

ALB Spread Simulation

Lora Murphy

The ALB spread simulation takes a set of trees with an initial ALB infestation, and models insect spread for a desired length of time.

Required setup

The APHIS simulation is performed in R, and takes advantage of R's spatial analysis capabilities. (More on R and spatial data at <https://cran.r-project.org/view=Spatial>.)

To get set up:

1. Install the latest version of R. <https://cran.r-project.org/>
2. If you are working on a computer running Windows, install RTools, which can be found under the R Binaries section of the website above. Follow the directions to make sure R and RTools can find each other.
3. Install the additional packages “Rcpp”, “sp” and “sf”. Be sure to use dependencies=TRUE when you do:

```
install.packages(c("sp", "sf", "Rcpp"), dependencies=TRUE)
```

4. Additionally, the packages “magick”, “ggplot2”, and “ggpubr” may be required if you want to use functions in the provided graphics script (not required to perform simulations).

How to run a simulation

Open the script called “Simulation.R”. Adjust the settings you see there as desired, and run in R.

Best practice is to save a copy of this script for each run you do. This provides a record of your settings.

The other scripts do the heavy lifting. They will take their settings from “Simulation.R” and you shouldn't need to adjust them to run the model.

Input

The input is a dataframe of trees that are potential ALB victims. All are assumed to be Acers. The dataframe should have the following fields, with these exact case-sensitive names (there can be other fields, they'll be ignored):

- “x”, “y”: location coordinates. Point of origin doesn't matter, this will only be used to calculate distances between points. SPECIFY THE LINEAR UNITS BELOW (feet, meters).

- “mean_noforestdist”: the mean distance to the forest edges in each of the 8 cardinal and intercardinal directions, in m. If the tree isn’t in a forest landcover, this is 0.
- “dbh”: tree’s DBH in cm

OPTIONAL: - “infested”: integer, 0 being uninfested, 1 being infested. If this field is included and there is at least 1 tree with a 1 status, the trees marked as 1 will be the seed of the outbreak. If either this field is not included or all the values are 0, a select number of trees will be randomly assigned to be the seed of the outbreak (number settable below).

The “mean_noforestdist” field can be calculated with the script “Get non forest landcover in 8 directions.R.”

Output

Two objects get saved for every run: the trees, and the surveys done.

Trees

The trees are saved as a `SpatialPointsDataFrame` that is a copy of the input dataframe, with additional fields for dates of infestation and removal (“year_infested” and “year_removed”).

A `SpatialPointsDataFrame` is a dataframe with spatial information for each line, equivalent to a points shapefile in ArcGIS. In most cases, dataframe syntax can be used on it directly; for instance, the \$ operator will work as expected. It can be coerced into a regular dataframe using `as.data.frame`.

The `SpatialPointsDataFrame` can be easily be plotted as a set of points on a map using `plot` (for example: `plot(trees)` is all that’s necessary), or used in other spatial analyses. It can also be written out as a shapefile compatible with a variety of GIS software using the “`rgdal`” package in R.

Surveys

The surveys performed (if any) are saved as a list, with one slot for each year of the simulation. An empty list slot indicates that no survey occurred that year. If a survey did occur, it is saved as a `SpatialPolygons` object with one or more polygons indicating the surveyed regions. You can visualize the surveys using the `plot` command (see included graphics file for more detailed examples).

The trees and surveys are each saved in their own RDS file by default. This stores them in a native R format. Read them back in using the `readRDS` command, assigning whatever name you like. For example:

```
trees <- readRDS("mytrees.rds")
surveys <- readRDS("mysurveys.rds")
plot(trees)
plot(surveys[[3]], add=T)
```

Model flow

1. Choose the risk model and do any necessary setup.
2. Loop through the years of the simulation:
 - a. Calculate this year’s probability of infestation for potential hosts, based on the chosen risk model.

- b. Choose the trees that get infested this year. For each tree, compare a random number to its probability to make the choice.
 - c. For those trees that get infested, record the year they get infested to both track time until emergence, and generally track spread in output.
 - d. Apply management. As parameters, 4 different probabilities are defined for detection and removal: post-emergence tree in surveyed area, pre-emergence tree in surveyed area, post-emergence tree NOT in surveyed area, pre-emergence tree NOT in surveyed area.
 - e. Decide what's being surveyed: identify trees that were detected in the previous year. Define a circle of radius 1.5 miles around each detected tree: all areas so marked are surveyed areas. (This is limited to 500 trees, for computer memory reasons.) If no trees were detected in the previous year, no surveys will be done.
 - f. Identify all post-emergence trees within surveyed areas, and use a random number against the probability of detection of these trees to choose which are detected and removed.
 - g. If the removal budget has not been met, identify all pre-emergence infested trees within surveyed areas. Use a random number on each, compared to the detection probability, to choose which got detected and removed.
 - h. Repeat steps b and c, with non-surveyed areas and post- and pre-emergence trees. All detected trees will be removed, up to the budget.
 - i. Apply tree growth to all live trees.
3. Clean up; create graphics; save output files.

