# Allometry_testing

Kelly Heilman

9/4/2020

**Notes about setup–there are two options:**

1. Clone the pecan repo directly, then open this rmd in the repository. (Maybe best option because of dependancies)
2. You can install the allometry module from github with the code below (from allometry vignette): library(devtools) # # if not run, need to install the pecan modules install_github(“PecanProject/pecan”,subdir=“base/logg install_github(“PecanProject/pecan”,subdir=“allometry”)

## Basic Pecan allometry tutorial

Alot of this text and code is just annotated from the allometry vignette. To view the different components of the tree for which you could develop allomtrys + their abbreviations:

```
data("allom.components")
allom.components
```

```
##                                                description abbreviation ID
## 1                      Complete tree (above + belowground)          BTT  1
## 2                                 Whole tree (aboveground)          BAT  2
## 3                                 Whole tree (above stump)              3
## 4                                         Stem (wood only)          BSW  4
## 5                                         Stem (bark only)          BSB  5
## 6                                       Stem (wood + bark)          BST  6
## 7                                                 Stem top              7
## 8                                            Branches live          BBL  8
## 9                                   Branches live < 2.5 cm        BBL_1  9
## 10                               Branches live  2.5-7.6 cm        BBL_2 10
## 11                                 Branches live  > 7.6 cm        BBL_3 11
## 12                                           Branches dead          BBD 12
## 13                            Branches total (live + dead)          BBT 13
## 14                             Stem + branches (bark only)             14
## 15                             Stem + branches (wood only)             15
## 16                                  Stem + branches (live)          BAP 16
## 17 Wood, bark, branches (live + dead; no twigs or foliage)          BAE 17
## 18                                           Foliage total          BFT 18
## 19                                             Foliage new          BFN 19
## 20                                             Foliage old          BFO 20
## 21                                             Twigs total          BBG 21
## 22                                               Twigs old        BBG_O 22
## 23                                         Foliage + twigs          BFG 23
## 24                      Crown (branches + foliage + twigs)          BCT 24
## 25                                Roots, coarse > 3 mm dia          BKL 25
## 26                          Coarse stump roots          BSR 26
## 27                         Coarse lateral roots         BLR 27
## 28                                   Fine roots         BFR 28
## 29                                  Roots total         BRT 29
## 30                                   Stump wood             30
## 31                                   Stump bark             31
## 32                                  Stump total             32
## 33                                Stump + roots             33
## 34                                        Cones             34
## 35          Live crown (branches + foliage + twigs)     BCL 35
## 36          Dead crown (branches + foliage + twigs)     BCD 36
## 37                               Small branches         BBS 37
## 38                                       Height          HT 40
## 39                                Rooting Depth          Rd 41
## 40                               Rooting Volume         Vol 42
## 41                                  Canopy Area             43
```

```
pfts = list(PIPO = data.frame(spcd=122,acronym='PIPO')) # list our "Pfts--plant functional types" of in
```

## AllomAve function in PEcAn allometry module

- Fits the bayesian allomtric models for the pfts. ngibbs indicates the number of gibbs/mcmc samples
- looks like there is a burn in of ~200
- automatically outputs a pdf with the traceplots titled "allom.pipo.6.mcmc" and saves and .Rdata object.
- not totally clear to be what the different betas are

```
allom.stats = AllomAve(pfts,components =6, ngibbs=1000)
```

```
## [1] "writing output to"
## [2] "/home/miranda/agb_allometry/pecan/modules/allometry"
## $spcd
## [1] 122
##
## $acronym
## [1] PIPO
## Levels: PIPO
##
## [[3]]
## [1] 6
##
## [1] "Dropping allom rows: "
##   |                                                                               |
allom.stats$PIPO
```

```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
##
## [[5]]
## NULL
##
## [[6]]
##
## Iterations = 1:750
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 750
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean      SD Naive SE Time-series SE
## b0.1   -0.9523  0.7621 0.016065       0.016659
## b0.2   -0.9313  1.3981 0.029475       0.029477
## b0.3   -2.5069  0.7266 0.015319       0.021378
## b0.4   -2.7044  0.8791 0.018534       0.025256
## b0.5   -2.7890  0.9093 0.019170       0.024355
## b0.6   -3.4818  0.5453 0.011495       0.016797
## b1.1    1.1348  0.4301 0.009068       0.009437
## b1.2    1.1938  0.7471 0.015749       0.015750
## b1.3    2.3865  0.2451 0.005167       0.007142
## b1.4    2.3977  0.3038 0.006405       0.008457
## b1.5    2.5033  0.3103 0.006543       0.008246
## b1.6    2.6447  0.1303 0.002747       0.004001
## mu0    -2.2168  0.9403 0.019822       0.024052
## mu1     2.0343  0.5206 0.010976       0.011944
## sigma   0.1315  0.0685 0.001444       0.001684
## tau11   4.4226  8.2397 0.173709       0.251329
## tau12  -2.4383  5.2708 0.111119       0.145956
## tau22   1.5184  3.6768 0.077513       0.093979
## D      66.6461 60.7116 1.279913       1.530546
## Bg0    -2.3370  0.6158 0.012983       0.012996
## Bg1     2.3338  0.1759 0.003709       0.003621
## Sg      0.4620  0.2381 0.005019       0.004873
## Dg    183.4849 54.2587 1.143873       1.130324
##
## 2. Quantiles for each variable:
##
```

