plantTracker: An R package to translate maps of plant occurrence into demographic data

Alice E. Stears[1*], Peter B. Adler[2], Shannon E. Albeke[3], David H. Atkins[1], Jared Studyvin[4], Daniel C. Laughlin[1]

[1]Botany Department and Program in Ecology, University of Wyoming, Laramie, WY;

[2]Department of Wildland Resources and the Ecology Center, Utah State University, Logan, UT;

[3]Wyoming Geographic Information Science Center, University of Wyoming, Laramie, WY;

[4]Department of Mathematics and Statistics, University of Wyoming, Laramie, WY

[*]Corresponding Author: astears@uwyo.edu

Abstract

1. Long-term demographic data are rare yet invaluable for conservation, management, and basic research on the underlying mechanisms of population and community dynamics. Historical and contemporary mapped datasets of plant location and basal area present a relatively untapped source of demographic records that, in some cases, span over 20 years of sequential data collection. However, these maps do not uniquely mark individual plants, making the process of collecting growth, survival, and recruitment data difficult.

2. Recent efforts to translate historical maps of plant occurrence into shapefiles make it possible to use computer algorithms to track individuals through time and determine individual growth and survival. We summarize the plantTracker R package, which contains user-friendly functions to extract neighborhood density, growth, and survival data from repeatedly-sampled maps of plant location and basal area. These functions can be used with data derived from quadrat maps, aerial photography, and remote sensing, and while designed for use with perennial plants, can be applied to any repeatedly mapped sessile organism.

3. This package contains two primary functions: `trackSpp()`, which tracks individuals through time and assigns demographic data, as well as `getNeighbors()`, which calculates both within and between-species neighborhood occupancy around each mapped individual. `plantTracker` also contains functions to estimate plot-level recruitment, calculate plot-level population growth rate, and create quadrat maps.

4. We tested the accuracy of the `trackSpp()` function on two spatial demographic datasets. The function was nearly perfect at assigning individual identities and survival status when tested on maps of tree basal area and perennial forb point locations. In both cases, the function correctly assigned survival and recruitment with 99% accuracy. These accurate and precise functions will expand the amount of data available to investigate demographic processes, which are fundamental drivers of population, community, and ecosystem processes.

Key-words: demography, quadrat maps, R, neighborhood density, perennial plants

1 | Introduction

Long-term observations of plant demographic rates are uncommon yet necessary for answering many pressing ecological questions. Demographic rates, including growth, survival, and reproduction, are the primary mechanisms underlying individual and population responses to the abiotic environment, and understanding them is important for both theoretical research and practical applications. For example, the success of conservation efforts requires a clear understanding of the sensitivity of each demographic rate to changes in the environment (Crone et al., 2011).

One challenge is that collecting data to calculate demographic rates is time-consuming and expensive. Typically, each individual must be tagged with a unique marker, mapped, and measured annually for a minimum of two years (Caswell, 2001). The value of demographic data typically increases with the number of consecutive years of data collection, yet demographic datasets that span long periods of time are uncommon (Lindenmayer et al., 2012). However, demographic data collection is time-consuming and tedious even in the short-term, and longer studies face many obstacles ranging from shortage of funding to lack of personnel.

However, computer algorithms make it possible to retroactively extract growth and survival information by comparing spatially-explicit datasets that show the location and size of organisms across years. Datasets of this nature range from high-resolution aerial photographs showing woody plant location and size, to hand-drawn maps of herbaceous plants in 1-meter quadrats. An example of a particularly extensive source of spatial data are chart quadrat maps of plant basal areas, generated by a method proposed by Clements (1907) to track plant communities through time. The "chart quadrat" is a permanent, 1 m^2 quadrat within which the basal cover and species identity of every individual plant is mapped , either as a polygon that represents its basal area, or as a point if the plant has stems with negligeable basal areas (Hill, 1920; White, 1985). Annual re-sampling of chart quadrats generates maps that show how overall cover and location of individuals has shifted over the sampling period (e.g., Fig. 1). Scientists established (and are still establishing) hundreds of chart quadrats throughout western North America, beginning in the early 20th century (Hill, 1920). Historic chart quadrat maps from seven sites have been recovered and digitized into shapefiles (Adler, Tyburczy, & Lauenroth, 2007; Zachmann, Moffet, & Adler, 2010; Anderson, Vermeire, & Adler, 2011; Anderson, McClaran, & Adler, 2012; Chu et al., 2013; Christensen et al., 2021; Moore et al., 2022), and are being used to test ecological theory (Laughlin, Moore, & Fulé, 2011; Chu & Adler, 2015).

