Package 'plantTracker'

January 11, 2023

Title Extract Demographic and Competition Data from Fine-Scale Maps

Version 1.0.1

Date 2023-01-09

Author Alice Stears [aut, cre] (<<https://orcid.org/0000-0002-1325-6641>>),
Jared Studyvin [aut],
David Atkins [aut]

Maintainer Alice Stears <alice.e.stears@gmail.com>

Description Extracts growth, survival, and local neighborhood density
information from repeated, fine-scale maps of organism occurrence. Further
information about this package can be found in our journal article,
``plantTracker: An R package to translate maps of plant occurrence into
demographic data'' published in 2022 in Methods in Ecology and Evolution
(Stears, et al., 2022) <[doi:10.1111/2041-210X.13950](https://doi.org/10.1111/2041-210X.13950)>.

License MIT + file LICENSE

URL <https://github.com/aestears/plantTracker>

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Depends R (>= 2.10)

Suggests testthat (>= 3.0.0), knitr, rmarkdown, spelling, minidown,
ggplot2

Config/testthat/edition 3

Imports Matrix, igraph, sf, methods, stats, graphics, grDevices, units

VignetteBuilder knitr

Language en-US

NeedsCompilation no

Repository CRAN

Date/Publication 2023-01-11 10:20:08 UTC

R **topics documented:**

plantTracker-package	*Extract Demographic and Competition Data from Fine-Scale Maps*

Description

Extracts growth, survival, and local neighborhood density information from repeated, fine-scale maps of organism occurrence. Further information about this package can be found in our journal article, "plantTracker: An R package to translate maps of plant occurrence into demographic data" published in 2022 in Methods in Ecology and Evolution (Stears, et al., 2022) <doi:10.1111/2041-210X.13950>.

Details

The DESCRIPTION file:

Package:	plantTracker
Title:	Extract Demographic and Competition Data from Fine-Scale Maps
Version:	1.0.1
Date:	2023-01-09
Authors@R:	c(person(given = "Alice", family = "Stears", role = c("aut", "cre"), email = "alice.e.stears@gmail.com
Author:	Alice Stears [aut, cre] (<https://orcid.org/0000-0002-1325-6641>), Jared Studyvin [aut], David Atki
Maintainer:	Alice Stears <alice.e.stears@gmail.com>
Description:	Extracts growth, survival, and local neighborhood density information from repeated, fine-scale map
License:	MIT + file LICENSE
URL:	https://github.com/aestears/plantTracker
Encoding:	UTF-8
LazyData:	true
Roxygen:	list(markdown = TRUE)
RoxygenNote:	7.2.3
Depends:	R (>= 2.10)

Suggests:	testthat (>= 3.0.0), knitr, rmarkdown, spelling, minidown, ggplot2
Config/testthat/edition:	3
Imports:	Matrix, igraph, sf, methods, stats, graphics, grDevices, units
VignetteBuilder:	knitr
Language:	en-US

Index of help topics:

aggregateByGenet	aggregateByGenet
assign	Tracks genets through time
checkDat	checkDat
drawQuadMap	Create maps of a quadrat over time
getBasalAreas	Calculates basal area for each species in a quadrat
getLambda	Calculates lambda, the population growth rate, for each species in a quadrat based on changes in basal cover.
getNeighbors	Calculates local neighborhood density around each individual in a mapped dataset
getRecruits	Calculates the number of recruits of each species per year in each quadrat
grasslandData	Chart Quadrat Dataset for grassland species in MT and AZ An example dataset that contains data from mapped plant chart quadrats. Each row is an observation of one plant.
grasslandInventory	A dataset indicating sampling years for the grasslandData example dataset
groupByGenet	Group polygon observations together into a 'genet' based on proximity
plantTracker-package	Extract Demographic and Competition Data from Fine-Scale Maps
trackSpp	Tracks genets through time for multiple species and sites

Author(s)

Alice Stears [aut, cre] (<https://orcid.org/0000-0002-1325-6641>), Jared Studyvin [aut], David Atkins [aut]

Maintainer: Alice Stears <alice.e.stears@gmail.com>

See Also

Use the following code to see the vignette will guide you through using the plantTracker package
vignette(topic = "Suggested_plantTracker_Workflow", package = "plantTracker")

Use the following code to see the vignette goes through the trackSpp() function step-by-step vignette(topic = "Using_the_plantTracker_trackSpp_function", package = "plantTracker")

---

aggregateByGenet   *aggregateByGenet*

Description

This function aggregates a data.frame by unique trackIDs so that each row represents a genet (unique genetic individual) in a given year.

Usage

```
aggregateByGenet(
  dat,
  site = "Site",
  quad = "Quad",
  species = "Species",
  year = "Year",
  trackID = "trackID",
  ...
)
```

Arguments

dat	An sf data.frame. Typically this is a data.frame that has been returned from the `trackSpp()` function, although 'dat' can be any sf data.frame of plant demographic data in which each genet is given unique identifying value (what we call here a 'trackID'). If 'dat' already only has one row for each unique trackID in each unique year (i.e. there are no vegetative individuals–no ramets), then the output of `aggregateByGenet()` will have the same number of rows as 'dat'. If 'dat' has multiple rows (each of which is a ramet) for the same trackID in the same year, then the output of `aggregateByGenet()` will have fewer rows than the input 'dat' data.frame. 'dat' MUST have columns called 'basalArea_genet', 'age', 'recruit', 'survives_tplus1', 'nearEdge', and 'size_tplus1', although they can be populated with NAs. The `trackSpp()` function adds these columns, so if you have made no changes to the data.frame that was returned from `trackSpp()`, then your data.frame should have these columns already.
site	A character string giving the name of the column in 'dat' that contains the values for site. The default is "Site".
quad	A character string giving the name of the column in 'dat' that contains the values for quadrat. The default is "Quad".
species	A character string giving the name of the column in 'dat' that contains the values for species. The default is "Species".
year	A character string giving the name of the column in 'dat' that contains the values for year. The default is "Year".
trackID	A character string giving the name of the column in 'dat' that contains the values for the unique identifier, or trackID. The default is "trackID".
...	Other arguments passed on to methods. Not currently used.

Details

This function is a **plantTracker**-specific, user-friendly wrapper for the 'aggregate' function. This function was designed for use within the [trackSpp](#) function, but can be also called independently by the user. The function is designed to take a data.frame of the same format that is returned by [trackSpp()](#), but it can be used to aggregate any dataset by genet (or some other unique identifier), as long as there is a column that identifies genet (or unique identifier) identity for each observation.

Value

An sf data.frame that has been aggregated by genet so that each row represents one unique genetic individual in one year. This data.frame has columns containing data for site, quadrat, species, year, and trackID, as well as columns called 'basalArea_genet', 'age', 'recruit', 'survives_tplus1', 'nearEdge', and 'size_tplus1'. This data.frame will most likely be shorter than the input 'dat', because a genet that was previously broken into multiple rows representing each ramet in each year will now have only one row in each year.

See Also

This function is called inside the [trackSpp()](#) function.

