, KinPairSimulation-method
 (kernelshape, KinPairSimulation-method),
 22
 spacing<- (upper), 50
 spacing<- , KinPairSimulation-method
 (kernelshape, KinPairSimulation-method),
 22

 tsv_to_kinpair, 49

 unmount_appdata, 49
 upper, 50
 upper, KinPairSimulation-method
 (kernelshape, KinPairSimulation-method),
 22
 upper<- (upper), 50
 upper<- , KinPairSimulation-method
 (kernelshape, KinPairSimulation-method),
 22

vector_to_kinpair, [51](#)

write_kindata, [51](#)