Risk model: determining which trees are infested

Uninfested trees have a certain probability of being infested each timestep, as a function of number of and distance to post-emergence infested source trees and various tree characteristics (see below).

Source trees are trees which are infested, post-emergence, and have not been removed. A tree which is infested is pre-emergence until the lag period has passed (see biology settings of the model). Pre-emergence trees do not infest other trees. The lag period could be set to 0 to remove the pre- and post-emergence distinction.

There are multiple possible risk models. This simulator is the result of analyses conducted in several different geographic regions in the U.S. These regions had different risk models that best captured their local conditions and landscapes.

Risk model parameters

Each risk model requires several parameter values to control its behavior. The risk models were parameterized with local data in each geographic region. The best-fit parameter values are included as defaults with each model.

Risk model 13

This model assesses tree risk as a function of source pressure, DBH of tree, number of nearby Acer neighbors, and landcover. This model best described conditions in Long Island, NY.

The probability that a tree will be infested is:

$$prob = \mu * source.term * dbh.term * density.term * distance.term$$

where μ is a parameter representing the upper limit of infestation probability.

$$source.term = \frac{1}{1 + \left(\frac{sp}{\alpha}\right)^\gamma}$$

where sp is a measure of source pressure and α and γ are parameters.

$$sp = \sum_{i=1}^N \beta * e^{(dsn * (distance_i^\delta))}$$

for N source trees within the maximum distance of 5280 feet, where dsn , β , and δ are parameters and $distance_i$ is the distance from the source tree to the i th neighbor.

$$dbh.term = \frac{1}{1 + \left(\frac{DBH}{b_1}\right)^{b_2}}$$

where DBH is tree's DBH, in cm, and b_1 and b_2 are parameters.

$$density.term = \frac{1}{1 + \left(\frac{N_{acer}}{c_1}\right)^{c_2}}$$

where N_{acer} is the number of Acer neighbors within 30 m of the target (will be calculated and updated by the script) and c_1 and c_2 are parameters.

$$distance.term = \frac{1}{1 + \left(\frac{dist_{lc}}{e_1}\right)^{e_2}}$$

where $dist_{lc}$ is the mean distance to no-forest landcover and e_1 and e_2 are parameters.

Risk model 11

This model assesses tree risk as a function of source pressure, DBH of tree, number of nearby Acer neighbors, and landcover. This was the best model in New York City and Worcester, MA.

The probability that a tree will be infested is:

$$prob = \mu * source.term * dbh.term * density.term * distance.term$$

where μ is a parameter representing the upper limit of infestation probability.

$$source.term = \frac{1}{1 + \left(\frac{sp}{\alpha}\right)^\gamma}$$

where sp is a measure of source pressure and α and γ are parameters.

$$sp = \sum_{i=1}^N \beta * dir_i * (d_1 + d_2)$$

for N source trees within the maximum distance of 5280 feet, where β is a parameter, dir_i is a parameter for the effect of direction from the tree to source i , in increments of 30 degrees, north 0 and increasing in the clockwise direction (0-1). d_1 and d_2 are separate near and far distance kernels.

$$d_1 = \delta_1 * \exp\left(\frac{dsn}{1000} * dist_i\right)$$

$$d_2 = \delta_2 * \exp\left(-0.5 * \left(\frac{\ln\left(\frac{dist_i}{dof}\right)}{dbf}\right)^2\right)$$

where $\delta_1, \delta_2, dsn, dof$ and dbf are parameters, and $dist_i$ is the distance to the source tree.

$$dbh.term = \frac{1}{1 + \left(\frac{DBH}{b_1}\right)^{b_2}}$$

where DBH is tree's DBH, in cm, and b_1 and b_2 are parameters.

$$density.term = \frac{1}{1 + \left(\frac{N_{acer}}{c_1}\right)^{c_2}}$$

where N_{acer} is the number of Acer neighbors within 30 m of the target (will be calculated and updated by the script) and c_1 and c_2 are parameters.

$$distance.term = \frac{1}{1 + \left(\frac{dist_{lc}}{e_1}\right)^{e_2}}$$

where $dist_{lc}$ is the mean distance to no-forest landcover and e_1 and e_2 are parameters.

Tree growth

In the case of a long simulation, growth may change a tree's risk over time. This script can grow trees using a linear function of DBH:

$$DBH_{t+1} = DBH_t + (m * DBH_t + b)$$

where this year's DBH is DBH_{t+1} , last year's DBH is DBH_t , m is the slope of growth, and b is the intercept. Either m or b can be set to 0.