The major difference between traditional demographic monitoring methods and map data such as chart quadrats is that individual plants are not uniquely tagged in map data. However, the spatially explicit nature of maps can be leveraged to extract individual rates of growth and survival, and plot-level rates of seedling recruitment. If a plant of the same species occurs at approximately the same location from year to year, we can assume it is the same individual and assign it values for survival and growth. This process of tracking unmarked, mapped individuals to generate demographic data can also be used in any scenario where sessile organisms are mapped annually. Additionally, unlike traditional tagged demographic data, mapped datasets typically include information for multiple species, making it possible to generate individual-level estimates of neighborhood density. These can be used to understand how competition within and between species affects individual plant fitness, population dynamics and vegetation patterns (Chu & Adler, 2015; Kunstler et al., 2016).

Leveraging maps to generate demographic data and neighborhood occupancy opens many exciting possibilities. We can now use spatial data such as long-term chart quadrat maps to ask questions about plant demographic processes over long timescales. For example, chart quadrat maps from the Jornada Experimental Range yield growth and survival rates for 22 consecutive annual transitions (Christensen et al., 2021). A study of this duration would be extremely time-consuming to conduct using traditional demographic collection methods. These long-term demographic datasets allow us to determine how growth, survival, and recruitment have shifted in response to environmental change. Additionally, some chart quadrat locations were sampled as early as 1915, making them ideal reference points against which to compare modern plant communities and populations.

Here, we describe `plantTracker` (Stears, Studyvin, & Atkins, 2022), a package for the `R` statistical software available for download through GitHub (R Core Team, 2021). plantTracker contains functions to extract competition, growth, survival, and recruitment information from digitized maps of plant occurrence. This package provides users with a straightforward set of tools to fully leverage the potential of fine-scale maps of plant occurrence and size, as well as any other data where sessile organisms are mapped across time.

2 | **plantTracker**

plantTracker depends on the "sf" (Pebesma, 2018), "Matrix" (Bates & Maechler, 2019), and "igraph" (Csardi & Nepusz, 2006) R packages, and contains two primary functions, trackSpp() and getNeighbors(). plantTracker also has functions to generate plot-level metrics such as recruitment, species basal area, and population growth rate (Table 1), which are described in the "Suggested plantTracker Workflow" package vignette. Tracking algorithms that generate demographic data from maps were first implemented by Lauenroth and Adler (2008), but their code was not published or easily generalizable. While plantTracker consists of entirely new code, our general approach was inspired by this previous work.

Input Data

Although designed for use with chart quadrat maps, plantTracker can be used with any dataset where organisms are mapped and identified to species across at least two time points. plantTracker functions can be used with a variety of data sources, but the data itself must follow a specific format. The main input data frame, or "dat″, must be an "sf" spatial data frame in which each row represents one observation or individual in a single year, and contains a "polygon" or "mulitpolygon" geometry corresponding to the location of that observation. Although not required, it is ideal if this geometry also indicates the basal area of the observation. If this is not the case, plantTracker functions will only produce faithful estimates of survival, not growth. Observations originally mapped as points must be converted to small polygons of negligible area. `trackSpp()` and `getNeighbors()` also require an “`inv`” argument, short for “inventory,” which is a named list indicating the years in which each quadrat was sampled.

`trackSpp()` Function

The `trackSpp()` function overlays maps of quadrats from sequential timesteps and assigns overlapping individuals the same unique identifier. The function assumes an annual sampling interval, but this is not strictly necessary. It then uses these identifiers, or "trackIDs", to assign values to individuals indicating survival to the next year (0 = death, or 1 = survival), size in the next year, a value indicating whether it is a recruit (0 = not recruit/1 = recruit), and age. Recruit and age data are not assigned if the observation was recorded in the first year of sampling or followed a gap in sampling. Beyond `dat` and `inv`, there are four additional required arguments to `trackSpp()` that can be defined globally or uniquely for each species.