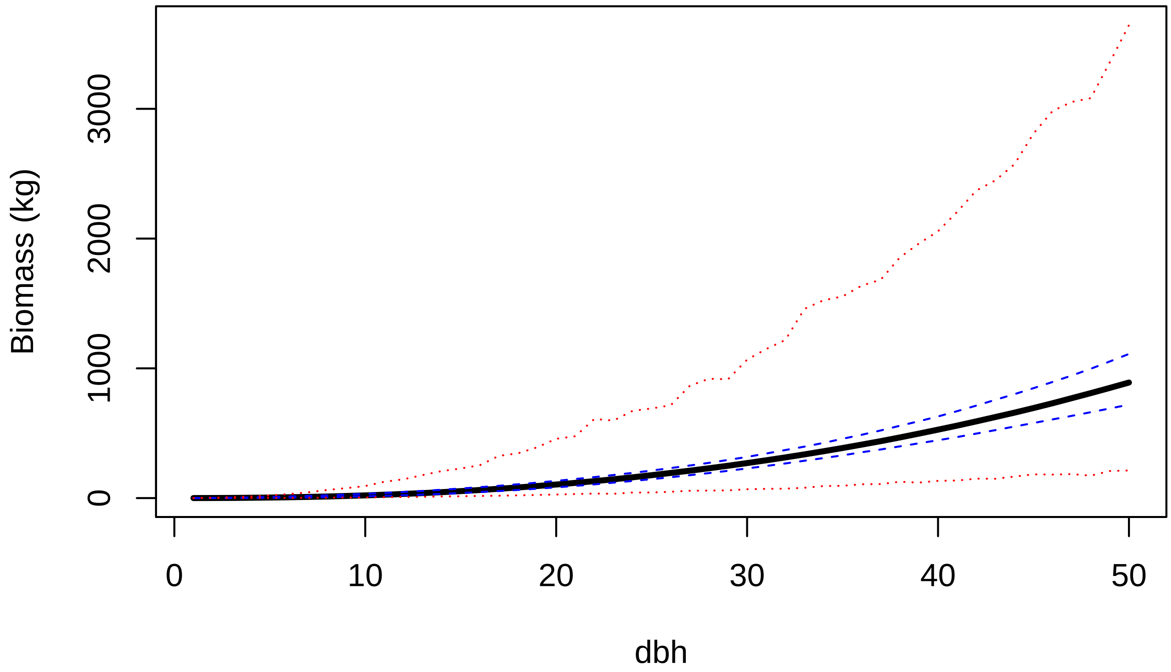
```
##            2.5%       25%      50%       75%    97.5%
## b0.1   -2.69658  -1.37241  -0.8308  -0.41975   0.2091
## b0.2   -4.50763  -1.64672  -0.6186   0.08552   0.9032
## b0.3   -3.92746  -2.96619  -2.5104  -2.05727  -1.0503
## b0.4   -4.47173  -3.26932  -2.6821  -2.13836  -0.9816
## b0.5   -4.59567  -3.35675  -2.7766  -2.23158  -1.0209
## b0.6   -4.54504  -3.83625  -3.4744  -3.14461  -2.3972
## b1.1    0.46900   0.82182   1.0737   1.38727   2.0667
## b1.2    0.16537   0.63913   1.0529   1.59236   3.0019
## b1.3    1.90295   2.23364   2.3886   2.54015   2.8621
## b1.4    1.80130   2.19708   2.3923   2.59235   3.0033
## b1.5    1.88430   2.31358   2.5032   2.69454   3.0920
## b1.6    2.38859   2.56335   2.6429   2.72939   2.9038
## mu0    -4.06829  -2.75280  -2.2013  -1.68668  -0.3246
## mu1     0.94255   1.75547   2.0396   2.32275   3.0656
## sigma   0.05771   0.08685   0.1132   0.15488   0.3190
## tau11   0.46344   1.43589   2.5545   4.83135  18.3376
## tau12 -10.03039  -2.62083  -1.4225  -0.79702  -0.2328
## tau22   0.19125   0.53116   0.9020   1.61909   5.8851
## D     -13.74784  27.78100  54.9703  91.45138 218.8039
## Bg0    -3.46674  -2.77279  -2.3433  -1.90377  -1.1147
## Bg1     1.98478   2.21225   2.3353   2.45911   2.6583
## Sg      0.13629   0.28115   0.4161   0.61514   1.0171
## Dg     72.84246 146.65021 187.6497 226.26288 276.6505
```