Examples

```
dat <- grasslandData[grasslandData$Site == c("AZ") &
 grasslandData$Species %in% c("Bouteloua rothrockii", "Calliandra eriophylla" ),]
names(dat)[1] <- "speciesName"
inv <- grasslandInventory[unique(dat$Quad)]
outDat <- trackSpp(dat = dat,
 inv = inv,
 dorm = 1,
 buff = .05,
 buffGenet = 0.005,
 clonal = data.frame("Species" = unique(dat$speciesName),
 "clonal" = c(TRUE,FALSE)),
 species = "speciesName",
 aggByGenet = FALSE
 )

aggregateByGenet(dat = outDat,
species = 'speciesName')
```

---

assign   *Tracks genets through time*

Description

This function tracks individual plants through time, but only for one species in one quadrat. It is designed for use within the [trackSpp](#) function, and is not intended for use on its own.

Usage

```
assign(
  dat,
  inv,
  dorm,
  buff,
  buffGenet,
  clonal,
  flagSuspects = FALSE,
  shrink = 0.1,
  dormSize = 0.05,
  inheritsFromTrackSpp = FALSE,
  nearEdgeBox = NULL,
  ...
)
```

Arguments

dat	An sf data.frame of the same format as [grasslandData](#). 'dat' must contain data for only one species in one quadrat. It must have columns that contain a unique identification for each research site (default name is "Site"), species name (default name is "Species"), quadrat identifier (default name is "Quad"), year of data collection (default name is "Year") and an s.f 'geometry' column that contains a polygon or multipolygon data type for each individual observation. This function will add columns called "basalArea_ramet", "trackID", "age", "size_tplus1", "recruit" and "survives_tplus1", so 'dat' should not contain columns with these names.
inv	An integer vector that contains all of years in which this quadrat (or other unique spatial area) was sampled. Years must be ordered sequentially.
dorm	A numeric vector of length 1, indicating the number of years this species is allowed to go dormant, i.e. be absent from the map but be considered the same individual when it reappears. This must be an integer greater than or equal to 0.
buff	A numeric vector of length 1 that is greater than or equal to zero, indicating how far (in the same units as spatial values in 'dat') a polygon can move from year t to year t+1 and still be considered the same individual.
buffGenet	A numeric vector of length 1 that is greater than or equal to zero, indicating how close (in the same units as spatial values in 'dat') polygons must be to one another in the same year to be grouped as a genet (if 'clonal' argument = TRUE). This argument is passed to the [groupByGenet](#) function, which is used inside the [assign](#) function.
clonal	A logical argument of length 1, indicating whether this species is allowed to be clonal or not (i.e. if multiple polygons (ramets) can be grouped into one individual (genet)). If clonal = TRUE, the species is allowed to be clonal, and if clonal = FALSE, the species is not allowed to be clonal.
flagSuspects	A logical argument of length 1, indicating whether observations that are 'suspect' will be flagged. The default is flagSuspects = FALSE. If flagSuspects

= TRUE, then a column called 'Suspect' is added to the output data.frame. Any suspect observations get a 'TRUE' in the 'Suspect' column, while non-suspect observations receive a 'FALSE'. There are two ways that an observation can be classified as 'suspect'. First, if two consecutive observations have the same trackID, but the basal area of the observation in year t+1 is less that a certain percentage (defined by the shrink arg.) of the basal area of the observation in year t, it is possible that the observation in year t+1 is a new recruit and not the same individual. The second way an observation can be classified as 'suspect' is if it is very small before going dormant. It is unlikely that a very small individual will survive dormancy, so it is possible that the function has mistakenly given a survival value of '1' to this individual. A 'very small individual' is any observation with an area below a certain percentile (specified by 'dormSize') of the size distribution for this species, which is generated using all of the size data for this species in 'dat'.

shrink   A single numeric value. This value is only used when flagSuspects = TRUE. When two consecutive observations have the same trackID, and the ratio of size_t+1 to size_t is smaller than the value of shrink, the observation in year t gets a 'TRUE' in the 'Suspect' column. For example, shrink = 0.2, and an individual that the tracking function has identified as 'BOUGRA_1992_5' has an area of 9 cm^2 in year t and an area of 1.35 cm^2 in year t+1. The ratio of size t+1 to size t is 1.35/9 = 0.15, which is smaller than the cutoff specified by shrink, so the observation of BOUGRA_1992_5' in year t gets a 'TRUE' in the 'Suspect' column. The default value is shrink = 0.10.

dormSize   A single numeric value. This value is only used when flagSuspects = TRUE and dorm is greater than or equal to 1. An individual is flagged as 'suspect' if it 'goes dormant' and has a size that is less than or equal to the percentile of the size distribution for this species that is designated by dormSize. For example, dormSize = 0.05, and the focal individual has a basal area of 0.5 cm^2. The 5th percentile of the distribution of size for this species, which is made using the mean and standard deviation of all observations in 'dat' for the species in question, is 0.6 cm^2. This individual does not have any overlaps in the next year (year t+1), but does have an overlap in year t+2. However, because the basal area of this observation is smaller than the 5th percentile of size for this species, the observation in year t will get a 'TRUE' in the 'Suspect' column. It is possible that the tracking function has mistakenly assigned a '1' for survival in year t, because it is unlikely that this individual is large enough to survive dormancy. The default value is dormSize = .05.

inheritsFromTrackSpp
                      A logical argument that is only applicable when 'assign()' is used internally in 'trackSpp()'.

nearEdgeBox   An sf data frame indicating the bounding box of the quadrat. This argument is only used if 'assign()' is used internally in 'trackSpp()'.

...   Other arguments passed on to methods. Not currently used.

Details

see [trackSpp](#) for details of arguments and usage.

See Also

[trackSpp()](), which is a wrapper for the [assign()]() function that applies it over many species and quadrats. The [assign()]() function uses the [groupByGenet()]() function to group ramets into genets (if 'clonal' argument = TRUE).

---

checkDat   *checkDat*

Description

A function that both checks and prepares a data.frame for use in the [trackSpp]() function.

Usage

```
checkDat(
  dat,
  inv = NULL,
  species = "Species",
  site = "Site",
  quad = "Quad",
  year = "Year",
  geometry = "geometry",
  reformatDat = FALSE,
  ...
)
```

Arguments

dat   An sf data.frame of the same format as [grasslandData](). It must have columns that contains a unique identification for each research site (default name is "Site"), species name (default name is "Species"), quadrat identifier (default name is "Quad"), year of data collection (default name is "Year") and an s.f 'geometry' column that contains a polygon or multipolygon data type for each individual observation.

inv   The name of each element of the list is a quadrat name in 'dat', and the contents of that list element is a numeric vector of all of the years in which that quadrat (or other unique spatial area) was sampled. Make sure this is the years the quadrat was actually sampled, not just the years that have data in the 'dat' argument! This argument allows the function to differentiate between years when the quadrat wasn't sampled and years when there just weren't any individuals of a species present in that quadrat.

species   An optional character string argument. Indicates the name of the column in 'dat' that contains species name data. It is unnecessary to include a value for this argument if the column name is "Species" (default value is 'Species').

site   An optional character string argument. Indicates the name of the column in 'dat' that contains site name data. It is unnecessary to include a value for this argument if the column name is "Site" (default value is 'Site').

quad	An optional character string argument. Indicates the name of the column in 'dat' that contains quadrat name data. It is unnecessary to include a value for this argument if the column name is "Quad" (default is 'Quad').
year	An optional character string argument. Indicates the name of the column in 'dat' that contains data for year of sampling. It is unnecessary to include a value for this argument if the column name is "Year" (default is 'Year').
geometry	An optional character string argument. Indicates the name of the column in 'dat' that contains sf geometry data. It is unnecessary to include a value for this argument if the column name is "geometry" (default is 'geometry').
reformatDat	A TRUE/FALSE argument. If 'FALSE', which is the default value, then checkDat() prints a message that says the 'dat' and 'inv' datasets are ready for use in the trackSpp() function. This message includes any column name arguments that must be included in the trackSpp() function call for these datasets. If 'TRUE', checkDat() returns a list with three elements: a version of 'dat' that has been checked for errors and is ready to be input directly into the trackSpp or assign functions, a version of 'inv' that is checked and ready for input into trackSpp() or assign(), and a character vector called 'userColNames' that has all of the user-defined column names that were used in the original version of 'dat'.
...	Other arguments passed on to methods. Not currently used.