First, the `buff` argument defines the distance that an individual can “move” from year to year and still be considered the same individual (Fig. 2). This distance accounts both for small errors in mapping, and for true variation in where an organism reoccurs each year. If `buff` is set to zero, a plant must occur in precisely the same location in consecutive years to be considered the same individual. It is important to select biologically realistic values for `buff`, since too-large values can overestimate survival and underestimate recruitment, while too-small values can underestimate survival and overestimate recruitment. Previous analyses of perennial graminoids and forbs in western North American grasslands suggest that defining a buffer of 5 cm leads to the most accurate trackID assignment (Lauenroth & Adler, 2008; Chu et al., 2014).

Second, the `dorm` argument defines the maximum number of years a plant is allowed to go dormant, or disappear, from the map before reappearing and still be considered the same individual (Fig. 3). Again, it is important to select biologically relevant values for dorm. Allowing one to two years of dormancy is reasonable for some forbs, but is unreasonable for shrubs, trees, or large grasses. The dorm argument also helps account for plants being inadvertently missed by mappers and can allow plants to survive through a year when sampling was skipped completely. Otherwise, all plants receive an "NA" for survival in the year before a break in sampling, and tracking starts again after the break. Larger values of dorm will overestimate survival, while smaller values will underestimate survival.

Third, the clonal argument determines whether vegetative reproduction is allowed. If clonal=FALSE, every row of data is considered a distinct genetic individual. If clonal=TRUE, a genet (a genetic individual) can consist of several rows, each of which is a ramet (a vegetative segment of the genet). The groupByGenet() function is used within trackSpp() to group polygons into genets based on their proximity in the first year. A "previous year" genet can pass on its trackID to multiple "current year" polygons that it overlaps, which are then assigned the same trackID. The fourth argument in trackSpp() is buffGenet, which is only required when clonal=TRUE. This determines how close polygons must be to be grouped into one genet by groupByGenet()(Fig. 4).

trackSpp() iterates through dat and inv by site, quadrat, and then species to compare quadrat maps from sequential years. It assigns trackIDs and demographic data to individuals based on overlap from year to year (Fig. 5). Table 2 describes this workflow in greater detail. Further description of the trackSpp() function, including a detailed description of the implications of different buff, dorm, and clonal arguments for the demographic data returned by the function, can be found in the "Using the plantTracker trackSpp() function" package vignette.

getNeighbors() Function

The getNeighbors() function calculates either the number or basal area of competitors in the neighborhood of each individual. This makes it possible to quantify the effect of density dependence, or the response of organisms to the presence and density of neighboring conspecific individuals, which is a fundamental process driving population dynamics (Bertness & Callaway, 1994; Vellend, 2010). Additionally, the proximity of heterospecifics to a focal individual approximates the effect of inter-species interactions on that individual, which can be negative, as in competition, or positive, as in facilitation.

In getNeighbors(), the buff argument specifies the width of the neighborhood buffer drawn around each individual, and can be specified uniquely for each focal species. buff values can be based on empirical observations, or determined by a modeling approach detailed in Adler et al. (2010). The next argument, method, determines how neighborhood occupancy is tabulated. If method=count, the function returns the number of individuals inside the buffer zone. If method=area, the function returns the size of the buffer zone area around that focal individual and the area of the buffer zone area that is occupied by other individuals (Fig. 6). The third argument is compType. If compType=oneSpp, the getNeighbors() function only considers conspecifics of the focal individual. If compType=allSpp (the default) the function considers all other plants in the buffer zone, regardless of species. The final argument is output, which determines whether neighbor occupancy values are summed across species, or given uniquely for each neighbor species. The default, output=summed, means neighbor values are summed, while if output=bySpecies, each row in the column added by getNeighbors() contains a list giving the counts or areas of neighbors in the buffer area by species.

