The forestecology R package for fitting and assessing neighborhood models of the effect of interspecific competition on the growth of trees

Albert Y. Kim *
Program in Statistical & Data Sciences, Smith College
and
David N. Allen
Biology Department, Middlebury College
and
Simon P. Couch
Mathematics Department, Reed College

March 11, 2021

Abstract

1. Neighborhood competition models are powerful tools to measure the effect of interspecific competition. Statistical methods to ease the application of these models are currently lacking.
2. We present the `forestecology` package providing methods to i) specify neighborhood competition models, ii) evaluate the effect of competitor species identity using permutation tests, and iii) measure model performance using spatial cross-validation. Following Allen & Kim (2020), we implement a Bayesian linear regression neighborhood competition model.
3. We demonstrate the package's functionality using data from the Smithsonian Conservation Biology Institute's large forest dynamics plot, part of the ForestGEO global network of research sites. Given ForestGEO's data collection protocols and data formatting standards, the package was designed with cross-site compatibility in mind. We highlight the importance of spatial cross-validation when interpreting model results.
4. The package features i) `tidyverse`-like structure whereby verb-named functions

*Assistant Professor, Statistical & Data Sciences, Smith College, Northampton, MA 01063 (e-mail: `akim04@smith.edu`).

can be modularly "piped" in sequence, ii) functions with standardized inputs/outputs of simple features `sf` package class, and iii) an S3 object-oriented implementation of the Bayesian linear regression model. These three facts allow for clear articulation of all the steps in the sequence of analysis and easy wrangling and visualization of the geospatial data. Furthermore, while the package only has Bayesian linear regression implemented, the package was designed with extensibility to other methods in mind.

Keywords: forest ecology, interspecific competition, neighborhood competition, tree growth, R, ForestGEO, spatial cross-validation

1 Introduction

Repeat-censused forest plots offer excellent opportunities to test neighborhood models of the effect of competition on the growth of trees (Canham et al. 2004). Neighborhood models of competition have been used to: test whether the species identity of a competitor matters [Uriarte et al. (2004); measure species-specific competition coefficients (Das 2012, Tatsumi et al. (2016)); test competing models to see what structures competitive interactions, e.g. traits or phylogeny (Allen & Kim 2020, Uriarte et al. 2010); and inform selective logging practices (Canham et al. 2006). Although these are well-described methods, few methods are currently available for easy application.

We address this shortcoming with the `forestecology` R package providing methods and data for forest ecology model fitting and assessment, available on CRAN (`https://cran.r-project.org/package=forestecology`) and on GitHub (`https://github.com/rudeboybert/forestecology`). The package is written to model stem diameter growth between two censuses based on neighborhood competition, largely following the methods in Allen & Kim (2020).

Let $i = 1, \ldots, n_j$ index all n_j trees of "focal" species j; let $j = 1, \ldots, J$ index all J focal species; and let $k = 1, \ldots, K$ index all K "competitor" species. The average annual growth in diameter at breast height (DBH) y_{ij} (in centimeters/year) of the i^{th} tree of focal species j is modeled as

$$y_{ij} = \beta_{0,j} + \beta_{\text{dbh},j} \cdot \text{dbh}_{ij} + \sum_{k=1}^{K} \lambda_{jk} \cdot \text{BA}_{ijk} + \epsilon_{ij} \tag{1}$$

where $\beta_{0,j}$ is the diameter-independent growth rate of species j; dbh_{ij} is the DBH of the focal tree at the earlier census and $\beta_{\text{dbh},j}$ the slope of that species's diameter-growth relationship; BA_{ijk} is the sum of the basal area of all trees of competitor species k and λ_{jk} quantifies the corresponding change in growth for individuals of group j from these competitors; and ϵ_{ij} is a random error term distributed $\text{Normal}(0, \sigma^2)$. Allen & Kim (2020) estimate all parameters via Bayesian linear regression, while exploiting Normal/Inverse Gamma conjugacy to derive closed-form solutions to all posterior distributions[1]. These closed-form solutions are not as computationally expensive as approximations from Markov Chain Monte Carlo algorithms.

To evaluate whether competitor species identity matters, Allen & Kim (2020) run a permutation test where a null hypothesis of no species grouping-specific effects of competition is assumed, thus the species identity of all competitors can be permuted:

$$\begin{aligned} & H_0 : \lambda_{jk} = \lambda_j \text{ for all } k = 1, \ldots, K \\ \text{vs.} \quad & H_A : \text{at least one } \lambda_{jk} \text{ is different} \end{aligned} \tag{2}$$

Furthermore, to account for the spatial autocorrelation in their estimates of out-of-sample model error, Allen & Kim (2020) use spatial cross-validation. Estimates of model error that do not account for this dependence tend to underestimate the true model error (Roberts et al. 2017).

The package is designed with "tidy" design principles in mind (Wickham et al. 2019). Much like all `tidyverse` packages, `forestecology` has verb-named functions that can be modularly composed using the pipe `%>%` operator to sequentially complete all necessary analysis steps (Bache & Wickham 2020). Furthermore, the inputs and outputs of most functions use the same "simple features for R" data structures from the `sf` package, a package for standardized and `tidyverse`-friendly wrangling and visualizing of spatial data (Pebesma 2018).

Currently the package only implements the Bayesian linear regression model detailed in Equation 1. As we demonstrate in Section 2.4 however, the fitting of this model is self-contained in a single function `comp_bayes_lm()` which returns an object of S3 class type

[1] See S1 Appendix of Allen & Kim (2020), available at `https://doi.org/10.1371/journal.pone.0229930.s004`

`comp_bayes_lm`. This class has generic methods implemented to print, make predictions, and plot all results. Therefore the package can be modularly extended to fit other models as long as they are coded similarly to `comp_bayes_lm()` and have equivalent generic methods implemented.

2 forestecology workflow: a case study

We present a case-study of `forestecology`'s functionality on data from the Smithsonian Conservation Biology Institute (SCBI) large forest dynamics plot in Front Royal, VA, USA, part of the ForestGEO global network of research sites (Bourg et al. 2013, Anderson-Teixeira et al. (2015)). The 25.6 ha (640 x 400 m) plot is located at the intersection of three of the major physiographic provinces of the eastern US—the Blue Ridge, Ridge and Valley, and Piedmont provinces—and is adjacent to the northern end of Shenandoah National Park.

