Species Distribution Modeling in R

Daniel M Griffith

November 9, 2017

Introduction & Objectives

This workshop will follow a workflow from loading species occurrence data and environmental layers through to fitting simple species distribution models (SDM) and visualizing model outputs. The goal is to build R skills in manipulating spatial vector and raster data by focusing on an SDM application. Participants should be familiar with base R and RStudio. Previous work with spatial data in R is not required, but the workshop will assume familiarity with basic raster/vector data types.

- The primary objectives of this workshop are to learn basic spatial data management skills:
  - Several ways to load point and polygon data
  - Several ways to load raster layers
  - Combining spatial data of various types
  - Simple visualizations
  - Map projections
- Our secondary objective is to show how these data can be used to:
  - Fit a selection of SDMs
  - Map/visualize model results
  - Use model to predict future distributions

This will be a fairly basic demonstration and I encourage you to check the relevant literature for standards for your application, as well as the citations in this workshop.

Installation & load key packages

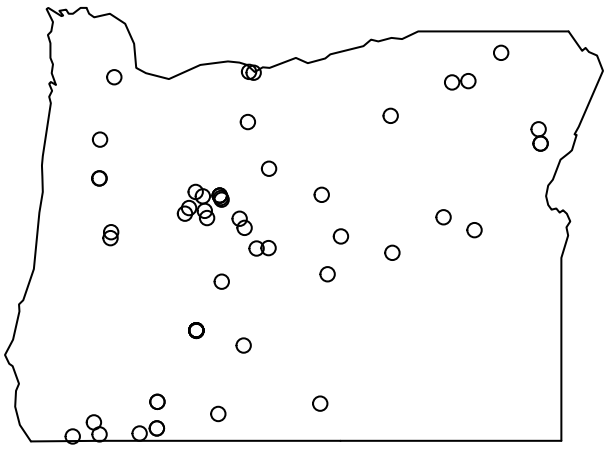
```
library(maptools) # Vector data management (sp)
library(raster) # Raster data management
library(rgdal) # Geospatial Data Abstraction Library
library(rgbif) # One of many GBIF access points (Global Biodiversity Information Facility)
library(maps) # Easy access to basic map layers
library(biomod2) # Ensemble SDM package
```

These are the only packages that we need for our workshop today. However, just like most things in R, there are many ways to do everything, and many packages I am not showing. For example, many people retrieve GBIF (a repository for species occurrence records) data using a package called "dismo," which can also build SDMs. A second example is "sf," which is a package for vector data that gaining traction. At the end of this document, I have listed a selection of spatial R packages and active research directions as a sample of what else is out there for spatial applications. Bottom line, find methods that work for you!

Assemble and organize spatial data

Ponderosa Pine (PIPO) is a dominant tree species across many dry areas of the Pacific Northwest. We might be interested in modeling its current distribution, and predicting that distribution in response to future climate change (see, Mathys et al 2017; Case et al. 2017; Coops et al 2012; Schroeder et al. 2010). An obvious first step is to check GBIF to see what data are available there.

```
gbif.PIPO.OR <- occ_search(scientificName = "Pinus ponderosa",
                           return = "data",
                           hasCoordinate = TRUE,
                           hasGeospatialIssue = FALSE,
                           limit = 200000,
                           country = "US", stateProvince = c("Oregon"),
                           fields = c("name", "decimalLongitude", "decimalLatitude"))

# The data arrive as a data.frame, so convert it to a spatial object
  colnames(gbif.PIPO.OR) <- c("name", "lon", "lat")
  coordinates(gbif.PIPO.OR) <- ~ lon + lat

# If we want to associate data with each of the points in the data.frame
# then we can convert to a SpatialPointsDataFrame
  gbif.PIPO.OR <- SpatialPointsDataFrame(coords = coordinates(gbif.PIPO.OR),
                                         data = data.frame(gbif.PIPO.OR))

# Same as "define projection" in ArcCRAP
  proj4string(gbif.PIPO.OR) <- CRS("+proj=longlat +datum=WGS84
                                   +no_defs +ellps=WGS84 +towgs84=0,0,0")

# Plot a simple map
  map(database = "state", regions = "oregon")
    points(gbif.PIPO.OR)
```

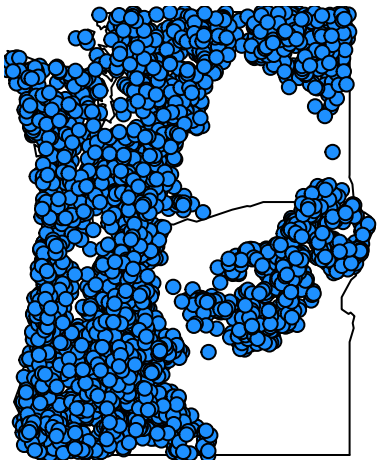


As it turns out, there are not that many data points for PIPO in the GBIF database. So, imagine you contact

a colleague from Washington who sends you a lot more data, but they are in shapefile format. There are many ways to read a shapefile, and a common one is readOGR().