3 | Proof of Concept

We tested the performance of trackSpp() on two mapped datasets in which individuals were tagged following typical demographic monitoring protocols. The first is a large-scale and inventory of tree basal areas over 21 years in La Selva, Panama (Clark & Clark, 2021). The second is a three-year monitoring study of the monocarpic perennial forb *Oenothera coloradensis* in southeast Wyoming and northern Colorado, and includes the point-location of every individual in 18, 2-m^2 quadrats (Stears, unpublished data). We used trackSpp() to generate demographic data for both datasets, and compared the function-generated growth, survival, and recruit data to values measured in the field.

For the tree dataset, we used trackSpp() with dorm=0, buff=0.01 (10 cm), and clonal=FALSE. trackSpp() identified 5,212 unique individuals, the same number of unique trees in the field-collected data. The function accurately attributed survival or death for 99.99% of individuals (Table 3). The function also identified recruit status correctly for 99.7% of individuals. Errors occurred for the same four individuals that had mis-attributions for survival (Table 3). For the O. coloradensis dataset, we used trackSpp() with dorm=0, buff=0.02 (2 cm), and clonal=FALSE. The function identified 3,128 unique individuals, 99.4% of the 3,146 individuals recorded in the field-collected data. The function correctly attributed survival or death for 99.6% of plants. The function also identified recruit status correctly for 99.5% of all plants (Table 4). Mis-attributions occurred in both datasets when two plants (e.g. A and B) were extremely close to one another in the current year and only one (A) survived. Because plants A and B were equally close to one another, the function happened to attribute survival to the wrong plant (B).

4 | Conclusion

plantTracker provides a suite of user-friendly R functions to easily generate demographic and neighborhood occupancy data from fine-scale maps of plants that do not uniquely identify individuals. These tools allow us to translate both historical and contemporary map datasets into growth, survival, and recruitment information, expanding the amount of demographic data at our disposal. Robust and long-term estimation of demographic rates is critical to advancing many disciplines in ecology, and we hope users will find `plantTracker` helpful for filling gaps in our ecological understanding.

Conflict of interest statement: The authors declare no conflicts of interest.

Author contribution statement: All authors contributed to the design of `plantTracker`. Alice Stears developed `plantTracker`, with extensive contributions from Jared Studyvin, Shannon Albeke, and David Atkins and higher-level input from Peter Adler and Daniel Laughlin. Alice Stears wrote the manuscript with contributions from all authors.

Data Availability: `plantTracker` and its source code can be downloaded from GitHub (Stears, Studyvin, & Atkins, 2022) (https://github.com/aestears/plantTracker; https://zenodo.org/record/6609335, DOI: 10.5281). We are submitting `plantTracker` to the Comprehensive `R` Archive Network (CRAN), where it will also be available for download in the future.

References

Adler, P. B., Ellner, S. P., & Levine, J. M. (2010). Coexistence of perennial plants: An embarrassment of niches. *Ecology Letters*, *13*(8), 1019–1029. doi:10.1111/j.1461-0248.2010.01496.x

Adler, P. B., Tyburczy, W. R., & Lauenroth, W. K. (2007). Long-term mapped quadrats from Kansas prairie: Demographic information for herbaceous plants. *Ecology*, *88*(10), 2673. doi:10.1890/0012-9658(2007)88[2673:LMQFKP]2.0.CO;2

Anderson, J., McClaran, M. P., & Adler, P. B. (2012). Cover and density of semi-desert grassland plants in permanent quadrats mapped from 1915 to 1947. *Ecology*, *93*(6), 1492. doi:10.1126/science.127.3295.414-b

Anderson, J., Vermeire, L., & Adler, P. B. (2011). Fourteen years of mapped, permanent quadrats in a northern mixed prairie, USA: Ecological Archives E092-143. *Ecology*, *92*(8), 1703. doi:10.1126/science.127.3295.414-b

Bates, D., & Maechler, M. (2019). Matrix: Sparse and dense matrix classes and methods, (R package version 1.2-18). Retrieved from https://cran.r-project.org/package=Matrix

Bertness, M. D., & Callaway, R. (1994). Positive interactions in communities. *TREE*, *9*(5), 191–193.

Caswell, H. (2001). *Matrix Population Models: Construction, Analysis, and Interpretation* (2nd ed.). Sinauer Associates.