The package has the following goals: to evaluate i) the effect of competitor species identity using permutation tests and ii) model performance using spatial cross-validation. We outline the four-step basic analysis sequence:

1. Compute the growth of stems based on two censuses.
2. Add spatial information:
    1. Define a buffer region of trees.
    2. Add spatial cross-validation block information.
3. Identify all focal trees and their competitors.
4. Apply model, which includes:
    1. Fit model.
    2. Compute predicted values.
    3. Visualize posterior distributions.

We start by loading all packages.

```
library(tidyverse)
library(lubridate)
library(sf)
library(patchwork)
library(forestecology)
library(blockCV)

# Resolve conflicting functions
filter <- dplyr::filter
select <- dplyr::select
```

2.1 Step 1: Compute the growth of trees based on census data

We first compute the growth of trees using data from two censuses. `compute_growth()` computes the average annual growth based on census data that roughly follows ForestGEO standards. Despite such standards, minor variations will still exist between sites, thereby necessitating some data wrangling. For example, the SCBI site records all DBH values in millimeters (Bourg et al. 2013), whereas the Michigan Big Woods site used in Allen & Kim (2020) records them in centimeters (Allen et al. 2020).

We load both 2008 and 2014 SCBI census `.csv` files as they existed on GitHub on 2020/11/20 and perform minor data wrangling (Gonzalez-Akre et al. 2020). We then only consider a 9 ha subsection of the 25.6 ha of the site to speed up computation for this example: `gx` from 0–300 instead of 0–400 and `gy` from 300–600 instead of 0–640.

```
census_2013_scbi <- read_csv("scbi.stem2.csv") %>%
  select(stemID, sp, date = ExactDate, gx, gy, dbh, codes, status) %>%
  mutate(
    # Convert date from character to date
    date = mdy(date),
    # Convert dbh to be in cm
    dbh = as.numeric(dbh)/10
  ) %>%
  filter(gx < 300, between(gy, 300, 600))

census_2018_scbi <- read_csv("scbi.stem3.csv") %>%
  select(stemID, sp, date = ExactDate, gx, gy, dbh, codes, status) %>%
  mutate(
    date = mdy(date),
    dbh = as.numeric(dbh)/10
  ) %>%
  filter(gx < 300, between(gy, 300, 600))
```

These two data frames are then used as inputs to `compute_growth()`, along with `id` specifying the variable that uniquely identifies each tree-stem. We also discard all resprouts with `code == R` in the later census, since we are only interested in the growth of surviving, and not resprouted, stems.

```
growth_scbi <-
  compute_growth(
    census_1 = census_2013_scbi,
    census_2 = census_2018_scbi %>% filter(!str_detect(codes, "R")),
    id = "stemID"
```

```
  )
growth_scbi %>%
  select(stemID, sp, dbh1, dbh2, growth, geometry)
## Simple feature collection with 7954 features and 5 fields
## geometry type:  POINT
## dimension:      XY
## bbox:           xmin: 0.2 ymin: 300 xmax: 300 ymax: 600
## CRS:            NA
## # A tibble: 7,954 x 6
##   stemID sp     dbh1  dbh2 growth   geometry
##    <dbl> <fct> <dbl> <dbl>  <dbl>    <POINT>
## 1      4 nysy  13.6   14.2  0.103 (14.2 428)
## 2      5 havi   8.8    9.6  0.150  (9.4 436)
## 3      6 havi   3.25  4     0.140  (1.3 434)
## 4     77 qual  65.2   66    0.141 (34.7 307)
## 5     79 tiam  47.7   46.8 -0.161   (40 381)
## # ... with 7,949 more rows
```

The output `growth_scbi` is a data frame of class `sf` that includes among other variables the species variable `sp` converted to a factor, the average annual `growth` in DBH (cm · y^{-1}) for all stems that were alive at both time points, and the `sf` package's encoding of geolocations of `geometry` type <POINT>. Given that `growth_scbi` is of class `sf`, it can be easily plotted in `ggplot2` using `geom_sf()` as seen in Figure 1.

```
ggplot() +
  geom_sf(data = growth_scbi %>% sample_n(500), aes(size = growth)) +
  scale_size_binned(limits = c(0.1, 1)) +
  labs(size = expression(paste(Growth, " (cm ",y^{-1},')')) )
```

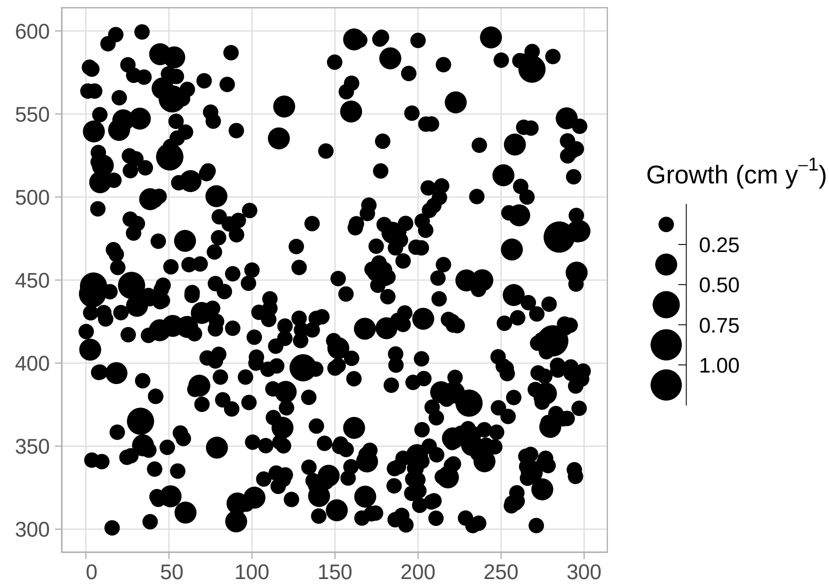

Figure 1: Compute growth of trees based on census data. A map of the growth of a random sample of 500 trees from a 9 ha subsection of the Smithsonian Conservation Biology Institute (SCBI) forest plot.

