MakeMyForests: Using USDA Forest Service Forest Inventory and Analysis data to parameterize SORTIE-ND

Samantha L. Davis

June 22, 2015

1 Introduction

MakeMyForests is an R package designed with a relatively narrow objective: to develop parameters for SORTIE-ND. SORTIE-ND is an individual-tree forest gap model that focuses on neighborhood dynamics to drive forest succession. In other words, SORTIE-ND models forest dynamics based on the growth and success of individual trees, instead of the success of a stand or species. This type of model can be slow and computationally intensive, because it performs calculations for each time step on each tree, which can number in the thousands.

SORTIE-ND was originally conceived of by Pacala et al. (1996), and extensively tested with data from eastern North American deciduous forests. Although SORTIE-ND is incredibly useful "out of the box" for eastern forests, it is not as accessible for those interested in other forest ecosystems. This R package is intended to help those interested in applying SORTIE-ND to new systems. Specifically, this R package will help calculate broad-brush parameters for individual species by taking advantage of the extensive USDA Forest Service's Forest Inventory and Analysis dataset, which is a survey of trees across the United States. The examples contained within MakeMyForests are from the California data available at the FIA Data Mart (current as of 2015-06). We primarily use individual state's "TREE.CSV" files, but you'll also need the REF_SPECIES.CSV (current as of 2015-06) file from the same location. CA_TREE and REF_SPECIES are included with this package.

What follows is code that will guide you through parameter generation for several species of California trees. If you need help with the individual functions, use the ?function. These functions are not extensively tested with "wrong" data, so please be sure to read the help files if you're unsure about input or output within a particular function.

2 Installing MakeMyForests

MakeMyForests is a standalone package, but is not currently available on CRAN. To download MakeMyForests, you need to access the package through GitHub. Luckily, the devtools package has a function to download and install packages from GitHub directly, without the use of a browser.

```
> ## install and load the devtools package
> #install.packages("devtools")
> library(devtools)
> ## install and load the MakeMyForests package
> #install_github("ecology-rocks/MakeMyForests")
> library(MakeMyForests)
```

3 Selecting Our Species

Now that we have MakeMyForests installed, we'll need to select the species we want to work with. For ease of use, MakeMyForests comes with the FIA data for California already loaded as an object called FullTreesCA. A quick look shows that this object is rather large.

```
> nrow(FullTreesCA)

[1] 296343

> ncol(FullTreesCA)

[1] 154

> head(colnames(FullTreesCA))

[1] "CN"          "PLT_CN"      "PREV_TRE_CN" "INVYR"       "STATECD"      "UNITCD"

> tail(colnames(FullTreesCA))

[1] "CENTROID_DIA_HT"         "CENTROID_DIA_HT_ACTUAL" "UPPER_DIA"
[4] "UPPER_DIA_HT"            "VOLCSSND"               "DRYBIO_SAWLOG"

>
```