Christensen, E., James, D., Maxwell, C., Slaughter, A., Adler, P. B., Havstad, K., & Bestelmeyer, B. (2021). Quadrat-based monitoring of desert grassland vegetation at the Jornada Experimental Range, New Mexico, 1915–2016. *Ecology*, 0–2. doi:10.1002/ecy.3530

Chu, C., & Adler, P. B. (2015). Large niche differences emerge at the recruitment stage to stabilize grassland coexistence. *Ecological Monographs*, *85*(3), 373–392. doi:10.1890/14-1741.1

Chu, C., Havstad, K. M., Kaplan, N., Lauenroth, W. K., McClaran, M. P., Peters, D. P., … Adler, P. B. (2014). Life form influences survivorship patterns for 109 herbaceous perennials from six semi-arid ecosystems. *Journal of Vegetation Science*, *25*(4), 947–954. doi:10.1111/jvs.12106

Chu, C., Norman, J., Flynn, R., Kaplan, Ni., Lauenroth, W. K., & Adler, P. B. (2013). Cover, density, and demographics of shortgrass steppe plants mapped 1997-2010 in permanent grazed and ungrazed quadrats. *Ecology*, *94*(6), 1435.

Clark, D., & Clark, D. (2021). Two decades of annual landscape-scale tree growth and dynamics in old-growth tropical rainforest in the CARBONO Project, La Selva Biological Station, 1997-2018. *Dryad, Dataset*. doi:https://doi.org/10.5061/dryad.51c59zw8n

Clements, F. E. (1907). *Plant Physiology and Ecology*. New York, New York: Henry Holt and Company.

Crone, E. E., Menges, E. S., Ellis, M. M., Bell, T., Bierzychudek, P., Ehrlén, J., … Williams, J. L. (2011). How do plant ecologists use matrix population models? *Ecology Letters*, *14*(1), 1–8. doi:10.1111/j.1461-0248.2010.01540.x

Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal*, *Complex Sy*, 1695.

Hill, R. R. (1920). Charting Quadrats with a Pantograph. *Ecology*, *1*(4), 270–273. Retrieved from https://www.jstor.org/stable/1929561

Kunstler, G., Falster, D., Coomes, D. A., Hui, F., Kooyman, R. M., Laughlin, D. C., … Westoby, M. (2016). Plant functional traits have globally consistent effects on competition. *Nature*, *529*(7585), 204–207. doi:10.1038/nature16476

Lauenroth, W. K., & Adler, P. B. (2008). Demography of perennial grassland plants: Survival, life expectancy and life span. *Journal of Ecology*, *96*(5), 1023–1032. doi:10.1111/j.1365-2745.2008.01415.x

Laughlin, D. C., Moore, M. M., & Fulé, P. Z. (2011). A century of increasing pine density and associated shifts in understory plant strategies. *Ecology*, *92*(3), 556–561. doi:10.1890/09-2103.1

Lindenmayer, D. B., Likens, G. E., Andersen, A., Bowman, D., Bull, C. M., Burns, E., … Wardle, G. M. (2012). Value of long-term ecological studies. *Austral Ecology*, *37*(7), 745–757. doi:10.1111/j.1442-9993.2011.02351.x

Moore, M. M., Jenness, J. S., Laughlin, D. C., Strahan, R. T., Bakker, J. D., Dowling, H. E., & Springer, J. D. (2022). Cover and density of southwestern ponderosa pine understory plants in permanent chart quadrats (2002-2020). *Ecology*, 0–2. doi:10.1002/ecy.3661

Pebesma, E. (2018). Simple Features for R: Standardized Support for Spatial Vector Data. *The R Journal*, *10*(1), 439–446. doi:10.32614/RJ-2018-009

R Core Team. (2021). R: A language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.r_project.org

Stears, A., Studyvin, J., & Atkins, D. (2022). plantTracker. Zenodo. doi:10.5281/zenodo.6326566

Vellend, M. (2010). Conceptual Synthesis in Community Ecology. *The Quarterly Review of Biology*, *85*(2), 183–206. doi:10.1086/652373

White, J. (1985). The census of plants in vegetation. In J. White (Ed.), *The Population Structure of Vegetation.* (pp. 34–88). Dordrecht: Dr W. Junk Publishers.