Details

This function is used internally in trackSpp and other plantTracker functions to check the 'dat' and 'inv' arguments and ensure that they are in the correct format with the correct column names. checkDat() can also be used independently to check that a data.frame is in the correct format and contains the correct data to be used in the trackSpp() or other plantTracker functions.

Value

If the 'reformatDat' argument is FALSE (the default), the return of this function is a message that says the 'dat' and 'inv' datasets are ready for use in the trackSpp() function. This message includes any column name arguments that must be included in the trackSpp() function call for these datasets. If the 'reformatDat' argument is TRUE, then checkDat() returns a list that contains versions of 'dat' and 'inv' that are ready to go directly into the trackSpp() or assign() functions. The list has the following elements:

dat	An sf data.frame that has five columns called 'Species', 'Site', 'Quad', 'Year', and 'geometry'. Any additional columns that were present in the version of 'dat' input into the checkDat() function will also be included in this version, although the column names will be appended with "_USER".
inv	A named list. The name of each element of the list is a quadrat name in 'dat', and the contents of that list element is a numeric vector of all of the years in which that quadrat (or other unique spatial area) was sampled. For each list element, the vector of years is in sequential order from oldest to most recent.
userColNames	A named character vector. This vector contains the column names provided by the user for all of the required data columns in the original version of 'dat'. The name of each vector element indicates the type of data that is contained

in the column with the name in that vector element. For example, if the 'dat' data.frame input into [checkDat()](#) has the names 'Species','location' 'quadrat', 'Year', and 'geometry', then this list element will be a character vector of these values with the names 'species', 'site', 'quad', 'year', and 'geometry'.

Examples

```
checkDat(dat = grasslandData,
inv = grasslandInventory)

checkDat(dat = grasslandData,
inv = grasslandInventory,
reformatDat = TRUE)
```

---

drawQuadMap   *Create maps of a quadrat over time*

Description

This function creates maps of a quadrat over time, color-coded by either species or by genet (trackID).

Usage

```
drawQuadMap(
  dat,
  type = "bySpecies",
  addBuffer = FALSE,
  species = "Species",
  site = "Site",
  quad = "Quad",
  year = "Year",
  geometry = "geometry",
  trackID = "trackID",
  ...
)
```

Arguments

dat   An sf data.frame in which each row represents a unique polygon (either a genet
                   or a ramet) in a unique site/quadrat/year combination. It is recommended that
                   you only input data for one quadrat at a time. A data.frame returned by [trackSpp](#)
                   can be put into this function after being subset by quadrat. 'dat' must have
                   columns that contain a unique identification for each research site (default name
                   is "Site"), species name (default name is "Species"), quadrat identifier (default
                   name is "Quad"), year of data collection (default name is "Year"), a unique iden-
                   tifier for each genet (default name is 'trackID') and an s.f 'geometry' column that
                   contains a polygon or multipolygon data type for each individual observation.

type	A character argument indicating how the plots returned by drawQuadMap() will be color coded. If type = "bySpecies", then observations are color-coded by species. If type = "bytrackID", then observations are color-coded by trackID. The default value is "bySpecies".
addBuffer	A logical argument indicating whether drawQuadMap() will draw a small buffer around each polygon in the returned maps to make the observations more visible. This is particularly useful for observations that were originally mapped as points, which are hard to see when plotted in their original dimensions. The buffer distance is 1/20th of the quadrat width. The default value is FALSE.
species	An optional character string argument. Indicates the name of the column in 'dat' that contains species name data. It is unnecessary to include a value for this argument if the column name is "Species" (default value is 'Species').
site	An optional character string argument. Indicates the name of the column in 'dat' that contains site name data. It is unnecessary to include a value for this argument if the column name is "Site" (default value is 'Site').
quad	An optional character string argument. Indicates the name of the column in 'dat' that contains quadrat name data. It is unnecessary to include a value for this argument if the column name is "Quad" (default is 'Quad').
year	An optional character string argument. Indicates the name of the column in 'dat' that contains data for year of sampling. It is unnecessary to include a value for this argument if the column name is "Year" (default is 'Year').
geometry	An optional character string argument. Indicates the name of the column in 'dat' that contains sf geometry data. It is unnecessary to include a value for this argument if the column name is "geometry" (default is 'geometry').
trackID	An optional character string argument. Indicates the name of the column in 'dat' that contains unique identifiers for each genet. It is unnecessary to include a value for this argument if the column name is "trackID" (default is 'trackID')
...	Other arguments passed on to methods. Not currently used.

Value

This function returns a multipanel plot where each panel shows a map of the quadrat in a unique year. Panels are arranged in chronological order, and plots are color-coded either by species or trackID (unique genet identifier).

#' @seealso [trackSpp()](), which can be used to assign trackIDs to observations.

Examples

```
 dat <- grasslandData[grasslandData$Site == c("AZ") &
 grasslandData$Quad == "SG2" &
 grasslandData$Year %in% c(1922:1925),]
names(dat)[1] <- "speciesName"
inv <- grasslandInventory[unique(dat$Quad)]
outDat <- trackSpp(dat = dat,
 inv = inv,
 dorm = 1,
 buff = .05,
```

```
buffGenet = 0.005,
clonal = data.frame("Species" = unique(dat$speciesName),
"clonal" = c(TRUE,FALSE)),
species = "speciesName",
aggByGenet = TRUE
)
drawQuadMap(dat = outDat,
type = "bySpecies",
addBuffer = FALSE,
species = "speciesName"
)
```

---

getBasalAreas   *Calculates basal area for each species in a quadrat*

Description

This function calculates the total basal area for each species present in a quadrat for each year of data collection. Both the absolute basal area (in the same units of area as the input data.frame) is returned, as well as the percentage of the total occupied basal area in the quadrat that is occupied by a given species ("percent total basal area"). If you'd like to ultimately calculate the population growth rate (lambda) for each species, you can use the [getLambda](#) function directly, which takes raw species occurrence data (like in 'dat' here) and returns lambda values for each species and quadrat for each transition in the dataset. This function should only be used if the individuals in 'dat' were mapped as polygons that are representative of their actual basal area, i.e. were not mapped as points.