## Predict for individual trees:

- the allom.predict fuction will use the fit relationships from allomAve to predict the biomass component for the pft of interest (PIPO) over the DBH vector values:
- dbh = diameter to predict at
- component is which component of biomass to predict see defintation in the allom.components object
- unclear to me whether component = 6 is predicting Stem biomass, or is only suing the diameter of the stem in the allometry equation
- use = Bg

```
##Predict for individual trees:
allom.fit = load.allom(getwd()) # get the allometry listed in our working directory
dbh = 1:50 # vector of DBH values to predict over

pred = allom.predict(allom.fit, dbh = dbh, pft = "PIPO", component =6, use = "Bg", interval = "predicti
conf = allom.predict(allom.fit,dbh = dbh,pft = "PIPO",component = 6,use = "Bg",interval = "confidence")
PI = apply(pred,2,quantile,c(0.025,0.5,0.975),na.rm=TRUE)
CI = apply(conf,2,quantile,c(0.025,0.5,0.975),na.rm=TRUE)
plot(dbh,CI[2,],type='l',lwd=3,ylim=range(PI),ylab="Biomass (kg)")
lines(dbh,CI[1,],lty=2,col="blue")
lines(dbh,CI[3,],lty=2,col="blue")
lines(dbh,PI[1,],lty=3,col="red")
lines(dbh,PI[3,],lty=3,col="red")
```



### other stuff I learned from digging:

- there is a file in the pecan repository that has a table with all the beta parameters from jenkins
- really we want to use the query.allom.data function to query the “field” data, but there is a database that this function connects to via “con=” argument…and I can’t figure out where this database is located/what the connection should be

```
jenkins.table <- read.csv(system.file("data/Table3_GTR-NE-319.v2.csv", package = "PEcAn.allometry"), hea
#colnames(jenkins.table)[1] <- c("")
head(jenkins.table)
```

```
##   Table.3.....Equations.and.parameters.for.diameter.based.biomass.equations
## 1
## 2                                                                   Species
## 3                                                                         0
## 4                                                                         0
## 5                                                                         0
## 6                                                                         0
##                      X          X.1              X.2    X.3     X.4     X.5
## 1
## 2         Common Name Component ID Equation Form ID      a       b       c
## 3    eastern conifers            2                7    0.5   15000     2.7
## 4 softwoods (general)            3                1  -1.01    2.41       1
## 5 softwoods (general)            3                4 4.5966 -0.2364 0.00411
## 6 softwoods (general)            6                4  4.142  -0.227   0.003
##      X.6 X.7      X.8                X.9                 X.10 X.11 X.12
## 1
## 2      d   e Diameter Corrected for bias Bias correction (CF)    r   R2
## 3 364946       d.b.h.                 no                    0    0 0.98
## 4              d.b.h.                 no                    0    0 0.99
## 5      2       d.b.h.                 no                    0    0 0.96
## 6      2       d.b.h.                 no                    0    0 0.97
##          X.13        X.14        X.15         X.16       X.17           X.18
## 1
## 2 MinDiameter MaxDiameter Sample size Stump height Top d.o.b. Units diameter
## 3        1.00       72.00          83            0                          cm
## 4        0.80       34.10         108            6                          cm
## 5        2.50       25.00         131           12                          mm
## 6       12.50       55.00         131           12          0             mm
##            X.19      X.20          X.21           X.22               X.23
## 1
## 2 Units biomass Component Component sum Ratio equation Segmented equation
## 3            kg        na             a
## 4            kg        na
## 5            kg        na
## 6            kg        na
##              X.24   X.25
## 1
## 2 Equation number Source
## 3               1    140
## 4               1     51
## 5               1    107
## 6               1    107
##
## 1
## 2
## 3 assume 0-inch stump height, but data from other studies so stump heights are probably mixed; 43 pi
## 4                                                                          no stump height; tree data
## 5
## 6                                                                        12-inch stump including enti
```

```
# other functions
#??AllomAve


#??AllomUnitCoef # converts length units FROM cm TO specified units converts mass units TO kg FROM spec



#??allom.BayesFit #Module to fit a common power-law allometric model to a mixture of raw data and allom


#??allom.predict # Function for making tree-level Monte Carlo predictions from allometric equations est



#??load.allom() # load .allom files

# This is proabably what we want
#??query.allom.data# Module to grab allometric information from the raw data table Will grab both origi
                    #Tallied equation format based on Jenkins et al 2004 USFS General Technical Report

#??read.allom.data # Extracts PFT- and component-specific data and allometeric equations from the speci

# note that query.allom.data is not an exported fuction from the stand alone module....
```

```
#query.allom.data("PIPO")
```