2.2 Step 2: Add spatial information

We then add spatial information to `growth_scbi`. We first add a "buffer region" to the periphery of the study region. Since some of our model's explanatory variables are cumulative, we must ensure that all trees being modeled are not biased to have different neighbor structures. This is of concern for trees at the boundary of the study region who will not have all their neighbors included in the census stems. To account for such edge effects, only trees that are not part of this buffer region, i.e. are part of the interior of the study region, will have their growth modeled (Waller & Gotway 2004).

Our model of interspecific competition relies on a spatial definition of who competitor trees are: all trees within a distance `comp_dist` of a focal tree. We set `comp_dist` to 7.5m, a value informed by other studies (Canham et al. 2004, Uriarte et al. (2004), Canham et al. (2006)). We use `comp_dist` and a manually constructed `sf` representation of the study region's boundary as inputs to `add_buffer_variable()` to add a `buffer` boolean variable to `growth_scbi`. All trees with `buffer` equal to `FALSE` will be our focal trees whose growth will be modeled, whereas those with `TRUE` will only act as competitor trees.

```
# Define competitive distance range
comp_dist <- 7.5


# Manually construct study region boundary
study_region_scbi <- tibble(
  x = c(0, 300, 300, 0, 0),
  y = c(300, 300, 600, 600, 300)
) %>%
  sf_polygon()


growth_scbi <- growth_scbi %>%
  add_buffer_variable(size = comp_dist, region = study_region_scbi)
```

The second element of spatial information we add are blocks corresponding to folds of a spatial cross-validation algorithm. Conventional cross-validation algorithms assign individual observations to folds by randomly resampling them all while assuming they are statistically independent. In the case of forest census data however, observations exhibit spatial autocorrelation. We therefore incorporate this dependence into the cross-validation algorithm by resampling spatial blocks of trees (Roberts et al. 2017, Pohjankukka et al. (2017)).

We first manually define an `sf` object defining four folds that partition the study region. We then use the output of the `spatialBlock()` function from the `blockCV` package to associate each tree in `growth_scbi` to the correct `foldID` (Valavi et al. 2019).[2] This `foldID`

[2]In the Supporting Information we present an example where the folds themselves are created automatically, as opposed to manually as in the example.

variable will be used in Section 2.6.

Figure 2 illustrates the net effect of adding these two elements of spatial information to `growth_scbi`.

```
# Manually define spatial blocks to act as folds
n_fold <- 4
fold1 <- cbind(c(0, 150, 150, 0), c(300, 300, 450, 450))
fold2 <- cbind(c(150, 300, 300, 150), c(300, 300, 450, 450))
fold3 <- cbind(c(0, 150, 150, 0), c(450, 450, 600, 600))
fold4 <- cbind(c(150, 300, 300, 150), c(450, 450, 600, 600))

blocks_scbi <- bind_rows(
  sf_polygon(fold1), sf_polygon(fold2), sf_polygon(fold3),
  sf_polygon(fold4)
) %>%
  mutate(folds = c(1:n_fold) %>% factor())

# Associate each observation to a fold
spatial_block_scbi <-
  spatialBlock(speciesData = growth_scbi, k = n_fold,
               selection = "systematic", blocks = blocks_scbi,
               showBlocks = FALSE, verbose = FALSE)

growth_scbi <- growth_scbi %>%
  mutate(foldID = spatial_block_scbi$foldID %>% factor())
```

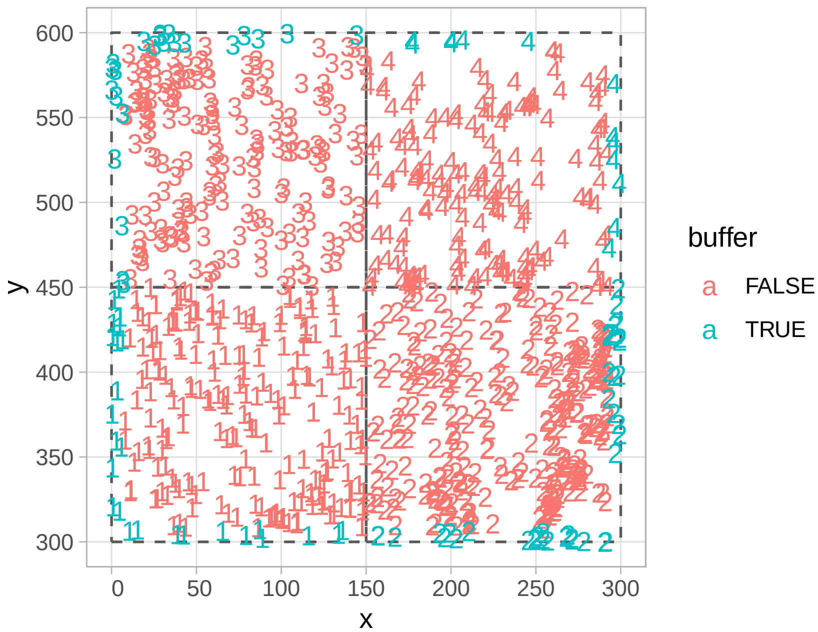

Figure 2: Add spatial information. A buffer region and spatial cross-validation blocks 1 througth 4. The location of each tree is marked with its fold number where the folds are delineated with solid lines. The color of each digit indicates whether the tree is part of the buffer region (thus will only be considered as a competitor tree) or is part of the interior of the study region (thus is a focal tree whose growth is of modeled interest).

```
ggplot() +
  geom_sf(data = blocks_scbi, fill = "transparent", linetype = "dashed") +
  geom_sf_text(data = growth_scbi %>% sample_n(1000),
               aes(label = foldID, col = buffer))
```

2.3 Step 3: Identify all focal and corresponding competitor trees

We then identify all focal trees and their corresponding competitor trees. More specifically, identify all trees that are not part of the buffer region, have a valid `growth` measurement, and have at least one neighbor within 7.5m. We do this using `create_focal_vs_comp()`, which takes the previously detailed `comp_dist` and `id` arguments as well as the `sf` representation of the spatial cross-validation blocks and returns a new data frame `focal_vs_comp_scbi`.

```
focal_vs_comp_scbi <- growth_scbi %>%
  create_focal_vs_comp(comp_dist, blocks = blocks_scbi, id = "stemID")
focal_vs_comp_scbi %>%
  select(focal_ID, focal_sp, geometry, growth, comp)
## # A tibble: 6,296 x 5
##   focal_ID focal_sp   geometry growth comp
##      <dbl> <fct>       <POINT>  <dbl> <list>
## 1        4 nysy     (14.2 428)  0.103 <tibble [20 x 4]>
## 2        5 havi      (9.4 436)  0.150 <tibble [32 x 4]>
## 3       79 tiam       (40 381) -0.161 <tibble [20 x 4]>
## 4       80 caca     (38.7 422)  0.253 <tibble [12 x 4]>
## 5       96 libe       (60 310)  0.262 <tibble [14 x 4]>
## # ... with 6,291 more rows
```