Usage

```
getBasalAreas(
  dat,
  inv,
  species = "Species",
  quad = "Quad",
  site = "Site",
  year = "Year",
  geometry = "geometry",
  ...
)
```

Arguments

dat   An sf data.frame in which each row represents a unique polygon (either a genet or a ramet) in a unique site/quadrat/year combination. A data.frame returned by [trackSpp](#) can be put directly into this function. However, it is not necessary for 'dat' to have demographic data or unique identifiers (i.e. 'trackIDs') assigned. 'dat' must have columns that contain a unique identification for each research site (default name is "Site"), species name (default name is "Species"), quadrat identifier (default name is "Quad"), year of data collection (default name

is "Year"), and an s.f 'geometry' column that contains a polygon or multipolygon data type for each individual observation. This function should only be used if the individuals in 'dat' were mapped as polygons that are representative of their actual basal area, i.e. were not mapped as points.

inv   The name of each element of the list is a quadrat name in 'dat', and the contents of that list element is a numeric vector of all of the years in which that quadrat (or other unique spatial area) was sampled. Make sure this is the years the quadrat was actually sampled, not just the years that have data in the 'dat' argument! This argument allows the function to differentiate between years when the quadrat wasn't sampled and years when there just weren't any individuals of a species present in that quadrat.

species   An optional character string argument. Indicates the name of the column in 'dat' that contains species name data. It is unnecessary to include a value for this argument if the column name is "Species" (default value is 'Species').

quad   An optional character string argument. Indicates the name of the column in 'dat' that contains quadrat name data. It is unnecessary to include a value for this argument if the column name is "Quad" (default is 'Quad').

site   An optional character string argument. Indicates the name of the column in 'dat' that contains site name data. It is unnecessary to include a value for this argument if the column name is "Site" (default value is 'Site').

year   An optional character string argument. Indicates the name of the column in 'dat' that contains data for year of sampling. It is unnecessary to include a value for this argument if the column name is "Year" (default is 'Year').

geometry   An optional character string argument. Indicates the name of the column in 'dat' that contains sf geometry data. It is unnecessary to include a value for this argument if the column name is "geometry" (default is 'geometry').

...   Other arguments passed on to methods. Not currently used.

Value

[getBasalAreas()](#) returns a data.frame with the columns "Site", "Quad", "Year" and "Species". No two rows will have the same values for all four of these columns. The column "absolute_basalArea" has the area of the quadrat that is occupied by a species in a given unique site/quadrat/year combination. This is in the same units as the area in 'dat'. "quad_basalArea" gives the combined basal area of all plants in a given site/quadrat/year. "percent_basalArea" gives the percentage of occupied basal area within a quadrat that is occupied by each species in a given site/quadrat/year. For example, species A has a basal area of 22 cm^2 in quadrat 1 in 2005 ("absolute_basalArea = 22). In 2005, there are 50 cm^2 of quadrat 1 that are occupied by plants ("quad_basalArea" = 55). 44% of the occupied basal area in quadrat 1 in 2005 is occupied by species A ("percent_basalArea" = 44). There may be an 'NA' in the "percent_basalArea" column if the "quad_basalArea" for that species and year is 0.

See Also

[getLambda()](#), which uses this function to calculate basal areas and ultimately return population growth rates (lambdas) for each species in each quadrat.

Examples

```
dat <- grasslandData[grasslandData$Site == "AZ" &
 grasslandData$Year %in% c(1922:1925),]
names(dat)[1] <- "speciesName"
inv <- grasslandInventory[unique(dat$Quad)]
outDat <- trackSpp(dat = dat,
 inv = inv,
 dorm = 1,
 buff = .05,
 buffGenet = 0.005,
 clonal = data.frame("Species" = unique(dat$speciesName),
 "clonal" = c(TRUE,FALSE)),
 species = "speciesName",
 aggByGenet = TRUE
 )
getBasalAreas(dat = outDat, inv = inv,
species = "speciesName")
```

---

getLambda · *Calculates lambda, the population growth rate, for each species in a quadrat based on changes in basal cover.*

Description

This function calculates the population growth rate (lambda) for every species in a quadrat. This value is the ratio of basal area or number of individuals in the next year to basal area or number of individuals in the current year (basal area in year t+1/ basal area in year t). A lambda value greater than 1 indicates a population is growing, while a value less than 1 indicates population decline. Lambda is 'infinity' when the basal area or number of individuals in year t is 0, and is NA when basal area or number of individuals in year t is zero (i.e. when there are no plants present in year t). Note that a lambda value is calculated between of the years when a quadrat was sampled, even if there is a gap in sampling. For example, a quadrat is sampled in 1998, 1999, 2001, and 2002 (but skipped in 2000). A lambda value will be calculated for 1998-1999 and 2001-2002, which is a transition from year t to year t+1. However, a lambda value is calculated in the same manner for 1999-2001, which is actually a transition from year t to year t+2. You can remove these values by subsetting the data.frame returned by getLambda() for rows when "Year_tplus1"- "Year_t" is equal to 1.

Usage

```
getLambda(
  dat,
  inv,
  method = "area",
  species = "Species",
  quad = "Quad",
  site = "Site",
  year = "Year",
```

```
    geometry = "geometry",
    ...
)
```

Arguments

dat	An sf data.frame in which each row represents a unique polygon (either a genet or a ramet) in a unique site/quadrat/year combination. A data.frame returned by [trackSpp](#) can be put directly into this function. However, it is not necessary for 'dat' to have demographic data or unique identifiers (i.e. 'trackIDs') assigned. 'dat' must have columns that contain a unique identification for each research site (default name is "Site"), species name (default name is "Species"), quadrat identifier (default name is "Quad"), year of data collection (default name is "Year"), and an s.f 'geometry' column that contains a polygon or multipolygon data type for each individual observation.
inv	The name of each element of the list is a quadrat name in 'dat', and the contents of that list element is a numeric vector of all of the years in which that quadrat (or other unique spatial area) was sampled. Make sure this is the years the quadrat was actually sampled, not just the years that have data in the 'dat' argument! This argument allows the function to differentiate between years when the quadrat wasn't sampled and years when there just weren't any individuals of a species present in that quadrat.
method	A single character argument that determines the method for calculating lambda. The default value is "area", which means that lambda is calculated by comparing total basal area for a given species in year t+1 to year t. If 'method' = "count", then lambda is calculated by comparing total number of individuals for a given species in year t+1 to year t. If each individual in 'dat' is mapped as a point, it is best to use 'method' = "count".
species	An optional character string argument. Indicates the name of the column in 'dat' that contains species name data. It is unnecessary to include a value for this argument if the column name is "Species" (default value is 'Species').
quad	An optional character string argument. Indicates the name of the column in 'dat' that contains quadrat name data. It is unnecessary to include a value for this argument if the column name is "Quad" (default is 'Quad').
site	An optional character string argument. Indicates the name of the column in 'dat' that contains site name data. It is unnecessary to include a value for this argument if the column name is "Site" (default value is 'Site').
year	An optional character string argument. Indicates the name of the column in 'dat' that contains data for year of sampling. It is unnecessary to include a value for this argument if the column name is "Year" (default is 'Year').
geometry	An optional character string argument. Indicates the name of the column in 'dat' that contains sf geometry data. It is unnecessary to include a value for this argument if the column name is "geometry" (default is 'geometry').
...	Other arguments passed on to methods. Not currently used.