Zachmann, L., Moffet, C., & Adler, P. (2010). Mapped quadrats in sagebrush steppe: long-term data for analyzing demographic rates and plant–plant interactions. *Ecology*, *91*(11), 3427–3427. doi:10.1890/10-0404.1

Table 1. Descriptions of functions in plantTracker

Function Name	**Description**
`trackSpp()`	track plants through time to determine growth and survival
`getNeighbors()`	calculate neighborhood occupancy
`groupByGenet()`	group polygons into genetic individuals based on proximity
`checkDat()`	Check format of input data
`drawQuadMap()`	draw maps of quadrats
`getBasalAreas()`	calculate the total basal area of each species/quadrat/year
`getLambda()`	calculate plot-level lambda (population growth rate) for each species across each annual transition, where lambda is $\lambda = {N_{t+1}}/{N_t}$ and N is either basal area or number of individuals * It is important to note that λ calculated with data from one plot is not necessarily a good metric of a population's growth rate, since a single plot may not encompass the entire spatial extent of that population. It may be necessary to compile data from several plots to accurately estimate λ.
`getRecruits()`	calculate the number of recruits of each species/quadrat/year
`aggregateByGenet()`	Group a `trackSpp()` output so each row contains data for all ramets of a genet

Table 2. Steps in the `trackSpp()` function. "i" indicates the iteration of the loop that compares sequential years. **Bold** indicates a data frame, and `this font` indicates a package function or argument.

Step	Action	Next Step…
1	Subset **dat** iteratively by each *Site*, *Quadrat*, and *Species*, and subset **inv** by *Quadrat.*	step 2
2	Put the first year of data in **PreviousYr** If `clonal=TRUE`, use `groupByGenet()` to assign each observation a unique trackID.	step 3
3	Is the data in **PreviousYr** from the first year of sampling (the first year in **inv**) ?	*yes*—step 4 *no*—step 6
4	Give these individuals "NA" in "recruit" and "age" cols.	step 6
5	Is the first year of data from the last year of sampling (the last year in **inv**)?	*yes*—step 21 *no*—step 7
6	Compare the previous year (**inv**[*i*-1]) to the current year (**inv**[*i*]). Is the gap between years *greater* than (`dorm` + 1)?	*yes*—step 19 *no*—step 8
7	Put "1" in "recruit" and "age" cols.	step 6
8	Get data observed in the *i*-th year and put it into **CurrentYr**. Does **PreviousYr** contain any observations?	*yes*—step 9 *no*—step 23
9	Add a buffer of width = `buff` around every genet in **PreviousYr** and put the data in **PreviousBuff** Does **CurrentYr** contain any observations?	*yes*—step 10 *no*—step 11
10	Is there any overlap between polygons in **PreviousBuff** and **CurrentYr**?	*yes*—step 14 *no*—take **PreviousYr** to step 11 and **CurrentYr** to step 15
11	Take **PreviousYr** to the next iteration (the next *i*). Is `dorm` greater than zero?	*yes*—step 12 *no*—step 13
12	Put the data in **ghosts**, which contains "dormant" individuals. For *each* individual in **ghosts**, is the difference between the current year (**inv**[*i*]) and the year when that observation was measured	*yes*—put in **deadGhosts**—step 13