For our purposes, we definitely do not need all of the 154 columns. Nor do we need all of the different species that can be found in FullTrees CA. For this exercise, we will pick 11 species from FullTrees CA and only a few of the columns. Our end data.frame will look something like this:

```
> str(TreesCA)
```

```
'data.frame':          180115 obs. of  16 variables:
 $ PLT_CN   : num  12636 12636 12636 12636 12636 ...
 $ TREE     : int  685099 685100 685101 685102 685107 685106 687153 687178 687159 687167 ...
 $ SPCD     : int  15 15 15 15 15 15 15 15 15 15 ...
 $ DIA      : num  6.86 9.65 11.68 8.13 53.59 ...
 $ HT       : num  4.88 6.1 5.18 4.88 24.08 ...
 $ CR       : int  65 65 45 45 75 75 75 75 85 55 ...
 $ CDENCD   : int  NA NA NA NA NA NA NA NA NA NA ...
 $ TRANSCD  : int  NA NA NA NA NA NA NA NA NA NA ...
 $ BHAGE    : int  41 41 48 58 83 46 20 7 30 25 ...
 $ TOTAGE   : int  61 61 68 78 103 66 40 27 50 45 ...
 $ CLIGHTCD : int  NA NA NA NA NA NA NA NA NA NA ...
 $ H1       : num  64.3 64.3 64.3 64.3 64.3 ...
 $ b0       : num  4.5 4.5 4.5 4.5 4.5 ...
 $ b1       : num  0.924 0.924 0.924 0.924 0.924 ...
 $ b2       : num  -0.012 -0.012 -0.012 -0.012 -0.012 -0.012 -0.012 -0.012 -0.012 -0.012 ...
 $ minAge   : num  40 40 40 40 40 40 40 40 40 40 ...

> head(TreesCA)

      PLT_CN   TREE SPCD    DIA      HT CR CDENCD TRANSCD BHAGE TOTAGE CLIGHTCD      H1
16080  12636 685099   15  6.858  4.8768 65     NA      NA    41     61       NA 64.3128
16081  12636 685100   15  9.652  6.0960 65     NA      NA    41     61       NA 64.3128
16082  12636 685101   15 11.684  5.1816 45     NA      NA    48     68       NA 64.3128
16083  12636 685102   15  8.128  4.8768 45     NA      NA    58     78       NA 64.3128
16089  12636 685107   15 53.594 24.0792 75     NA      NA    83    103       NA 64.3128
16095  12636 685106   15 37.084 21.9456 75     NA      NA    46     66       NA 64.3128
          b0     b1     b2 minAge
16080 4.4965 0.9238 -0.012     40
16081 4.4965 0.9238 -0.012     40
16082 4.4965 0.9238 -0.012     40
16083 4.4965 0.9238 -0.012     40
16089 4.4965 0.9238 -0.012     40
16095 4.4965 0.9238 -0.012     40
```

But, to get to the trimmed table, we'll need to select our species and subset FullTreesCA. Here's our list of soft- and hard-woods, by species short code. The species short codes for your species of interest can be found at the USDA PLANTS Database. After you get a vector of shortcodes, you can subset FullTreesCA in a two-step process. First, you'll need to get the FIA special "SPCD" codes from the REF_SPECIES table, and then you'll need to subset based on those codes. The REF_SPECIES table is included in this package as "RefSpecies."

```
> ## Manually species shortcodes from USDA PLANTS Database in a vector
> speciesShortCodes <- c("ABCO", "ABMA", "CADE27",
+                        "PIMO3", "PICO", "PIJE",
```

```
+                              "PIPO", "PILA", "PSME",
+                              "QUCH2", "QUKE")
> ## Get SPCD from your list of species short codes
> speciesCodes <- pullSpeciesCodes(speciesShortCodes,
+                                  RefSpecies)
> ## subset FullTreesCA based on the speciesCodes, and select 12 columns out of
> ## the original 154.
> trees <- pullTrees(speciesCodes, FullTreesCA,
+                  c("PLT_CN", "TREE", "STATUSCD",
+                    "SPCD", "DIA", "HT",
+                    "CR", "CDENCD", "TRANSCD",
+                    "BHAGE", "TOTAGE", "CLIGHTCD"))
> ## examine the resulting ``trees'' data.frame, which should be much easier to
> ## work with.
> str(trees)

'data.frame':         180115 obs. of  12 variables:
 $ PLT_CN  : num  12636 12636 12636 12636 12636 ...
 $ TREE    : int  685099 685100 685101 685102 685107 685106 687153 687178 687159 687167 ...
 $ STATUSCD: int  1 1 1 1 1 1 1 1 1 1 ...
 $ SPCD    : int  15 15 15 15 15 15 15 15 15 15 ...
 $ DIA     : num  6.86 9.65 11.68 8.13 53.59 ...
 $ HT      : num  4.88 6.1 5.18 4.88 24.08 ...
 $ CR      : int  65 65 45 45 75 75 75 75 85 55 ...
 $ CDENCD  : int  NA NA NA NA NA NA NA NA NA NA ...
 $ TRANSCD : int  NA NA NA NA NA NA NA NA NA NA ...
 $ BHAGE   : int  41 41 48 58 83 46 20 7 30 25 ...
 $ TOTAGE  : int  61 61 68 78 103 66 40 27 50 45 ...
 $ CLIGHTCD: int  NA NA NA NA NA NA NA NA NA NA ...
```

You may notice that our "trees" object has one column more than "TreesCA"
– this is because TreesCA was generated as a sample object early in the process,
and did not have the "STATUSCD" column, which is necessary for mortality
calculations.