Value

This function returns a data.frame with columns containing site, quadrat, and species data, as well as the following columns:

Year_t the 'current' year

absolute_basalArea_t basal area (in the same units as the spatial elements of 'dat') for this species in this quadrat in year 't'

Year_tplus1 the 'next' year

absolute_basalArea_tplus1 basal area (in the same units as the spatial elements of 'dat') for this species in this quadrat in year 't+1'

lambda The population growth rate for this species in this quadrat from year t to year t+1.

See Also

[getBasalAreas()](), used internally in this function, which calculates absolute and relative basal areas for each species in each quadrat for each year of sampling.

Examples

```
dat <- grasslandData[grasslandData$Site == "AZ" &
 grasslandData$Year %in% c(1922:1925),]
names(dat)[1] <- "speciesName"
inv <- grasslandInventory[unique(dat$Quad)]
outDat <- trackSpp(dat = dat,
 inv = inv,
 dorm = 1,
 buff = .05,
 buffGenet = 0.005,
 clonal = data.frame("Species" = unique(dat$speciesName),
 "clonal" = c(TRUE,FALSE)),
 species = "speciesName",
 aggByGenet = TRUE
 )
getLambda(dat = outDat, inv = inv, method = "area",
species = "speciesName")
```

---

getNeighbors   *Calculates local neighborhood density around each individual in a*
                                      mapped dataset

Description

This function calculates the density of individuals around each distinct individual in a mapped dataset. It is intended for use on a dataset that has been returned by the [trackSpp]() function, but will work with any sf data.frame where each individual row represents a distinct individual (genet) in a unique year.

Usage

```
getNeighbors(
  dat,
  buff,
  method,
  compType = "allSpp",
  output = "summed",
  trackID = "trackID",
  species = "Species",
  quad = "Quad",
  year = "Year",
  site = "Site",
  geometry = "geometry",
  ...
)
```

Arguments

dat   An sf data.frame. Each row must represent a unique individual organism in a unique year. This argument can be a data.frame that is returned by the [trackSpp](#) function. It must have columns that contain a unique identification for each research site (default name is "Site"), species name (default name is "Species"), quadrat identifier (default name is "Quad"), year of data collection (default name is "Year"), genet identity (default name is "trackID"), and an s.f 'geometry' column that contains a polygon or multipolygon data type for each individual observation.

buff   A numeric value that is greater than or equal to zero. This indicates the distance (in the same units as the spatial data in 'dat') around each focal individual within which you want to look for competitors. OR buff can be a data.frame with the columns "Species" and "buff". This data.frame must have a row for each unique species present in 'dat', with species name as a character string in the "Species" column, and a numeric value in the 'buff' column you'd like to use for that species.

method   A character string, either 'count' or 'area'. This argument determines which method is used to calculate local neighborhood density. If 'method' = 'count', then the number of other individuals within the buffer will be returned in the 'neighbors' column. If method = 'area', then the proportion of the buffer area that is occupied by other individuals will be returned in the 'neighbors' column. If the data in 'dat' was mapped initially as points, it is best to use 'method' = 'count'. If the data was mapped as polygons that are representative of individual basal area, using 'method' = 'area' is likely a more accurate representation of the crowding the focal individual is experiencing.

compType   A character string, either 'allSpp' or 'oneSpp'. If compType = 'allSpp', then local neighborhood density is calculated considering all individuals around the focal individual, regardless of species (interspecific competition). If compType = 'oneSpp', then local neighborhood density is calculated considering only individuals in the buffer area that are the same species as the focal individual (intraspecific competition).

output   A character string, either 'summed' or 'bySpecies'. The default is 'summed'.
                  This argument is only important to consider if you are using compType = 'all-
                  Spp'. If output = 'summed', then only one count/area value is returned for each
                  individual. This value is the total count or area of all neighbors within the fo-
                  cal species buffer zone, regardless of species. If output = 'bySpecies', there
                  is a count or area value returned for each species present in the buffer zone.
                  For example, there are 15 individuals inside a buffer zone. Five are species A,
                  three are species B, and 7 are species C. If output = 'summed', then the 'neigh-
                  bors_count' column in the output data.frame will have the single value '15' in
                  the row for this focal individual. However, if output = 'bySpecies', the row for
                  this focal individual in the output data.frame will contain a named list in the
                  'neighbors_count' column that looks like the one below. If 'method' = 'area'
                  and 'output' = 'bySpecies', a similar list will be returned, but will be in the
                  'neighbors_area' column and will contain areas rather than counts.

                  ```
                  list("Species A "= 5, "Species B" = 3, "Species C" = 7)
                  #> $`Species A `
                  #> [1] 5
                  #>
                  #> $`Species B`
                  #> [1] 3
                  #>
                  #> $`Species C`
                  #> [1] 7
                  ```

trackID   An optional character string argument. Indicates the name of the column in
                  'dat' that contains a value that uniquely identifies each individual/genet. It is
                  unnecessary to include a value for this argument if the column name is 'trackID'
                  (the default value is 'trackID').

species   An optional character string argument. Indicates the name of the column in
                  'dat' that contains species name data. It is unnecessary to include a value for
                  this argument if the column name is "Species" (default value is 'Species').

quad   An optional character string argument. Indicates the name of the column in
                  'dat' that contains quadrat name data. It is unnecessary to include a value for
                  this argument if the column name is "Quad" (default is 'Quad').

year   An optional character string argument. Indicates the name of the column in 'dat'
                  that contains data for year of sampling. It is unnecessary to include a value for
                  this argument if the column name is "Year" (default is 'Year').

site   An optional character string argument. Indicates the name of the column in
                  'dat' that contains site name data. It is unnecessary to include a value for this
                  argument if the column name is "Site" (default value is 'Site').

geometry   An optional character string argument. Indicates the name of the column in
                  'dat' that contains sf geometry data. It is unnecessary to include a value for this
                  argument if the column name is "geometry" (default is 'geometry').

...   Other arguments passed on to methods. Not currently used.

Details

This function draws a buffer around each individual genet of a distance specified by the user. Then it either counts the number of other genets within that buffer, or calculates the proportion of that buffer area that is occupied by other individuals. getNeighbors can calculate either interspecific local neighborhood density (between the focal individual and all other individuals within the buffer area) or intraspecific local neighborhood density (between the focal individual and other individuals of the same species within the buffer area).

Value

This function returns a data.frame with the same number of rows as 'dat' and all of the same columns, but with an additional column or columns. If method = 'count', then a column called "neighbors_count" is added, which contains a count of the number of individuals within the buffer area that is occupied by other individuals. If method = 'area', two columns are added. The first is called "nBuff_area", which contains the area of the buffer around the focal individual. The second is called "neighbors_area", which contains the area of the individuals within the buffer zone around the focal individual.

See Also

The `trackSpp()` function returns a data.frame that can be input directly into this function. If a data.frame is not aggregated by genet such that each unique genet/year combination is represented by only one row, then passing it through the `aggregateByGenet()` function will return a data.frame that can be input directly into this function.