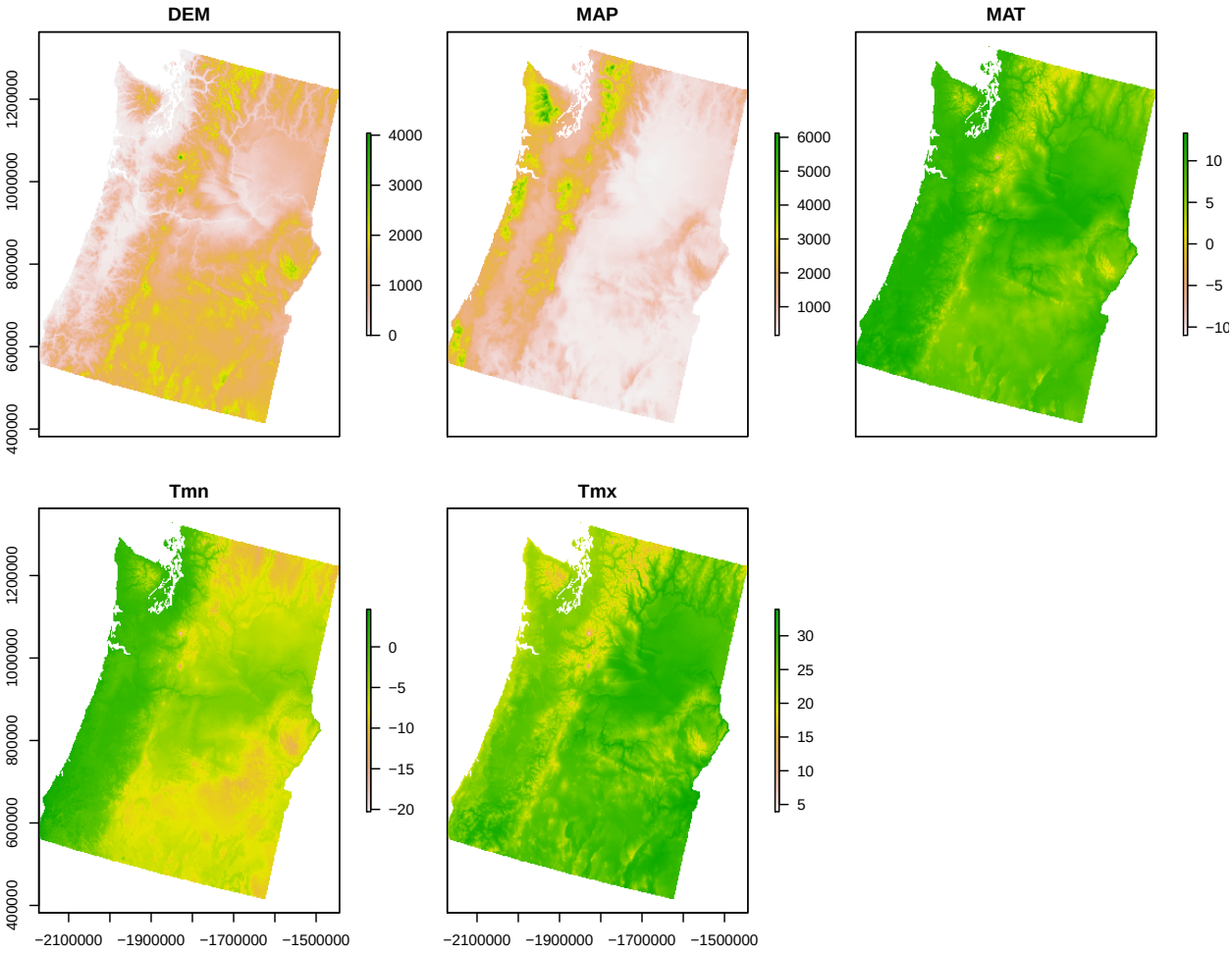
These data are presences and absences in vegetation survey plots collected by the Forest Inventory and Analysis (FIA) program (Bechtold and Patterson 2015).

```
PIPO.FIA.present <- readOGR(dsn = "data/FIA", # dsn is directory
                            layer = "FIA-PIPO-present", # layer is the layer name
                            verbose = FALSE)
PIPO.FIA.absent <- readOGR(dsn = "data/FIA",
                           layer = "FIA-PIPO-absent",
                           verbose = FALSE)

# The data were read in already as SpatialPointsDataFrames, so we could set a field
# within each that represents whether each point is a presence (1) or absence (0)
  PIPO.FIA.present@data <- data.frame(present = rep(1, nrow(PIPO.FIA.present)))
  PIPO.FIA.absent@data <- data.frame(present = rep(0, nrow(PIPO.FIA.absent)))
  gbif.PIPO.OR@data <- data.frame(present = rep(1, nrow(gbif.PIPO.OR)))

# There are many useful base R methods that have been coded up for spatial data
  PIPO.dat <- rbind(PIPO.FIA.present,
                    gbif.PIPO.OR,
                    PIPO.FIA.absent)

# Plot our new dataset for both OR and WA
# Please note that these are ALL of the data, without formatting to indicate presence
  map(database = "state", regions = c("oregon", "washington"))
    points(PIPO.dat, pch = 21, bg = "dodgerblue")
```