Also, an important thing to note. pullTrees() not only subsets the data, it
also converts it from the standard feet and inches to metric meters and centime-
ters. This conversion is wrapped in the pullTrees function, which means that
if you just subset your TREE.CSV file manually, the numbers will still be in
feet and inches. If you want to manually subset and convert later, you can use
?unitConvert to do so.

Finally, we need to set up an object to hold our parameters as we gather
them. I've called this data.frame "speciesInfo", and created it below:

```
> speciesInfo <- data.frame(shortcode=speciesShortCodes,
+                           SPCD=speciesCodes,
+                           stringsAsFactors=FALSE)
> speciesInfo
```

```
   shortcode SPCD
1       ABCO   15
2       ABMA   20
3     CADE27   81
4      PIMO3  119
5       PICO  108
6       PIJE  116
7       PIPO  122
8       PILA  117
9       PSME  202
10     QUCH2  805
11      QUKE  818
```

4   Allometry Parameters

In this section, we'll figure out how to calculate different allometric parameters required by SORTIE-ND. The first parameter, asymptotic (maximum) height, is relatively simple to calculate. Pacala et al. (1996) nicknamed this parameter "H_1", and we will follow the convention. In essence, the way we calculate H_1 is to use the max() function in R. However, we do this within a wrapper that you'll see time and time again, "doFxBySort()". This function takes four parameters: the function to use; the sorting column, the column to do the function on, and that data.frame to use. This allows us to perform a function by a sorting variable, like "SPCD" (the species code), without having to wrap in a for-loop or mess with the "apply" class of functions in R. The "HT" column is the height of an individual tree.

```
> speciesInfo$H1=doFxBySort(max,
+                           "SPCD",
+                           "HT",
+                           trees
+                           )
> head(speciesInfo)

  shortcode SPCD      H1
1      ABCO   15 64.3128
2      ABMA   20 67.0560
3    CADE27   81 58.2168
4     PIMO3  119 54.5592
5      PICO  108 44.8056
6      PIJE  116 58.5216
```

Part of the reason that we need H_1 is that we need to calculate H_3, or the slope of asymptotic height. Pacala et al. (1996) established the relationship between these two variables in the following equation:

$$h = H_1(1 - e^{-(H_2/H_1)d}) \tag{1}$$

where h is height, H_1 is the asymptotic (maximum) height of a particular species, H_2/H_1 is the slope of asymptotic height, and d is the DBH. For convenience, we will refer to the H_2/H_1 parameter as H_3. To calculate H_3, we'll need to reconstruct the model in equation 1. We can do that with the function getH3, wrapped in doFxBySort for each species. To make it easier, first we'll need to put H_1 into a new column in "trees", so that it can be readily accessed by the non-linear model. We'll do this with the function "putChar".

```
> ## put a vector of H1 values into "trees" according to species
> trees$H1 <- unlist(putChar(trees,
+                            speciesInfo[,c("SPCD", "H1")],
+                            "H1"))
> ## now, find the H3 values for each species.
> speciesInfo$H3 <- doFxBySort(getH3,
+                              "SPCD",
+                              c("HT", "H1", "DIA"),
+                              trees
+                              )
> ## look at new speciesInfo table
> speciesInfo

   shortcode SPCD      H1        H3
1       ABCO   15 64.3128 0.3551388
2       ABMA   20 67.0560 0.3987871
3     CADE27   81 58.2168 0.2950537
4      PIMO3  119 54.5592 0.4175876
5       PICO  108 44.8056 0.4547066
6       PIJE  116 58.5216 0.3581279
7       PIPO  122 68.8848 0.3426520
8       PILA  117 75.5904 0.4182356
9       PSME  202 82.2960 0.3555081
10     QUCH2  805 46.9392 0.2115674
11      QUKE  818 46.9392 0.3213840
```

Obviously, we need more than just the height and diameter relationship if we're going to model trees. We also need crown characteristics. SORTIE-ND makes use of several crown characteristics for its model, including the ratio of tree crown radius (m) to stem diameter (cm), known as C_1 in Pacala et al. (1996). This can be hard to estimate for some species, because crown radius is not an often-measured characteristic in forestry. If you are working outside of the Western United States, you may need to consult the scientific literature to either find appropriate measures, or find models that correlate crown radius to some aspect of the data that you already have.