Examples

```
dat <- grasslandData[grasslandData$Site == c("AZ") &
 grasslandData$Species %in% c("Bouteloua rothrockii",
 "Calliandra eriophylla"),]
names(dat)[1] <- "speciesName"
inv <- grasslandInventory[unique(dat$Quad)]
outDat <- trackSpp(dat = dat,
 inv = inv,
 dorm = 1,
 buff = .05,
 buffGenet = 0.005,
 clonal = data.frame("Species" = unique(dat$speciesName),
 "clonal" = c(TRUE)),
 species = "speciesName",
 aggByGenet = TRUE
 )

 finalDat <- getNeighbors(dat = outDat,
 buff = .15,
 method = 'count',
 compType = 'oneSpp',
 species = "speciesName")
```

---

getRecruits *Calculates the number of recruits of each species per year in each quadrat*

Description

This function calculates the number of new plant recruits of each species in each quadrat in each year. The input data must already contain a column indicating whether each observation is classified as a recruit or not. This recruit status can be generated from the [trackSpp](#) function in plantTracker, or can be information that was collected in the field. This function includes an option that determines whether each ramet of a clonal species is considered an individual recruit, or if the entire genet is considered a single recruit.

Usage

```
getRecruits(
  dat,
  byGenet = TRUE,
  species = "Species",
  quad = "Quad",
  site = "Site",
  year = "Year",
  trackID = "trackID",
  recruit = "recruit",
  ...
)
```

Arguments

dat	An sf data.frame in which each row represents a unique polygon (either a genet or a ramet) in a unique site/quadrat/year combination. A data.frame returned by [trackSpp](#) can be put directly into this function. 'dat' must have columns that contain a unique identification for each research site (default name is "Site"), species name (default name is "Species"), quadrat identifier (default name is "Quad"), year of data collection (default name is "Year"), a unique identifier for each genet (default name is 'trackID'), and an s.f 'geometry' column that contains a polygon or multipolygon data type for each individual observation.
byGenet	A logical argument. TRUE indicates that a new genet will be considered as only one recruit, even if it consists of multiple ramets. FALSE indicates that each new ramet will be considered as a new recruit, even if other ramets of the same genet were present in previous years.
species	An optional character string argument. Indicates the name of the column in 'dat' that contains species name data. It is unnecessary to include a value for this argument if the column name is "Species" (default value is 'Species').
quad	An optional character string argument. Indicates the name of the column in 'dat' that contains quadrat name data. It is unnecessary to include a value for this argument if the column name is "Quad" (default is 'Quad').

site   An optional character string argument. Indicates the name of the column in 'dat' that contains site name data. It is unnecessary to include a value for this argument if the column name is "Site" (default value is 'Site').

year   An optional character string argument. Indicates the name of the column in 'dat' that contains data for year of sampling. It is unnecessary to include a value for this argument if the column name is "Year" (default is 'Year').

trackID   An optional character string argument. Indicates the name of the column in 'dat' that contains unique identifiers for each genet. It is unnecessary to include a value for this argument if the column name is "trackID" (default is 'trackID')

recruit   An optional character string argument. Indicates the name of the column in 'dat' that contains information indicating whether or not this row represents data for a recruit. It is unnecessary to include a value for this argument if the column name is "recruit" (default is "recruit").

...   Other arguments passed on to methods. Not currently used.

Value

This function returns a table with columns for site, quadrat, species name, year, and number of recruits.

This function returns a data frame with the columns 'Site', 'Quad', 'speciesName', 'Year', and 'recruits'. The 'recruits' column contains a count of the number of individuals of the species and in the site, quadrat, and year indicated in the other columns of the data frame.

Examples

```
dat <- grasslandData[grasslandData$Site == c("AZ") &
 grasslandData$Species %in% c("Bouteloua rothrockii",
 "Calliandra eriophylla") &
 grasslandData$Year %in% c(1922:1925),]
names(dat)[1] <- "speciesName"
inv <- grasslandInventory[unique(dat$Quad)]
outDat <- trackSpp(dat = dat,
 inv = inv,
 dorm = 1,
 buff = .05,
 buffGenet = 0.005,
 clonal = data.frame("Species" = unique(dat$speciesName),
 "clonal" = c(TRUE,FALSE)),
 species = "speciesName",
 aggByGenet = TRUE
 )
getRecruits(dat = outDat,
 byGenet = TRUE,
 species = "speciesName"
 )
```

---

grasslandData	*Chart Quadrat Dataset for grassland species in MT and AZ An example dataset that contains data from mapped plant chart quadrats. Each row is an observation of one plant.*

Description

Chart Quadrat Dataset for grassland species in MT and AZ An example dataset that contains data from mapped plant chart quadrats. Each row is an observation of one plant.

Usage

```
grasslandData
```

Format

An sf data frame with 1242 rows and 15 variables:

Species A character string indicating the specific epithet for this observation. The ideal formats are the 1) the latin binomial separated by a space (i.e. "Bouteloua gracilis"), 2) the latin binomial separated by an underscore (i.e. "Bouteloua_gracilis"), or 3) a species code consisting of the first n' letters of each element of the latin binomial (i.e. "BOUGRA" or "BOGR"). Whichever format is used, it must be used for all rows.

Clone An integer vector. (1) indicates that this individual is a clone (genet). (0) indicates that this individual is not a clone. NA indicates that this data was not recorded.

Seedling An integer vector indicating whether this observation was a seedling (1), or not (0). This data was not collected for many observations, which have an NA in this column.

Stems An integer greater than 0 indicating how many stems were present for the observed plant. This data was not collected for many observations, which have an NA in this column.

Basal An integer vector. (1) indicates that the basal area of this individual was mapped. (0) indicates that the basal area of this individual was not mapped (in which case it cannot be used in the `assign` function.) NA indicates that this data was not recorded.

Type A factor with two levels: 'point' or 'poly', which indicates whether the observation was originally mapped as a 'point' or a 'polygon'.

Site A character string indicating the sampling location of this observation. In this dataset, this value is either:

- AZ (Santa Rita Experimental Range, Tucson, AZ)
- MT (Fort Keogh Livestock and Range Research Laboratory, Miles City, MT)

Quad A character string indicating the name of the quadrat where this plant was mapped.

Year An integer indicating the year when this plant was mapped.

sp_code_4 A character string of the 4-letter species code for this observation (The first two letters of the genus name, and first two letters of the species name, capitalized with no spaces).

sp_code_6 A character string of the 6-letter species code for this observation (The first three letters of the genus name, and first three letters of the species name, capitalized with no spaces).

Area A numeric variable indicating the basal area in m^2 of this observation. In this dataset, obser-
vations originally mapped as points were assigned an area of 2.46e-05 m^2. For individuals
mapped as polygons, the value in this column indicates the basal area of that plant.

geometry The sf geometry of class "MULTIPOLYGON" for this observation. The location of this
polygon indicates the plant's location in a 1 m x 1 m square. If the plant was mapped as a
polygon, the area of this polygon indicates the basal area of that plant. ...

Source

doi:dx.doi.org/10.1890/112200.1
doi:doi.org/10.6084/m9.figshare.c.3304113.v1

---

grasslandInventory	*A dataset indicating sampling years for the grasslandData example dataset*

Description

This dataset indicates years in which sampling was conducted for each of the four quadrats included
in the `grasslandData` example dataset.