Now we have our vector data. We need some environmental data to associate with our occurrence data. In this case, you already have a basic set of environmental data already available from PRISM (http://prism.oregonstate.edu). This includes layers for a DEM, MAT, MAP, Tmin, and Tmax.

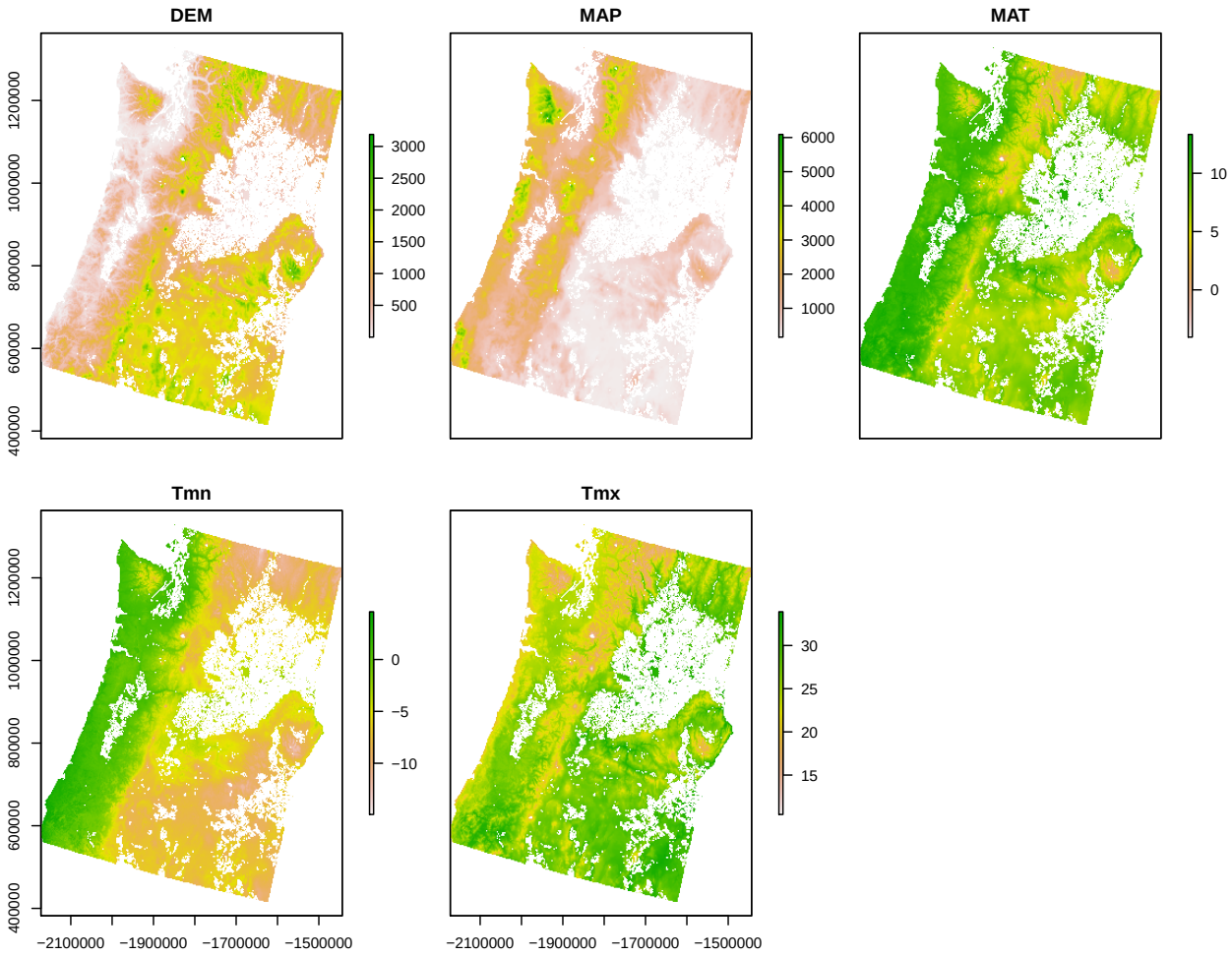
```
# read in PRISM data (I have provided it but you could access the data using the
# "prism" R package)
  env.present <- stack(list.files(path = "data/ENV-Present/",
                                  full.names = TRUE,
                                  pattern = ".tif"))

  plot(env.present)
```