	greater than (`dorm` + 1)?	*no*—keep in **ghosts**—step 23
13	Put an “NA” in the “size_tplus1” col and “a “0” in the “survives_tplus1” col.	step 22
14	Compare the overlap between every genet in **PreviousYr** and every genet in **CurrentYr.** If `clonal=FALSE`, a parent and child pair with the most overlap get the same trackID. If `clonal=TRUE`, each child can have one parent, but each parent can have multiple children. If there is a child with multiple parents, then its parent is the polygon it overlaps with the most.	steps 16 and 17
15	Put a “1” in the “recruit” and “age” cols.	step 23
16	For every polygon in **PreviousYr**: Does it share a trackID with polygon(s) in **CurrentYr** (i.e., have a “child”)?	*yes*—put in **parents**—step 18 *no*—step 12
17	For every polygon in **CurrentYr**: Does it share a trackID with a polygon in **PreviousYr** (i.e., have a “parent”)?	*yes*—put in **children**—step 20 *no*—put in **orphans**—step 15
18	Put “1” in the “survives_tplus1” col., and size of the “child” genet in the “size_tplus1” col.	step 22
19	Put “NA” in the “size_tplus1” and “survives_tplus1” cols.	step 22
20	Put “0” in the “recruit” col and (age of parent + 1) in the “age” col.	step 23
21	Put “1” in the “recruit” and “age” cols., and “NA” in the “survives_tplus1” and “size_tplus1” cols.	step 22
22	Store the data in **output**, which the function will return. Is **inv**[*i*] the last year of sampling for this quadrat?	*yes*—step 1 *no*—step 23
23	Put either **ghosts** and **CurrentYr** (if all **CurrentYr** individuals are new recruits) *or* **ghosts**, **children,** and **orphans** (if there are “children” in this *i*) into **PreviousYr**	step 24
24	Go to the next *i*. Put **dat** data from year *i* into **CurrentYr**	step 6

Table 3. Accuracy of trackSpp() compared to tagged data

Test Dataset	**No. of individuals from trackSpp() / No. of individuals from tag dataset (% correct ID assignment)**	**No. of correct survivors / Total observations (% correct surv. assignments)**	**No. of correct recruits / Total observations (% correct recruit assignments)**
Tree basal areas	5,212/5,212 (100%)	77,059/77,062 (99.9%)	77,037/77,062 (99.7%)
O. coloradensis point locations	3,128/3,146 (99.4%)	5,225/5,245 (99.6%)	5,224/5,245 (99.5%)

Figure 1. Examples of digitized chart quadrat maps collected over six years measured at the Santa Rita Experimental Range near Tucson, AZ. Two grasses (*Bouteloua rothrockii* and *Heteropogon contortus*) were mapped as polygons, and two forbs (*Ambrosia artemisiifolia* and *Calliandra eriophylla*) were mapped as points.

Figure 2. Examples of different `buff` arguments. (**A**) Using `trackSpp()` with `buff` = 4 cm assigns the two green observations different trackIDs, and determines that the 1922 individual died, and a recruit sprouted in 1923. (**B**) Using `trackSpp()` with `buff` = 10 cm assigns these two observations shown in green the same trackID, and determines that the 1922 individual survived to 1923.

Figure 3. A potential “dormancy” scenario: The observation in 2000 (**A**) has no overlap with any observation in 2001 or 2002 (**B** and **C**). However, an observation in 2003 (**D**) overlaps in space with the 2000 observation. If `dorm`=2, the observations in 2000 and 2003 will be assigned the same trackID. If `dorm` = 1, the observations in 2000 and 2003 will be assigned different trackIDs.

Figure 4. (**A**) If `clonal=FALSE`, every polygon is given a unique trackID, represented by a unique color. (**B - C**) If `clonal=TRUE`, then `groupByGenet()` draws buffers around each polygon, shown here in lighter colors. If buffered polygons overlap, they get the same trackID. (**B**) With a 1 cm buffer (`buffGenet=0.01), groupByGenet()` identified 21 genets. (**C**) With a 5 cm buffer (`buffGenet=0.05`), `groupByGenet()` identified 11 genets.

Figure 5. Panels **A-D** show trackIDs assigned to individuals over four years using `trackSpp()` with `dorm`=1, `clonal=TRUE`, `buff=0.05` and `buffGenet=0.01`. Labels and color indicate

trackID assignment. Panels **E-F** show the same data over the same time period, but with trackIDs assigned using `dorm=1`, `clonal=FALSE`, and `buff=0.0`.

Figure 6. The two methods `getNeighbors()` can use to calculate neighborhood density. The focal individual is outlined in pink, and the neighborhood buffer around it is shown in dashed pale pink. Individuals inside the buffer are outlined in dark grey. These examples show interspecific neighborhood occupancy (`type=allSpp`). (**A)** The "count" method counts each individual inside of the buffer zone and returns the total number, which here is five. (**B)** The "area" method returns two values: the area of the buffer zone around the focal individual, and the area of the buffer zone that is occupied by neighbors (summed area of the shaded grey portions of polygons).