Usage

```
grasslandInventory
```

Format

A named list with 8 items, each of which if named for a quadrat included in the grasslandData
dataset.

C5 An integer vector indicating the years in which quadrat C5 was mapped.

D1 An integer vector indicating the years in which quadrat D1 was mapped.

SG2 An integer vector indicating the years in which quadrat SG2 was mapped.

SG4 An integer vector indicating the years in which quadrat SG4 was mapped. ...

Source

doi:dx.doi.org/10.1890/112200.1
doi:doi.org/10.6084/m9.figshare.c.3304113.v1

---

groupByGenet   *Group polygon observations together into a 'genet' based on proximity*

Description

This function assigns a unique 'genetID' to polygons if they are within a user-defined distance from one another. Each ramet is still represented by a single row of data, but all ramets of the same genet have the same 'genetID'.

Usage

```
groupByGenet(dat, buffGenet, ...)
```

Arguments

dat   An sf object that contains polygons to be grouped. Typically should include data
                only for one species, one quadrat, and one year. For intended use, this dataset
                should be of the format described in grasslandData. This function will run if
                'dat' contains only 'geometry' data, but it is *strongly* recommended that other
                columns specified in grasslandData are included.

buffGenet   A numeric argument indicating half of the maximum distance between individ-
                ual ramets (polygons) that the function will group together as one genet. Note
                that this argument is in the same units that are used in the 'dat' argument. For
                example, if buffGenet = 0.005 and we use the grasslandData (in which mea-
                surements are in meters), then polygons that are 0.01 m (1 cm) apart or less will
                be grouped as a genet.

...   Other arguments passed on to methods. Not currently used.

Details

If polygons are 'grouped,' they are given the same number in the 'genetID' column. This assign-
ment is made using network analysis to group together polygons that are closest to one another.
In the context of **plantTracker**, this function was designed to group ramets (vegetative clones) to-
gether into one genet (genetically distinct individual). This function was designed for use within
the assign function (and then the trackSpp function), but can be used independently.

Value

A numeric vector of unique genetIDs that is as long as the number of rows in 'dat.' Each element i
in the output is the genetID for the ith row in 'dat'.

See Also

assign() and trackSpp(), **plantTracker** functions that apply this function across multiple species,
quadrats, and years.

Examples

```
dat <- grasslandData[grasslandData$Site == "AZ" &
grasslandData$Quad == "SG2" &
grasslandData$Species == "Bouteloua rothrockii" &
grasslandData$Year == 1922,]

groupByGenet(dat = dat, buffGenet = 0.01)
groupByGenet(dat = dat, buffGenet = 0.1)
```

---

trackSpp   *Tracks genets through time for multiple species and sites*

Description

This function tracks individual plants in mapped quadrats through time to generate a demographic dataset that includes survival and growth for each individual.

Usage

```
trackSpp(
  dat,
  inv,
  dorm,
  buff,
  buffGenet,
  clonal,
  species = "Species",
  site = "Site",
  quad = "Quad",
  year = "Year",
  geometry = "geometry",
  aggByGenet = TRUE,
  printMessages = TRUE,
  flagSuspects = FALSE,
  shrink = 0.1,
  dormSize = 0.05,
  ...
)
```

Arguments

dat   An sf data.frame of the same format as [grasslandData](). It must have columns
                that contain a unique identification for each research site (default name is "Site"),
                species name (default name is "Species"), quadrat identifier (default name is
                "Quad"), year of data collection (default name is "Year") and an s.f 'geome-
                try' column that contains a polygon or multipolygon data type for each indi-
                vidual observation. This function will add columns called "basalArea_ramet",

	"trackID", "age", "size_tplus1", "recruit," "nearEdge," and "survives_tplus1", so 'dat' should not contain columns with these names.
inv	A named list of the same format as [grasslandInventory](). The name of each element of the list is a quadrat name in 'dat', and the contents of that list element is a numeric vector of all of the years in which that quadrat (or other unique spatial area) was sampled. Make sure this is the years the quadrat was actually sampled, not just the years that have data in the 'dat' argument! This argument allows the function to differentiate between years when the quadrat wasn't sampled and years when there just weren't any individuals of a species present in that quadrat.
dorm	A numeric vector of length 1, indicating the number of years an individual of these species is allowed to go dormant, i.e. be absent from the map but be considered the same individual when it reappears. This must be an integer greater than or equal to 0. *OR* dorm can be a data.frame with the columns "Species" and "dorm". This data.frame must have a row for each unique species present in 'dat', with species name as a character string in the "Species" column, and a numeric value greater than or equal to 0 in the 'dorm' column that indicates the number of years each species is allowed to go dormant.
buff	A numeric vector of length 1 that is greater than or equal to zero, indicating how far (in the same units as spatial values in 'dat') a polygon can move from year i to year i+1 and still be considered the same individual. *OR* buff can be a data.frame with the columns "Species" and "buff". This data.frame must have a row for each unique species present in 'dat', with species name as a character string in the "Species" column, and a numeric value in the 'buff' column specifying the 'buff' argument for each species.
buffGenet	A numeric vector of length 1 that is greater than or equal to zero, indicating how close (in the same units as spatial values in 'dat') polygons must be to one another in the same year to be grouped as a genet (if 'clonal' argument = TRUE). *OR* buffGenet can be a data.frame with the columns "Species" and "buffGenet". This data.frame must have a row for each unique species present in 'dat', with species name as a character string in the "Species" column, and a numeric value greater than or equal to 0 in the 'buffGenet' specifying the 'buffGenet' argument for each species. This argument is passed to the [groupByGenet]() function, which is used inside the [assign]() function.
clonal	A logical vector of length 1, indicating whether a species is allowed to be clonal or not (i.e. if multiple polygons (ramets) can be grouped as one individual (genet)). If clonal = TRUE, the species is allowed to be clonal, and if clonal = FALSE, the species is not allowed to be clonal. *OR* clonal can be a data.frame with the columns "Species" and "clonal". This data.frame must have a row for each unique species present in 'dat', with species name as a character string in the "Species" column, and a logical value in the 'clonal' specifying the 'clonal' argument for each species.
species	An optional character string argument. Indicates the name of the column in 'dat' that contains species name data. It is unnecessary to provide a value for this argument if the column name is "Species" (default value is 'Species').
site	An optional character string argument. Indicates the name of the column in 'dat' that contains site name data. It is unnecessary to provide a value for this