The resulting `focal_vs_comp_scbi` has 6296 rows, representing the subset of the 7954 trees in `growth_scbi` that will be considered as focal trees. The variables `focal_ID` and `focal_sp` relate to tree-stem identification and species information. Most notably however is the variable `comp`, which contains information on all competitor trees saved in `tidyr` package list-column format (Wickham 2020). To inspect this information, we flatten the `comp` list-column for the tree with `focal_ID` 4 in the first row, here a `tibble [20 × 4]`, into regular columns using `unnest()` from the `tidyr` package.

```
focal_vs_comp_scbi %>%
  filter(focal_ID == 4) %>%
  select(focal_ID, dbh, comp) %>%
  unnest(cols = "comp")
## # A tibble: 20 x 6
##   focal_ID   dbh comp_ID  dist comp_sp comp_basal_area
```

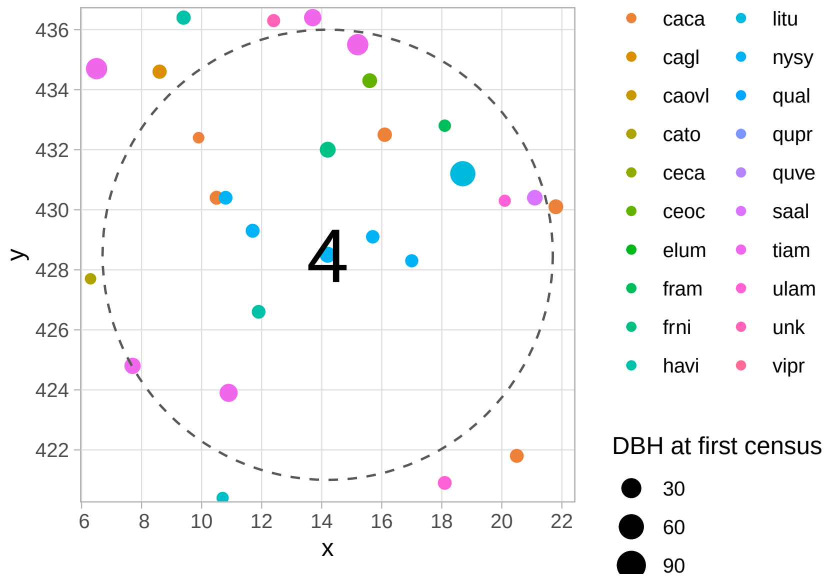

Figure 3: Identify all focal and corresponding competitor trees. The dashed circle extends 7.5m away from the focal tree 4 while all 20 competitor trees are within this circle.

```
##       <dbl> <dbl>    <dbl> <dbl> <fct>             <dbl>
## 1         4  13.6     1836  7.48 tiam             0.0176
## 2         4  13.6     1847  2.81 nysy             0.00332
## 3         4  13.6     1848  1.62 nysy             0.00396
## 4         4  13.6     1849  2.62 nysy             0.00535
## 5         4  13.6     1850  2.98 havi             0.00472
## # ... with 15 more rows
```

We observe 4 variables describing 20 competitor trees: the unique tree-stem ID, the distance to the focal tree (all ≤ 7.5 m), the species, and the basal area (in m^2) calculated as $\frac{\pi \times (\text{DBH}/2)^2}{10000}$ for the DBH in cm from the earlier census. Saving competitor information in list-column format minimizes redundancy since we do not need to repeat information on the focal tree 20 times. We visualize the spatial distribution of these trees in Figure 3.

2.4 Step 4: Fit model

Lastly, we fit the competition Bayesian linear regression model for tree growth outlined in Equation 1 using `comp_bayes_lm()`. This function has an option to specify prior distributions of all parameters, chosen here to be the defaults detailed in `?comp_bayes_lm`.

```
comp_bayes_lm_scbi <- focal_vs_comp_scbi %>%
  comp_bayes_lm(prior_param = NULL)
```

The resulting`comp_bayes_lm_scbi` is an object of S3 class type `comp_bayes_lm` containing the posterior values of all parameters. Furthermore, this class includes generics for three methods. First, the generic for `print()` displays the names of all prior and posterior parameters and the model formula:

```
comp_bayes_lm_scbi
## Bayesian linear regression model parameters with a multivariate Normal
## likelihood. See ?comp_bayes_lm for details:
##
##   parameter_type           prior posterior
## 1 Inverse-Gamma on sigma^2 a_0   a_star
## 2 Inverse-Gamma on sigma^2 b_0   b_star
## 3 Multivariate t on beta   mu_0  mu_star
## 4 Multivariate t on beta   V_0   V_star
##
## Model formula:
## growth ~ sp + dbh + dbh * sp + acne * sp + acru * sp + amar * sp + astr
## * sp + caca * sp + caco * sp + cade * sp + cagl * sp + caovl * sp + cato
## * sp + ceca * sp + ceoc * sp + chvi * sp + cofl * sp + crpr * sp + crsp
## * sp + divi * sp + elum * sp + fagr * sp + fram * sp + frni * sp + frpe
```

```
## * sp + havi * sp + ilve * sp + juci * sp + juni * sp + libe * sp + litu
## * sp + nysy * sp + pist * sp + pivi * sp + ploc * sp + prav * sp + prse
## * sp + qual * sp + quco * sp + qufa * sp + qumi * sp + qupr * sp + quru
## * sp + quve * sp + rops * sp + saal * sp + saca * sp + tiam * sp + ulam
## * sp + ulru * sp + unk * sp + vipr * sp
```