For this analysis we are only interested in forested areas, and we may want to screen, or mask them out (following Mathys et al. 2017).

```
# read a layer to mask out non-forest
  forest.mask <- raster("data/FIA/forestmask.tif")
```

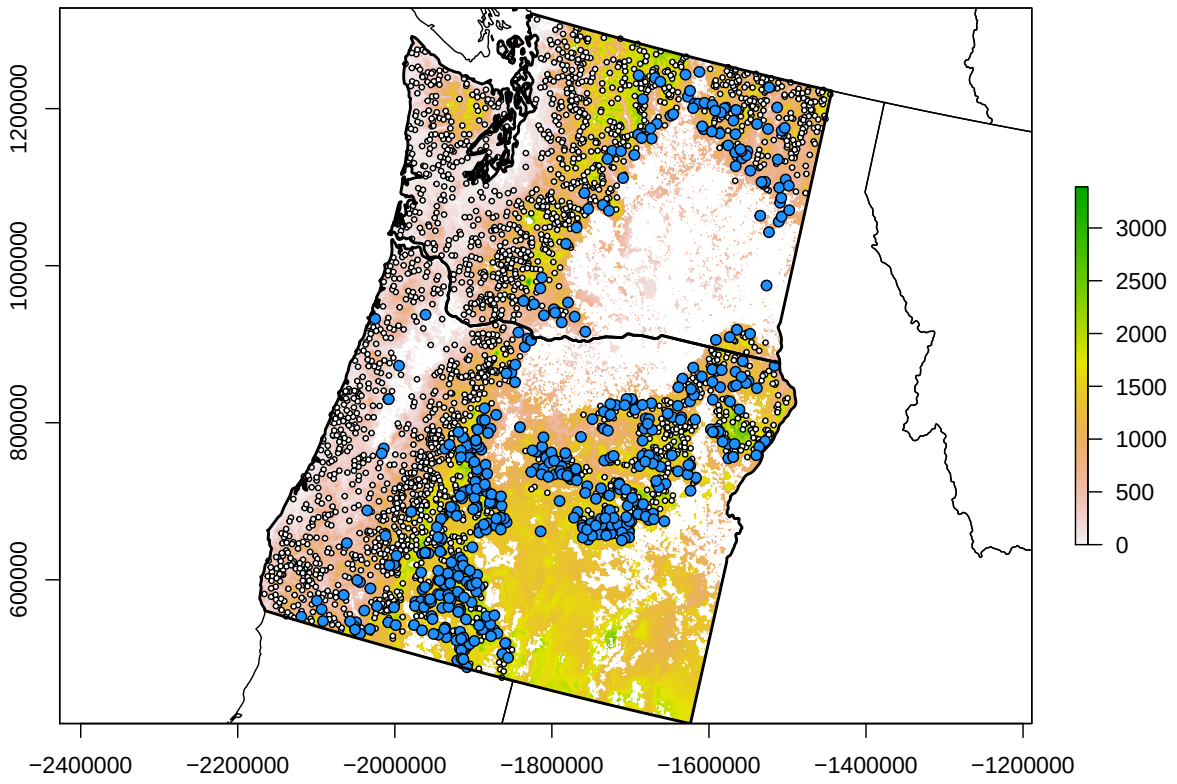
```
  env.present <- mask(x = env.present,
                      mask = forest.mask)

  plot(env.present)
```