In our case, we found a paper, Bechtold (2004), that has published models for the relationship between crown diameter and stem diameter for each of our

11 species. Since we have stem diameter, we can predict crown diameter, and then calculate the ratio of crown radius to stem diameter for each species. The equation used to relate stem diameter and crown diameter is as follows:

$$MCD = b_0 + b_1(d) + b_2(d^2) \tag{2}$$

where MCD is mean crown diameter, b_0, b_1, and b_2 are species specific constants established previously by Bechtold (2004), and d is DBH. So, to calculate this ratio, we need to manually import the parameters and then apply the model to our trees by species to get mean crown radius, then divide by stem diameter.

```
> ## manually imported from Table 3 in Bechtold 2004
> b0 <- c(4.4965, 4.7623, 4.1207,
+         4.284, -1.1994, 4.2675,
+         2.3081, 4.8657, 5.7753,
+         6.1397, 7.0284)
> b1 <- c(0.9238, 0.5222, 0.9773,
+         0.6949, 1.5151, 0.7714,
+         1.1388, 0.789, 1.0639,
+         1.0109, 1.047)
> b2 <- c(-0.012, 0, -0.0107,
+         0, -0.0232, 0,
+         -0.0089, 0, -0.0109,
+         0, 0)
> ## Put those into a separate dataframe for ease...
> speciesC1params <- data.frame(SPCD=unique(trees$SPCD),
+                               b0=b0, b1=b1, b2=b2)
> ## put b0, b1, and b2 into trees based on species
> trees$b0 <- unlist(putChar(trees,
+                            speciesC1params[,c(1:2)], "b0"))
> trees$b1 <- unlist(putChar(trees,
+                            speciesC1params[,c(1,3)], "b1"))
> trees$b2 <- unlist(putChar(trees,
+                            speciesC1params[,c(1,4)], "b2"))
> ## calculate C1 for each species from diameter
> speciesInfo$C1 <- doFxBySort(getC1,
+                              "SPCD",
+                              c("DIA", "b0", "b1", "b2"),
+                              trees
+                              )
>
```

The next two parameters, C_2 and E_1, are actually already available in the FIA data. We just need to clean them up a bit. C_2 is the crown ratio, or the crown length (m) divided by the tree height (m). It can be found in the TREES.CSV file under the column name "CR" for crown ratio. E_1 is the "light extinction coefficient", or the amount of light that reaches the ground after

penetrating the canopy. This is also found in the original FIA data, under the column name "TRANSCD", or transparency code. For both of these parameters, we can simply take the average by species, removing NA values to avoid errors, and dividing by 100 to get a value between 0 and 1.

```
> speciesInfo$C2 <- (doFxBySort(mean,
+                               "SPCD",
+                               "CR",
+                               trees,
+                               extraParams=list(na.rm=TRUE))
+                 )/100
> speciesInfo$E1 <- (doFxBySort(mean,
+                               "SPCD",
+                               "TRANSCD",
+                               trees,
+                               extraParams=list(na.rm=TRUE))
+                 )/100
> speciesInfo
```

	shortcode	SPCD	H1	H3	C1	C2	E1
1	ABCO	15	64.3128	0.3551388	0.07894895	0.5035585	0.1746172
2	ABMA	20	67.0560	0.3987871	0.06039629	0.5128897	0.1460145
3	CADE27	81	58.2168	0.2950537	0.08597444	0.4761010	0.1871859
4	PIMO3	119	54.5592	0.4175876	0.06580801	0.6069731	0.1760504
5	PICO	108	44.8056	0.4547066	0.06064882	0.5478239	0.1855140
6	PIJE	116	58.5216	0.3581279	0.07168850	0.5349663	0.2367560
7	PIPO	122	68.8848	0.3426520	0.07658776	0.5037061	0.2107527
8	PILA	117	75.5904	0.4182356	0.07390936	0.5041290	0.1912698
9	PSME	202	82.2960	0.3555081	0.09895018	0.4751789	0.1939979
10	QUCH2	805	46.9392	0.2115674	0.13272686	0.4213389	0.2272112
11	QUKE	818	46.9392	0.3213840	0.12621840	0.3936617	0.2448755