Next, the generic for `predict()` takes the posterior parameter values in `comp_bayes_lm_scbi` and a `newdata` data frame, and outputs a vector `growth_hat` of predicted DBH values $\widehat{y_{ij}}$ computed from the posterior predictive distribution.

```
focal_vs_comp_scbi <- focal_vs_comp_scbi %>%
  mutate(growth_hat = predict(comp_bayes_lm_scbi, newdata = focal_vs_comp_scbi))
```

```
focal_vs_comp_scbi %>%
  select(focal_ID, focal_sp, dbh, growth, growth_hat)
## # A tibble: 6,296 x 5
##   focal_ID focal_sp   dbh growth growth_hat
##      <dbl> <fct>    <dbl>  <dbl>      <dbl>
## 1        4 nysy     13.6   0.103     0.0809
## 2        5 havi      8.8   0.150     0.112
## 3       79 tiam     47.7  -0.161     0.229
## 4       80 caca      5.15  0.253     0.121
## 5       96 libe      2.3   0.262     0.142
## # ... with 6,291 more rows
```

We can now compare the observed and predicted growths to compute the root mean squared error (RMSE) of our model:

```
model_rmse <- focal_vs_comp_scbi %>%
  rmse(truth = growth, estimate = growth_hat) %>%
  pull(.estimate)
model_rmse
## [1] 0.128
```

Lastly, the generic for `ggplot2::autoplot()` allows us to visualize all posterior distributions, as seen in Figure 4. Setting `type` to `"intercepts"` and `"dbh_slopes"` returns species-specific posterior distributions for $\beta_{0,j}$ and $\beta_{dbh,j}$ respectively, while setting `type = "competition"` returns competition coefficients $\lambda_{j,k}$.

```
# Plot posteriors for only a subset of species
sp_to_plot <- c("litu", "quru", "cagl")

plot1 <- autoplot(comp_bayes_lm_scbi, type = "intercepts",
                  sp_to_plot = sp_to_plot)
plot2 <- autoplot(comp_bayes_lm_scbi, type = "dbh_slopes",
                  sp_to_plot = sp_to_plot)
plot3 <- autoplot(comp_bayes_lm_scbi, type = "competition",
                  sp_to_plot = sp_to_plot)

# Combine plots using the patchwork package
(plot1 | plot2) / plot3
```

For many users the visualizations of $\lambda_{j,k}$ will be of particular interest as they provide insight into species-specific competitive interactions, where negative values indicate a competitor species which slows the growth of a focal species. Here, for example, we see that tulip poplars (litu) have a strong negative effect on the growth of conspecifics but relatively

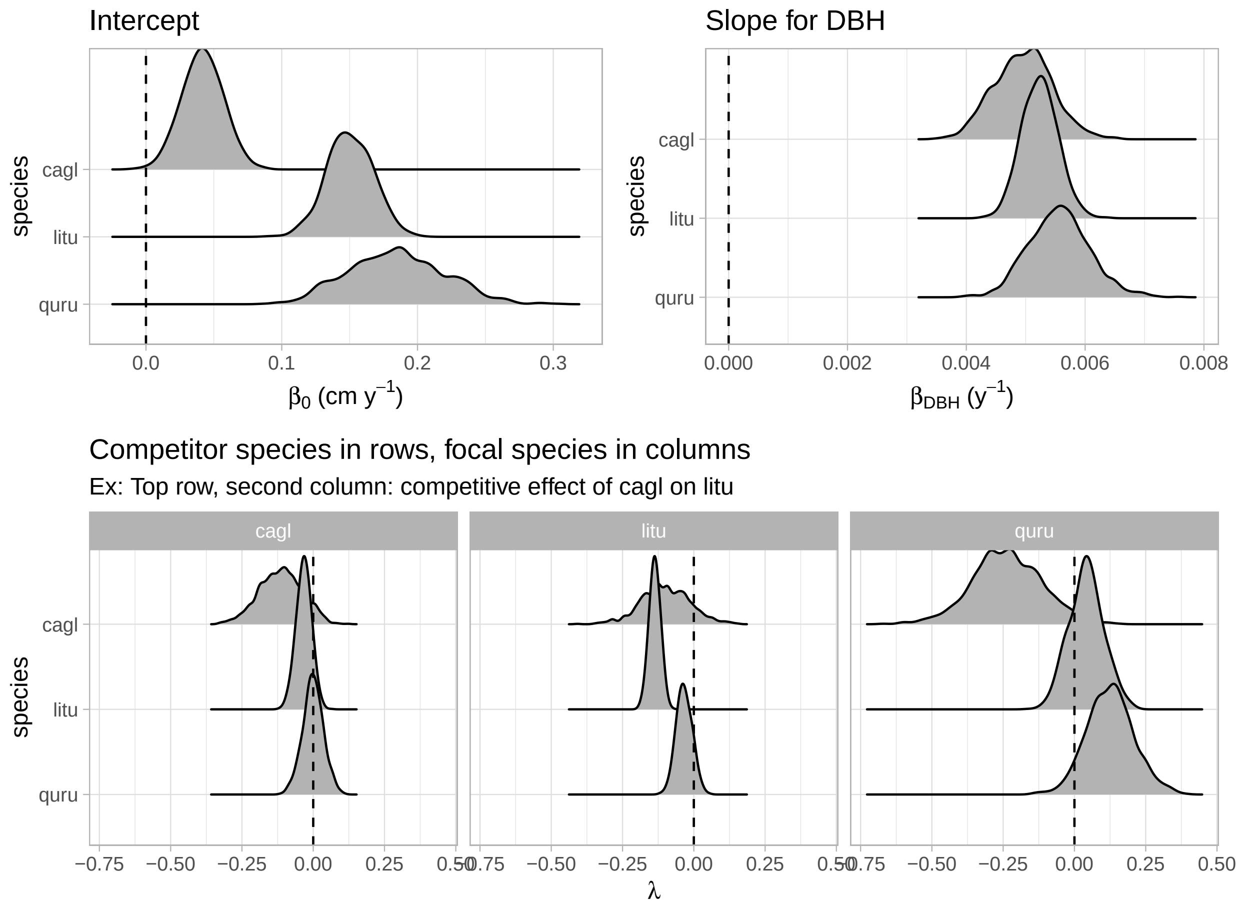

Figure 4: Fit model. Posterior distributions of all parameters. For compactness we include only three species.

lesser effect on pignut hickory (cagl) and red oak (quru) neighbors.