Now we have both the species occurrences and the environmental data. But, how can we merge these separate data types? Notice that the projections of the of the raster and vector data are different.

```
# Project the occurrence data to have the projection of the environmental data.
  PIPO.dat <- spTransform(PIPO.dat,
                          proj4string(env.present))

# Now we can plot the vector and raster data together, and adding
# formatting to indicate presences and absences
  plot(env.present$DEM)
    points(PIPO.dat, pch = 21, bg = "white", cex = 0.5)
    points(PIPO.dat[PIPO.dat$present == 1,], pch = 21, bg = "dodgerblue")

# Read in some polygons for state boundaries
```

```
USA <- readOGR(dsn = "data/NA_States_Provinces_Albers",
               layer = "NA_States_Provinces_Albers",
               verbose = FALSE)
ORWA <- readOGR(dsn = "data/ORWA",
                layer = "OR-WA",
                verbose = FALSE)

plot(USA, add = TRUE)
plot(ORWA, add = TRUE, lwd = 2)
```



In order to associate environmental data from our raster with our point species occurrence data, we can extract environmental information to each of the points using extract().

```
env.dat <- extract(x = env.present, y = PIPO.dat)
  head(env.dat, 3)
```

```
##           DEM      MAP      MAT       Tmn      Tmx
## [1,] 1502.661 561.0207 5.601939 -8.532395 25.22251
## [2,] 1434.423 611.8325 6.722266 -7.181122 25.61753
```

```
## [3,] 1614.092 543.4944 5.186131 -9.756168 25.40011
```

You can see how the data are now in a format where you could easily fit a simple model relating presences to these environmental data. However, in our example, we will use biomod2 (Thuiller et al. 2009; Thuiller et al. 2016). This package has functions for formatting the data specifically for this analysis. If you don't have true absence data, this package has several methods for producing "background" data, or pseudo-absences.

Be sure to take care when considering the use of pseudo-absences versus true absences for species distribution modeling. Similarly, it is extremely important to consider the influence of sampling bias in the data used to train models. Further reading: e.g., Guillera-Arroita et al. 2015, Kramer-Schadt et al. 2013, and Merow et al 2013.

```
PIPO.mod.dat <- BIOMOD_FormatingData(resp.var = PIPO.dat,
                                     expl.var = stack(env.present),
                                     #eval.resp.var = ,
                                     #PA.strategy = "random",
                                     #PA.nb.rep = 0, # common practice to resample!
                                     #PA.nb.absences = 0,
                                     resp.name = "Pinus.ponderosa")

PIPO.mod.dat
```

Now, let us fit a subset of the models in biomod2.

- GLM : Generalized Linear Model (glm in base R)
- GAM : Generalized Additive Model (gam from package mgcv)
- ANN: Artificial Neural Network (nnet package)
- RF: Random Forest (randomForest package)
- MAXENT.Tsuruoka: MAXENT (maxent package, but see below for Phillips version)

```
BIOMOD_ModelingOptions() # For Phillips you need to install java/maxent
  # myBiomodOptions <- BIOMOD_ModelingOptions(MAXENT.Phillips =
  #                            list(path_to_maxent.jar = "maxent/maxent.jar"))

PIPO.mod <- BIOMOD_Modeling(data = PIPO.mod.dat,
                            models = c('GLM','GAM','ANN','RF','MAXENT.Tsuruoka'),
                            SaveObj = TRUE,
                            # models.options = myBiomodOptions,
                            # DataSplit = 80, # common practice to validate!
                            VarImport = 1)

PIPO.mod
```

Now we need to assess how well our models fits and inspect the importance of the predictor variables. Many statistics for assessing classification accurracy are available in biomod2 (and many other packages). Three common ones are:

- 'ROC' : Relative Operating Characteristic
- 'KAPPA' : Cohen's Kappa (Heidke skill score)
- 'TSS' : True skill statistic (Hanssen and Kuipers discriminant, Peirce's skill score)