5   Growth, Death, and Reproduction

We need a few other parameters before we can get SORTIE-ND to run successfully. These are not really allometric parameters, but instead deal with growth, reproduction, and death. SORTIE-ND needs the minimum DBH at which a tree is reproductively active, parameters representing the relationship between growth at high- and low- light, and some version of mortality. We will go through each of these below.

5.1   Minimum Adult DBH

SORTIE-ND needs to know when a tree can start producing offspring, as some trees may be large (saplings or poles), but unable to have meaningful reproductive output. Unfortunately, the minimum adult DBH is a vague and challenging

8

piece of information to find about individual trees. We used a combination of literature search and modeling to estimate the minimum adult DBH of our eleven tree species. We retrieved estimates of minimum reproductive *age* from Burns and Honkala (1990), an excellent resource for trees in North America. We then developed a simple linear model to predict the relationship between DBH and age from the FIA data. Finally, we put the estimated minimum DBH into speciesInfo. You'll see one new function, called "predictYfromLin" – this is exactly what it sounds like. It takes a formula ("DIA BHAGE") and predicts new values of Y.

```
> ## Original minimum adult ages from literature searches.
> speciesInfo$minAge <- c(40, 35, 10,
+             7, 5, 8,
+            16, 40, 15,
+            20, 30)
> ## Put the minimum age into trees by species
> trees$minAge <- unlist(
+                      putChar(trees,
+                          speciesInfo[,c("SPCD", "minAge")],
+                          "minAge")
+                  )
> ## predict Y from a regression using the minAge column in trees
> speciesInfo$minDBH <- doFxBySort(predictYfromLin,
+                          "SPCD",
+                          c("minAge", "DIA", "BHAGE"),
+                          trees,
+                          extraParams=list(formula="DIA~BHAGE")
+                      )
>
```

5.2 Growth in High and Low Light

Another important piece of information that SORTIE-ND needs is how trees grow and respond to different light conditions. Specifically, SORTIE-ND needs the parameters of the following equation for each species:

$$Y = a + (b * GLI) \tag{3}$$

Where Y is the yearly increase in diameter, a is the species-specific intercept (growth at zero light), b is the species-specific slope (maximum yearly growth), and GLI is the global light index for the individuals, or the amount of light an individual receives. If we can estimate GLI and Y, we can calculate the values for *a* and *b*. Luckily, the FIA data has a column called "CLIGHTCD" which is a canopy light code. It ranges between 1-5 and represents no light received by the canopy (1) to light received from all sides (5). We can convert these into numerical equivalents of GLI. For Y, or the yearly diameter increase, we can estimate mean yearly increase by dividing the age of a tree by its current

9

DBH. Not all trees have all three of these values, so we first need to prepare by subsetting the data.frame to include only those trees with values in each column:

```
> ## This just subsets the data into trees with both BHAGE and CLIGHTCD available,
> ## and also does some housecleaning by converting CLIGHTCD into a usable format,
> ## calculating dia/yr, etc.
> LightedTrees <- prepLightData(trees)
> head(LightedTrees)
```

	PLT_CN	TREE	STATUSCD	SPCD	DIA	HT	CR	CDENCD	TRANSCD	BHAGE	TOTAGE
44884	7.113485e+12	112	1	15	32.258	14.3256	80	40	15	115	NA
44891	7.113485e+12	119	1	15	16.764	8.5344	80	35	20	140	NA
45486	2.367789e+13	103	1	15	79.248	25.6032	50	35	15	223	NA
45490	2.367789e+13	107	1	15	65.786	30.7848	45	30	15	247	NA
45815	2.369259e+13	103	1	15	18.288	9.1440	55	35	20	60	NA
45821	2.369259e+13	109	1	15	23.114	10.0584	65	30	20	65	NA