Currently the `forestecology` package can only fit the competition Bayesian linear regression model in Equation 1. However, it can be extended to any model as long as it is implemented in a function similar to `comp_bayes_lm()`.

2.5 Evaluate the effect of competitor species identity using permutation tests

To evaluate the effect of competitor species identity, we use the above four steps along with the permutation test in Equation 2. Under a null hypothesis where competitor species identity does not matter, we can permute the competitor species identities within each focal tree, compute the RMSE test statistic, repeat this process several times to construct a null distribution, and compare it to the observed RMSE to assess significance. Going back to our example in Section 2.3 of focal tree with `focal_ID` 4 and its 20 competitors, the permutation test only randomly resamples the `comp_sp` variable without replacement, leaving all other variables intact. This resampling is nested within each focal tree in order to preserve neighborhood structure. We perform this permutation test once again using `comp_bayes_lm()` but by setting `run_shuffle = TRUE`.

```
comp_bayes_lm_scbi_shuffle <- focal_vs_comp_scbi %>%
  comp_bayes_lm(prior_param = NULL, run_shuffle = TRUE)

focal_vs_comp_scbi <- focal_vs_comp_scbi %>%
  mutate(growth_hat_shuffle = predict(comp_bayes_lm_scbi_shuffle,
                                      newdata = focal_vs_comp_scbi))
```

```
model_rmse_shuffle <- focal_vs_comp_scbi %>%
  rmse(truth = growth, estimate = growth_hat_shuffle) %>%
  pull(.estimate)
model_rmse_shuffle
## [1] 0.131
```

The resulting permutation test RMSE of 0.131 is larger than the earlier RMSE of 0.128, suggesting that models that do incorporate competitor species identity better fit the data.

2.6 Evaluate model performance using spatial cross-validation

To evaluate model performance, we use spatial cross-validation. The model fit in Section 2.4 uses the same data to both fit and assess model performance. Given the spatial-autocorrelation of our data, this can potentially lead to overfit models (Roberts et al. 2017). To mitigate this risk, we use the spatial cross-validation blocking scheme encoded in the `foldID` variable from Section 2.2 and visualized in Figure 2.

At each iteration of the cross-validation, one fold acts as the test set and the remaining three act as the training set. We fit the model to all focal trees in the training set, apply the model to all focal trees in the test set, compute predicted values, and compute the RMSE. Furthermore, to maintain spatial independence between the test and training sets, a "fold buffer" that extends 7.5m outwards from the boundary of the test set is considered; all trees within this "fold buffer" are excluded from the training set (see Figure 5).

This process is repeated for each of the four folds acting as the test set, then the four RMSE's are averaged to provide a single estimate of model error. This algorithm is implemented in `run_cv()`, which acts as a wrapper function to both `comp_bayes_lm()` that fits the model and `predict()` that returns predicted values.

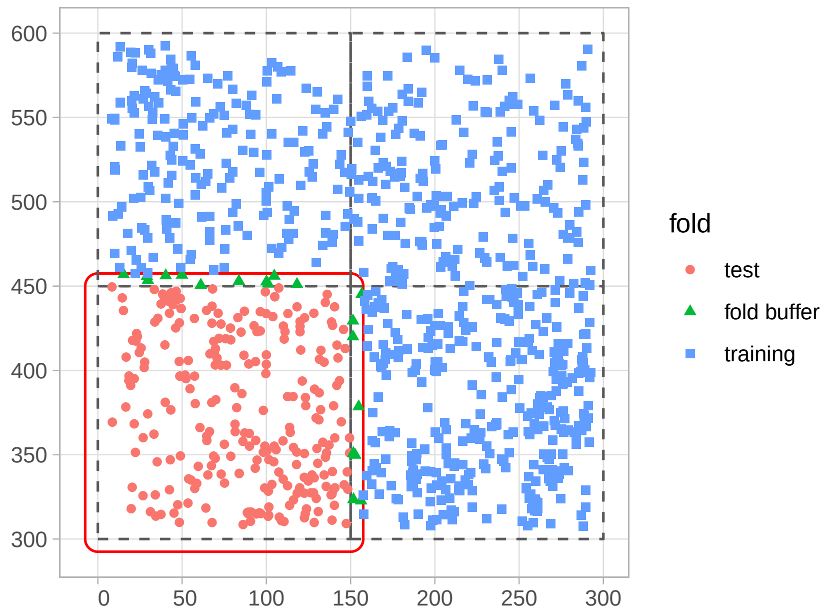

Figure 5: Schematic of spatial cross-validation. Using the k = 1 fold (bottom-left) as the test set, k = 2 through 4 as the training set, along with a "fold buffer" extending outwards from the test set to maintain spatial independence between it and the training set.

```
focal_vs_comp_scbi <- focal_vs_comp_scbi %>%
  run_cv(comp_dist = comp_dist, blocks = blocks_scbi)
```

```
model_rmse_cv <- focal_vs_comp_scbi %>%
  rmse(truth = growth, estimate = growth_hat) %>%
  pull(.estimate)
model_rmse_cv
## [1] 0.14
```

The resulting RMSE of 0.14 computed using cross-validation is larger than the earlier RMSE of 0.128, suggesting that models that do not account for spatial autocorrelation generate model error estimates that are overly optimistic, i.e. RMSE values that are too low.