We will focus TSS (see Allouche et al. 2006 for a comparison of all three). TSS is the sum of the rates that we correctly classified presences and absences, minus 1. Higher is better (in the range -1 to 1), and represents a balance between model maximizing sensitivity and specificity.

```
  PIPO.mod.eval <- get_evaluations(PIPO.mod)
    PIPO.mod.eval["TSS","Testing.data",,,]
```

```
##             GLM             GAM             ANN              RF
##           0.652           0.680           0.634           0.984
## MAXENT.Tsuruoka
##           0.639
```

```
    PIPO.mod.eval["KAPPA","Testing.data",,,]
```

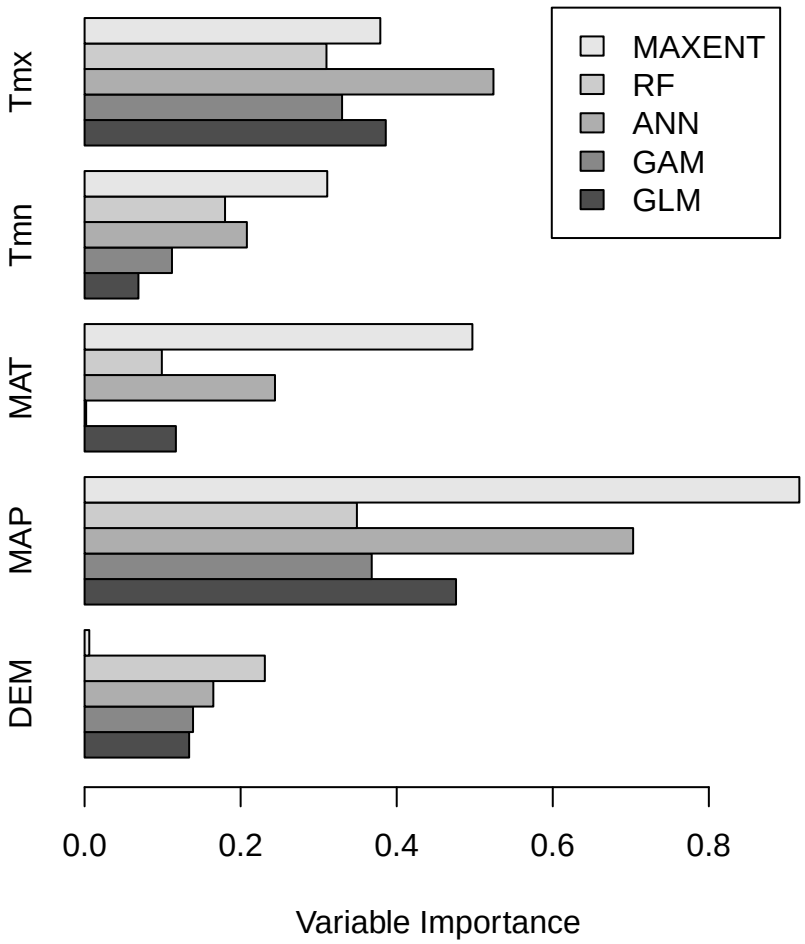
```
##             GLM             GAM             ANN              RF
##           0.536           0.535           0.535           0.971
## MAXENT.Tsuruoka
##           0.501
```

```
    PIPO.mod.eval["ROC","Testing.data",,,]
```

```
##             GLM             GAM             ANN              RF
##           0.899           0.904           0.897           1.000
## MAXENT.Tsuruoka
##           0.887
```

We can also calculate variable importances to compare influences of individual predictor variables within and among models. For a publication, you might also create partial dependence plots. Do the different models agree on the importance of the variables?

```
  var.imp <- drop(get_variables_importance(PIPO.mod))
  barplot(height = t(var.imp),
          beside = TRUE,
          horiz = TRUE,
          xlab = "Variable Importance",
          legend = c("GLM", "GAM", "ANN", "RF", "MAXENT"))
```



One approach for using the information in these various models is to combine them into an ensemble, or collection of models merged together (Thuiller et al. 2009). We can take all models above a given "quality" threshold and combine them.

```
myBiomodEM <- BIOMOD_EnsembleModeling(modeling.output = PIPO.mod,
                                      chosen.models = 'all',
                                      em.by = 'all',
                                      eval.metric = c('TSS'),
                                      eval.metric.quality.threshold = c(0.6),
                                      prob.mean = TRUE,
                                      prob.mean.weight = FALSE,
                                      prob.cv = FALSE,
                                      prob.ci = FALSE,
                                      prob.median = FALSE)
```

```
myBiomodEM
```