	CLIGHTCD	H1	b0	b1	b2	minAge	light	diayr
44884	2	64.3128	4.4965	0.9238	-0.012	40	40	0.2805043
44891	0	64.3128	4.4965	0.9238	-0.012	40	0	0.1197429
45486	1	64.3128	4.4965	0.9238	-0.012	40	20	0.3553722
45490	4	64.3128	4.4965	0.9238	-0.012	40	80	0.2663401
45815	2	64.3128	4.4965	0.9238	-0.012	40	40	0.3048000
45821	0	64.3128	4.4965	0.9238	-0.012	40	0	0.3556000

Now that we've calculated yearly diameter increase and a "light" column representing GLI, we can go ahead and compute the linear model in equation 3 and pull out the parameters. For convenience, I've done exactly that with "doSpecies", which is a little different (older) than doFxBySort, but can create a glm just fine. If you want to explore each model, you'll need to do that manually; this function only calculates the model and pulls the slope and intercept. It takes your trimmed dataframe, the function, and the parameter you want to pull out. doSpecies will also perform bootstrapped mean and standard deviation, check its help file (?doSpecies) for more information.

```
> ## put sloep and intercept of yearly growth as predicted by light into
> ## speciesInfo
> speciesInfo$SlopeYrlyGrowth <- doSpecies(LightedTrees, "glm", "slope")
> speciesInfo$IntYrlyGrowth <- doSpecies(LightedTrees, "glm", "int")
> speciesInfo
```

	shortcode	SPCD	H1	H3	C1	C2	E1	minAge	minDBH
1	ABCO	15	64.3128	0.3551388	0.07894895	0.5035585	0.1746172	40	23.163518
2	ABMA	20	67.0560	0.3987871	0.06039629	0.5128897	0.1460145	35	23.188689
3	CADE27	81	58.2168	0.2950537	0.08597444	0.4761010	0.1871859	10	15.464952
4	PIMO3	119	54.5592	0.4175876	0.06580801	0.6069731	0.1760504	7	16.568590
5	PICO	108	44.8056	0.4547066	0.06064882	0.5478239	0.1855140	5	14.880187

```
6      PIJE  116 58.5216 0.3581279 0.07168850 0.5349663 0.2367560    8 21.681201
7      PIPO  122 68.8848 0.3426520 0.07658776 0.5037061 0.2107527   16 22.446221
8      PILA  117 75.5904 0.4182356 0.07390936 0.5041290 0.1912698   40 34.225840
9      PSME  202 82.2960 0.3555081 0.09895018 0.4751789 0.1939979   15 19.701955
10    QUCH2  805 46.9392 0.2115674 0.13272686 0.4213389 0.2272112   20  7.682973
11     QUKE  818 46.9392 0.3213840 0.12621840 0.3936617 0.2448755   30 14.775212
   SlopeYrlyGrowth IntYrlyGrowth
1     0.0030695554    0.33410330
2     0.0016242621    0.36173434
3     0.0009724445    0.38083175
4     0.0079308096    0.02557311
5     0.0002203619    0.30995944
6     0.0037690508    0.41232379
7     0.0036112229    0.39320445
8     0.0034015821    0.39670194
9     0.0029481737    0.41825450
10    0.0011033417    0.27430085
11   -0.0011541909    0.54688807
```

5.3 Mortality

Finally, we need to calculate mortality rate for each species. This is a pretty straightforward calculation using the "STATUSCD" column in "trees." First, you get a count of alive and dead trees, which you can do with doFxBySort. STATUSCD gives a code of 1 for an alive tree, and a code of 2 for a dead tree. So, to count the number of live trees, we can sum where STATUSCD is equal to 1. For dead trees, we can sum and then divide by 2. Then, we can get the proportion of dead, as well as the inverse for the survival rate.

```
> ## get number of live trees by species
> speciesInfo$NumAlive <- doFxBySort(sum,
+                                    "SPCD",
+                                    c("STATUSCD"),
+                                    trees[trees$STATUSCD==1,]
+                                    )
> ## get number of dead trees by species
> speciesInfo$NumDead <- doFxBySort(sum,
+                                    "SPCD",
+                                    c("STATUSCD"),
+                                    trees[trees$STATUSCD==2,]
+                                    )/2
> ## calculate mortality rate as the proportion of total trees that were dead
> speciesInfo$MortalityRate <- speciesInfo$NumDead /
+                          (speciesInfo$NumAlive + speciesInfo$NumDead)
> ## get the inverse for survival
> speciesInfo$SurvivalRate <- 1-speciesInfo$MortalityRate
```

```
> ## look at the updated speciesInfoTable
> speciesInfo
```