	argument if the column name is "Site" (default value is 'Site').
quad	An optional character string argument. Indicates the name of the column in 'dat' that contains quadrat name data. It is unnecessary to provide a value for this argument if the column name is "Quad" (default is 'Quad').
year	An optional character string argument. Indicates the name of the column in 'dat' that contains data for year of sampling. It is unnecessary to provide a value for this argument if the column name is "Year" (default is 'Year').
geometry	An optional character string argument. Indicates the name of the column in 'dat' that contains sf geometry data. It is unnecessary to provide a value for this argument if the column name is "geometry" (default is 'geometry').
aggByGenet	A logical argument that determines whether the output of trackSpp() will be aggregated by genet. If the value is TRUE (the default), then each unique trackID (or genet) in each year will be represented by only one row in the output data.frame. This prepares the dataset for most demographic analyses. If the value is FALSE, then each unique trackID in each year may be represented by multiple rows in the data (each ramet gets a row). Note that if the value is TRUE, then some columns present in the input data.frame 'dat' will be dropped. If you do not wish this to happen, then you can aggregate the data.frame to genet by hand.
printMessages	A logical argument that determines whether this function returns messages about genet aggregation, as well as messages indicating which year is the last year of sampling in each quadrat and which year(s) come before a gap in sampling that exceeds the 'dorm' argument (and thus which years of data have an 'NA' for "survives_tplus1" and "size_tplus1"). If printMessages = TRUE (the default), then messages are printed. If printMessages = FALSE, messages are not printed.
flagSuspects	A logical argument of length 1, indicating whether observations that are 'suspect' will be flagged. The default is flagSuspects = FALSE. If flagSuspects = TRUE, then a column called 'Suspect' is added to the output data.frame. Any suspect observations get a 'TRUE' in the 'Suspect' column, while non-suspect observations receive a 'FALSE'. There are two ways that an observation can be classified as 'suspect'. First, if two consecutive observations have the same trackID, but the basal area of the observation in year t+1 is less that a certain percentage (defined by the shrink arg.) of the basal area of the observation in year t, it is possible that the observation in year t+1 is a new recruit and not the same individual. The second way an observation can be classified as 'suspect' is if it is very small before going dormant. It is unlikely that a very small individual will survive dormancy, so it is possible that the function has mistakenly given a survival value of '1' to this individual. A 'very small individual' is any observation with an area below a certain percentile (specified by 'dormSize') of the size distribution f or this species, which is generated using all of the size data for this species in 'dat'.
shrink	A single numeric value. This value is only used when flagSuspects = TRUE. When two consecutive observations have the same trackID, and the ratio of size t+1 to size t is smaller than the value of shrink, the observation in year t gets a 'TRUE' in the 'Suspect' column. For example, shrink = 0.2, and an individual that the tracking function has identified as 'BOUGRA_1992_5' has an area of 9 cm^2 in year t and an area of 1.35 cm^2 in year t+1. The ratio of size t+1 to

size t is 1.35/9 = 0.15, which is smaller than the cutoff specified by shrink, so
the observation of BOUGRA_1992_5' in year t gets a 'TRUE' in the 'Suspect'
column. The default value is shrink = 0.10.

dormSize   A single numeric value. This value is only used when flagSuspects = TRUE
                    and dorm is greater than or equal to 1. An individual is flagged as 'suspect'
                    if it 'goes dormant' and has a size that is less than or equal to the percentile
                    of the size distribution for this species that is designated by dormSize. For
                    example dormSize = 0.05, and an individual has a basal area of 0.5 cm^2. The
                    5th percentile of the distribution of size for this species, which is made using
                    the mean and standard deviation of all observations in 'dat' for the species in
                    question, is 0.6 cm^2. This individual does not have any overlaps in the next
                    year (year t+1), but does have an overlap in year t+2. However, because the
                    basal area of this observation is smaller than the 5th percentile of size for this
                    species, the observation in year t will get a 'TRUE' in the 'Suspect' column. It
                    is possible that the tracking function has mistakenly assigned a '1' for survival
                    in year t, because it is unlikely that this individual is large enough to survive
                    dormancy. The default value is dormSize = .05.

...   Other arguments passed on to methods. Not currently used.

Details

This is a wrapper function that applies [assign](#) across multiple species, quadrats, and sites. For each
species and quadrat, [trackSpp()](#) loads a spatially referenced data.frame ('dat'), and then uses the
[groupByGenet](#) function to assign genetIDs to polygons (if 'clonal' = TRUE) such that polygons
that form the same genet have the same genetID. A buffer of a distance defined by 'buff' is applied
around each genet polygon. Then, the spatial data for each genet from the current year (year t) is
compared to individuals in the next year (year t+1). Then [trackSpp()](#) calculates the amount of
overlapping area between polygons of each year t genet and polygons of each year t+1 genet (using
[st_intersection](#)). If there is unambiguous overlap between a 'parent' genet from year t and a
'child' genet from year t+1, then that 'child' gets the same identifying trackID as the parent. If
there is a 'tie,' where more than one parent overlaps the same child or more than one child overlaps
the same parent, the parent-child pair with the greatest amount of overlap receives the same trackID.
Polygons in year t+1 that do not have a parent are given new trackIDs and are identified as new
recruits. If dormancy is not allowed, then polygons in year t that do not have child polygons get a '0'
in the 'survival' column. If dormancy is allowed, parent polygons without child polygons are stored
as 'ghosts' and are then compared to data from year t+1+i to find potential child polygons, where
i='dorm' argument. For a more detailed description of the [trackSpp()](#) function, see the vignette:
vignette("Using_the_plantTracker_trackSpp_function",package = "plantTracker")

Value

An sf data.frame with the same columns as 'dat,' but with the following additional columns:

trackID   A unique value for each individual genet, consisting of the 6-letter species code,
                    the year in which this individual was recruited, and a unique index number, all
                    separated by a "_".

age   An integer indicating the age of this individual in year t. Values of NA indicate
                    that an accurate age cannot be calculated because this individual was observed

either in the first year of sampling or in a year following a gap in sampling, so the exact year of recruitment is not known.

size_tplus1   The size of this **genet** in year t+1, in the same units as the 'area' column in 'dat'.

recruit   A Boolean integer indicating whether this individual is a new recruit in year t (1), or existed in a previous year (0). Values of NA indicate that this individual was observed either in the first year of sampling or in a year following a gap in sampling, so it is not possible to accurately determine whether or not it is a new recruit in year t.

survives_tplus1

A Boolean integer indicating whether this individual survived (1), or died (0) in year t+1.

basalArea_genet

The size of this entire genet in year t, in the same units as the 'area' column in 'dat.' If the 'clonal' argument = FALSE, then this number will be identical to the 'basalArea_ramet' column.

basalArea_ramet

This is only included if 'aggByGenet' = FALSE. This is the size of this ramet in year t, in the same units as the 'area' column in 'dat'. If the 'clonal' argument = FALSE , then this number will be identical to the 'basalArea_genet' column.

nearEdge   A logical value indicating whether this individual is within a buffer (specified by the 'buff' argument) from the edge of the quadrat.

See Also

[assign()](), which is used inside the [trackSpp()]() function. [trackSpp()]() applies the [assign()]() function over multiple species and quadrats, and uses the [aggregateByGenet()]() function to aggregate the demographic results by genet (if 'aggByGenet' = TRUE). The [assign()]() function uses the [groupByGenet()]() function to group ramets into genets (if 'clonal' argument = TRUE).

Examples

```
dat <- grasslandData[grasslandData$Site == c("AZ") &
 grasslandData$Species %in% c("Bouteloua rothrockii",
  "Calliandra eriophylla"),]
names(dat)[1] <- "speciesName"
inv <- grasslandInventory[unique(dat$Quad)]
outDat <- trackSpp(dat = dat,
 inv = inv,
 dorm = 1,
 buff = .05,
 buffGenet = 0.005,
 clonal = data.frame("Species" = unique(dat$speciesName),
 "clonal" = c(TRUE,FALSE)),
 species = "speciesName",
 aggByGenet = TRUE
 )
```

Index