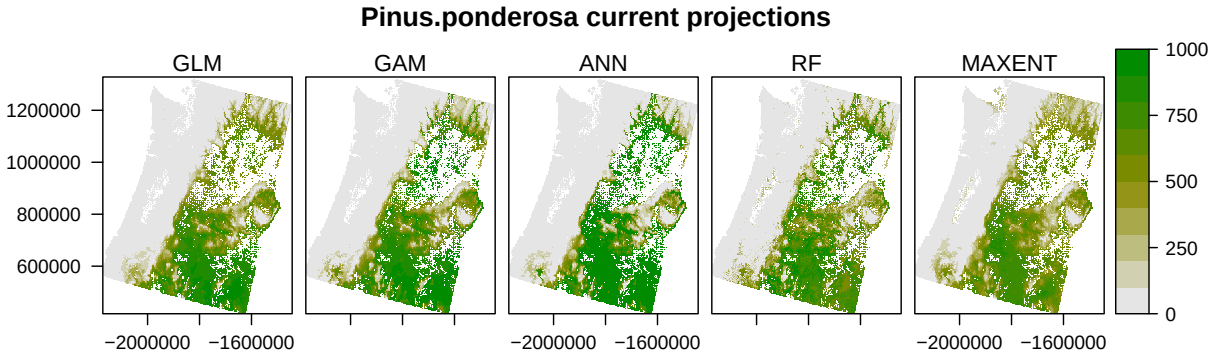
Now we can inspect the accurracy of the ensemble of several models.

```
get_evaluations(myBiomodEM)
```

```
## $Pinus.ponderosa_EMmeanByTSS_mergedAlgo_mergedRun_mergedData
##       Testing.data Cutoff Sensitivity Specificity
## KAPPA        0.629  362.0      72.979      92.774
## TSS          0.759  223.0      90.000      85.709
## ROC          0.947  227.5      89.787      86.110
```

In order to visualize the model outputs, we should now use the SDMs to predict species occurrence across space using our modern environmental data. The models predict probabilities of occurrence, and it is important to remember the interpretation of these outputs depend on the specific model, our data, and our assumptions (e.g., see discussion of Maxent outputs in Merow et al 2013).

```
myBiomodProj <- BIOMOD_Projection(modeling.output = PIPO.mod,
                                  new.env = stack(env.present), # modern environment
                                  proj.name = 'current' ,
                                  selected.models = 'all' ,
                                  binary.meth = 'TSS' ,
                                  compress = 'xz' ,
                                  clamping.mask = F,
                                  output.format = '.grd' )

myBiomodProj

# Just shortening the names so they can be read in the plot
myBiomodProj.plot <- myBiomodProj
myBiomodProj.plot@models.projected <- c("GLM","GAM","ANN","RF","MAXENT")

plot(myBiomodProj.plot)
```