	shortcode	SPCD	H1	H3	C1	C2	E1	minAge	minDBH
1	ABCO	15	64.3128	0.3551388	0.07894895	0.5035585	0.1746172	40	23.163518
2	ABMA	20	67.0560	0.3987871	0.06039629	0.5128897	0.1460145	35	23.188689
3	CADE27	81	58.2168	0.2950537	0.08597444	0.4761010	0.1871859	10	15.464952
4	PIMO3	119	54.5592	0.4175876	0.06580801	0.6069731	0.1760504	7	16.568590
5	PICO	108	44.8056	0.4547066	0.06064882	0.5478239	0.1855140	5	14.880187
6	PIJE	116	58.5216	0.3581279	0.07168850	0.5349663	0.2367560	8	21.681201
7	PIPO	122	68.8848	0.3426520	0.07658776	0.5037061	0.2107527	16	22.446221
8	PILA	117	75.5904	0.4182356	0.07390936	0.5041290	0.1912698	40	34.225840
9	PSME	202	82.2960	0.3555081	0.09895018	0.4751789	0.1939979	15	19.701955
10	QUCH2	805	46.9392	0.2115674	0.13272686	0.4213389	0.2272112	20	7.682973
11	QUKE	818	46.9392	0.3213840	0.12621840	0.3936617	0.2448755	30	14.775212

	SlopeYrlyGrowth	IntYrlyGrowth	NumAlive	NumDead	MortalityRate	SurvivalRate
1	0.0030695554	0.33410330	33700	1416	0.04032350	0.9596765
2	0.0016242621	0.36173434	11001	624	0.05367742	0.9463226
3	0.0009724445	0.38083175	15527	442	0.02767863	0.9723214
4	0.0079308096	0.02557311	2045	88	0.04125645	0.9587436
5	0.0002203619	0.30995944	9154	296	0.03132275	0.9686772
6	0.0037690508	0.41232379	10092	164	0.01599064	0.9840094
7	0.0036112229	0.39320445	19152	367	0.01880219	0.9811978
8	0.0034015821	0.39670194	5689	191	0.03248299	0.9675170
9	0.0029481737	0.41825450	37226	1004	0.02626210	0.9737379
10	0.0011033417	0.27430085	19808	324	0.01609378	0.9839062
11	-0.0011541909	0.54688807	11462	343	0.02905548	0.9709445

6 Conclusions

There are certainly parameters that I was not able to address using FIA data. One major parameter that comes to mind is spatial seed dispersal. There is little-to-no information about seeds and seedlings available in the FIA data, especially as spatially informative data. To get a good measure of spatial or even non-spatial seed dispersal, you'll need to consult other data.

We have an advantage in our study system. There are some data available as exhaustive plot maps. These plot maps include our target species, and researchers have established (x,y) locations of every seedling, sapling, and adult tree in a given area. Where possible, the researchers also measured DBH. If you have these data available for your trees, I encourage you to check out the disperseR package that I am developing. You can download it in the same fashion as MakeMyForests:

```
> #install_github("ecology-rocks/disperseR")
> #library(disperseR)
```

and access the vignette available there to create spatial seed dispersal parameters.

References

Bechtold, W. A., 2004. Largest-Crown- Width Prediction Models for 53 Species in the Western United States. *Western Journal of Applied Forestry* **19**:245–251.

Burns, R. M. and B. M. Honkala, 1990. Silvics of North America: 1. Conifers. Technical report.

Pacala, S. W., C. D. Canham, J. Saponara, J. A. Silander, R. K. Kobe, and E. Ribbens, 1996. Forest models defined by field measurements: estimation, error analysis and dynamics. *Ecological Monographs* **66**:1–43.