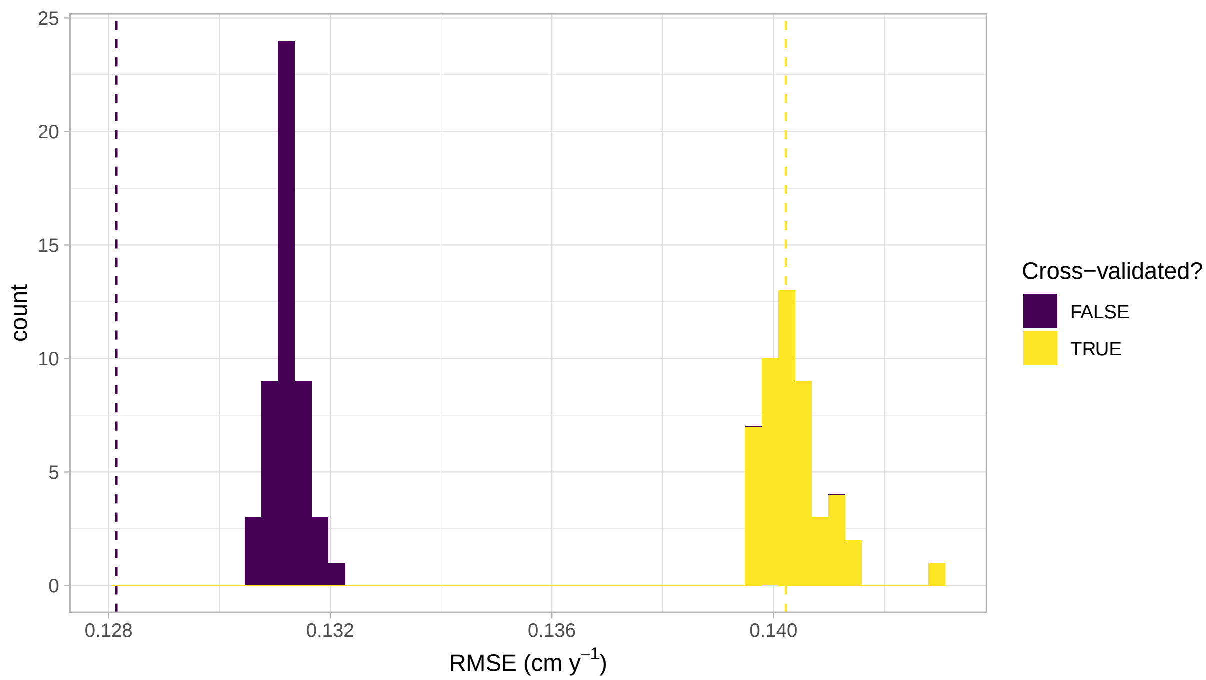

Figure 6: Comparison of root mean squared error of models for standard, permuted, and spatial cross-validated error estimates. The dotted lines show non-permuted competitor identity, while the histogras so the RMSE for 49 permutations. The colors indicate whether cross validaton was used.

3 Importance of spatial cross-validation

`run_cv()` also accepts the `run_shuffle` argument in order to permute competitor species identity as described in Section 2.5. Figure 6 compares model performance for 49 permutations of competitor species and RMSE calculations, both with and without cross-validation. Without cross-validation, competitor species identity does matter as the observed RMSE was significantly lower than the permutation null distribution of RMSE. However, once we incorporate spatial cross-validation, this improvement disappears. These results suggest that in this 9 ha subplot of the SCBI plot, competitive interactions do not depend on the identity of the competitor, which is the opposite of what has been observed in other locations (Allen & Kim 2020, Uriarte et al. (2004)). This provides a striking example of the importance of cross-validation, as without it the over-fit model gives rise to an incorrect conclusion.

4 Conclusion

The `forestecology` package provides an accessible way to fit and test models of neighborhood competition. While the package is designed with ForestGEO plot data in mind, we envision that it can be modified to work on any single large, mapped forest plot in which at least two measurements of each individual have been taken. Furthermore, we also envision the package being used more generally for inventory plot-structure with many spatially separated plots, e.g. the US Forest Service Forest Inventory and Analysis plots (Smith 2002). In future versions of `forestecology` we also hope to include models that account for tree mortality in addition to tree growth. The package follows the `tidy` data design principles, leverages the `sf` package for spatial data, and S3 open-oriented model implementation structure. We hope that the package will increase the use of neighborhood competition models to better understand what structures plant competition.

5 Acknowledgments

We thank Sophie Li for their feedback on the package interface. The authors declare no conflicts of interest.

6 Author's contributions

AYK and DNA conceived the ideas and coded a draft of the package. AYK wrote an initial manuscript draft. SPC rewrote much of the package's code to align with R and "tidy" best practices (Wickham et al. 2019). All authors contributed to subsequent drafts and gave final approval for manuscript.

7 Data accessibility

We intend to archive all data and source code for this manuscript on GitHub at `https://github.com/rudeboybert/forestecology` and on Zenodo upon acceptance. The example Smithsonian Conservation Biology Institute census data are available on GitHub at `https://github.com/SCBI-ForestGEO/SCBI-ForestGEO-Data/tree/master/tree_main_census/data/census-csv-files` and are archived on Zenodo at `https://doi.org/10.5281/zenodo.2649301` (Gonzalez-Akre et al. 2020).

References

Allen, D., Dick, C., Burnham, R. J., Perfecto, I. & Vandermeer, J. (2020), 'The Michigan Big Woods research plot at the Edwin S. George, Pinckney, MI, USA', *Miscellaneous Publications of the Museum of Zoology, University of Michigan* **207**.
URL: *http://hdl.handle.net/2027.42/156251*

Allen, D. & Kim, A. Y. (2020), 'A permutation test and spatial cross-validation approach to assess models of interspecific competition between trees', *PLOS ONE* **15**(3), e0229930. Publisher: Public Library of Science.
URL: *https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0229930*

Anderson-Teixeira, K. J., Davies, S. J., Bennett, A. C., Gonzalez-Akre, E. B., Muller-Landau, H. C., Wright, S. J., Salim, K. A., Zambrano, A. M. A., Alonso, A., Baltzer, J. L., Basset, Y., Bourg, N. A., Broadbent, E. N., Brockelman, W. Y., Bunyavejchewin, S., Burslem, D. F. R. P., Butt, N., Cao, M., Cardenas, D., Chuyong, G. B., Clay, K., Cordell, S., Dattaraja, H. S., Deng, X., Detto, M., Du, X., Duque, A., Erikson, D. L., Ewango, C. E. N., Fischer, G. A., Fletcher, C., Foster, R. B., Giardina, C. P., Gilbert, G. S., Gunatilleke, N., Gunatilleke, S., Hao, Z., Hargrove, W. W., Hart, T. B., Hau, B. C. H., He, F., Hoffman, F. M., Howe, R. W., Hubbell, S. P., Inman-Narahari, F. M.,

Jansen, P. A., Jiang, M., Johnson, D. J., Kanzaki, M., Kassim, A. R., Kenfack, D., Kibet, S., Kinnaird, M. F., Korte, L., Kral, K., Kumar, J., Larson, A. J., Li, Y., Li, X., Liu, S., Lum, S. K. Y., Lutz, J. A., Ma, K., Maddalena, D. M., Makana, J.-R., Malhi, Y., Marthews, T., Serudin, R. M., McMahon, S. M., McShea, W. J., Memiaghe, H. R., Mi, X., Mizuno, T., Morecroft, M., Myers, J. A., Novotny, V., Oliveira, A. A. d., Ong, P. S., Orwig, D. A., Ostertag, R., Ouden, J. d., Parker, G. G., Phillips, R. P., Sack, L., Sainge, M. N., Sang, W., Sri-ngernyuang, K., Sukumar, R., Sun, I.-F., Sungpalee, W., Suresh, H. S., Tan, S., Thomas, S. C., Thomas, D. W., Thompson, J., Turner, B. L., Uriarte, M., Valencia, R., Vallejo, M. I., Vicentini, A., Vrška, T., Wang, X., Wang, X., Weiblen, G., Wolf, A., Xu, H., Yap, S. & Zimmerman, J. (2015), 'CTFS-ForestGEO: a worldwide network monitoring forests in an era of global change', *Global Change Biology* **21**(2), 528–549.
URL: *http://onlinelibrary.wiley.com/doi/abs/10.1111/gcb.12712*

Bache, S. M. & Wickham, H. (2020), *magrittr: A Forward-Pipe Operator for R*. R package version 2.0.1.
URL: *https://CRAN.R-project.org/package=magrittr*

Bourg, N. A., McShea, W. J., Thompson, J. R., McGarvey, J. C. & Shen, X. (2013), 'Initial census, woody seedling, seed rain, and stand structure data for the SCBI SIGEO Large Forest Dynamics Plot', *Ecology* **94**(9), 2111–2112.
URL: *http://esajournals.onlinelibrary.wiley.com/doi/abs/10.1890/13-0010.1*

Canham, C. D., LePage, P. T. & Coates, K. D. (2004), 'A neighborhood analysis of canopy tree competition: effects of shading versus crowding', *Canadian Journal of Forest Research* **34**(4), 778–787. Publisher: NRC Research Press Ottawa, Canada.
URL: *https://cdnsciencepub.com/doi/abs/10.1139/x03-232*

Canham, C. D., Papaik, M. J., Uriarte, M., McWilliams, W. H., Jenkins, J. C.

& Twery, M. J. (2006), 'Neighborhood Analyses Of Canopy Tree Competition Along Environmental Gradients In New England Forests', *Ecological Applications* **16**(2), 540–554. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1890/1051-0761%282006%29016%5B0540%3ANAOCTC%5D2.0.CO%3B2.

Das, A. (2012), 'The effect of size and competition on tree growth rate in old-growth coniferous forests', *Canadian Journal of Forest Research* **42**, 1983–1995.

Gonzalez-Akre, E., McGregor, I., Anderson-Teixeira, K., Dow, C., Herrmann, V., Terrell, A., Kim, A. Y., Vinod, N. & Helcoski, R. (2020), 'SCBI-ForestGEO/SCBI-ForestGEO-Data: 2020 update'.
URL: *https://doi.org/10.5281/zenodo.4041595*

Pebesma, E. (2018), 'Simple Features for R: Standardized Support for Spatial Vector Data', *The R Journal* **10**(1), 439–446.
URL: *https://journal.r-project.org/archive/2018/RJ-2018-009/index.html*

Pohjankukka, J., Pahikkala, T., Nevalainen, P. & Heikkonen, J. (2017), 'Estimating the prediction performance of spatial models via spatial k-fold cross validation', *International Journal of Geographical Information Science* **31**(10), 2001–2019.

Roberts, D. R., Bahn, V., Ciuti, S., Boyce, M. S., Elith, J., Guillera-Arroita, G., Hauenstein, S., Lahoz-Monfort, J. J., Schröder, B., Thuiller, W., Warton, D. I., Wintle, B. A., Hartig, F. & Dormann, C. F. (2017), 'Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure', *Ecography* **40**(8), 913–929.
URL: *http://onlinelibrary.wiley.com/doi/abs/10.1111/ecog.02881*

Smith, W. B. (2002), 'Forest inventory and analysis: a national inventory and monitoring program', *Environmental pollution* **116**, S233–S242.

Tatsumi, S., Owari, T. & Mori, A. S. (2016), 'Estimating competition coefficients in tree communities: a hierarchical bayesian approach to neighborhood analysis', *Ecosphere* **7**, e01273.

Uriarte, M., Condit, R., Canham, C. D. & Hubbell, S. P. (2004), 'A spatially explicit model of sapling growth in a tropical forest: does the identity of neighbours matter?', *Journal of Ecology* **92**(2), 348–360. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.0022-0477.2004.00867.x.
URL: *http://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/j.0022-0477.2004.00867.x*

Uriarte, M., Swenson, N. G., Chazdon, R. L., Comita, L. S., Kress, W. J., Erickson, D., Forero-Montaṅa, J., Zimmeran, J. K. & Thompson, J. (2010), 'Trait similarity, shared ancestry and the structure of neighbourhood interactions in a subtropical wet forest: implications for community assembly', *Ecology Letters* **13**, 1503–1514.

Valavi, R., Elith, J., Lahoz-Monfort, J. J. & Guillera-Arroita, G. (2019), 'blockCV: An r package for generating spatially or environmentally separated folds for k-fold cross-validation of species distribution models', *Methods in Ecology and Evolution* **10**(2), 225–232. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/2041-210X.13107.
URL: *http://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/2041-210X.13107*

Waller, L. A. & Gotway, C. A. (2004), *Applied Spatial Statistics for Public Health Data*, John Wiley & Sons, Incorporated, Hoboken, UNITED STATES.
URL: *http://ebookcentral.proquest.com/lib/smith/detail.action?docID=214360*

Wickham, H. (2020), *tidyr: Tidy Messy Data*. R package version 1.1.2.
URL: *https://CRAN.R-project.org/package=tidyr*

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., Takahashi, K.,

Vaughan, D., Wilke, C., Woo, K. & Yutani, H. (2019), 'Welcome to the Tidyverse', *Journal of Open Source Software* **4**(43), 1686.
URL: *https://joss.theoj.org/papers/10.21105/joss.01686*