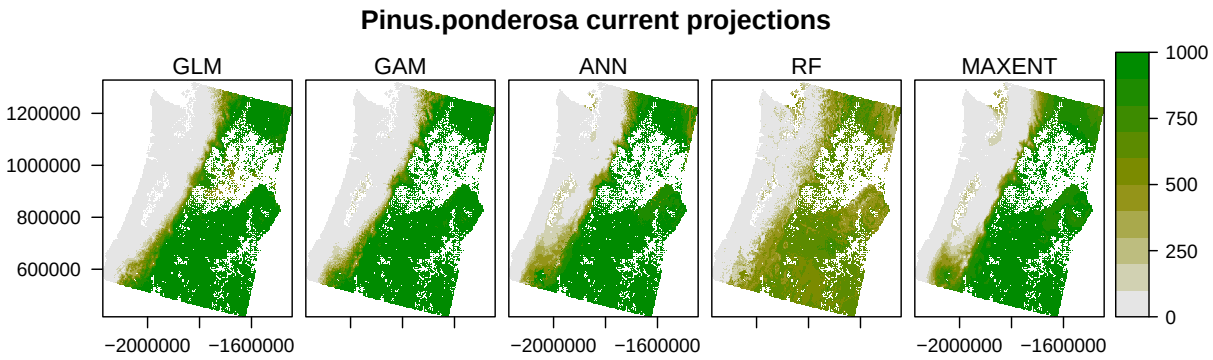
Next, we want to project these species distributions into the future (year 2070, RCP 8.5). In order to to do so we need to assemble environmental data for the future. The "raster" package has some nice functionality for downloading data layers.

```
tmin.future <- getData("CMIP5", download = TRUE,
                       var = "tmin",
                       res = 2.5,
                       rcp = 85,
                       year = 70,
                       model = 'AC')

    pnw.wgs84 <- projectRaster(from = env.present$DEM,
                               crs = "+proj=longlat +datum=WGS84")
    tmin.future <- crop(x = tmin.future, # crops a raster object to a new extent
                        y = extent(pnw.wgs84))
    tmin.future <- min(tmin.future) # The data are in a 12 month stack, so take the minimum
                                    # Note: for more complex operations you can use
                                    #       overlay, stackApply, calc, etc.
    tmin.future <- resample(x = tmin.future, # resample to a new resolution
                            y = pnw.wgs84,
                            method = 'bilinear')
    tmin.future <- mask(x = tmin.future, # You know mask
                        mask = pnw.wgs84)
    tmin.future <- projectRaster(from = tmin.future, # change projection
                                 to = env.present)
    tmin.future <- tmin.future / 10

# The above demonstrates how you might download and process the needed data so that
# it is in the correct format. The data must be in the same format that you used to
# fit the models originally, and have all the same predictors. However, I have
# already prepared the needed data.
  env.future <- stack("data/ENV-Future/env.future.grd")
```

We can now use the same biomod2 projection function that we used on our contemporary climate data in order to make a prediction for occurrence probabilities (~habitat suitability) in future conditions.

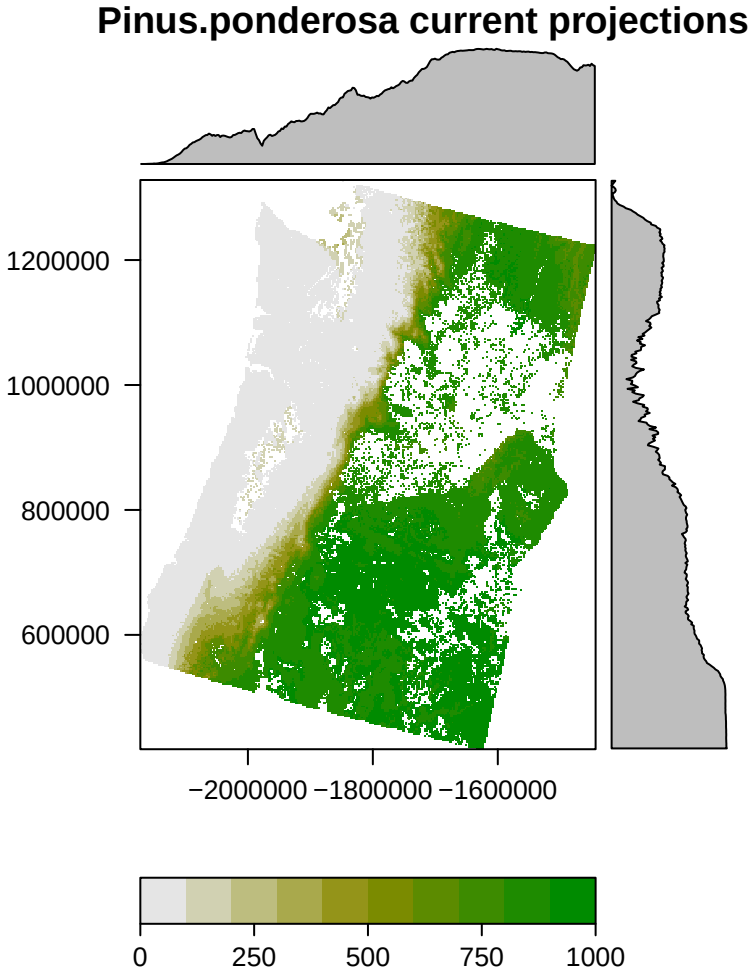
```
myBiomodProj.future <- BIOMOD_Projection(modeling.output = PIPO.mod,
                                         new.env = env.future, # future environment
                                         proj.name = 'current' ,
                                         selected.models = 'all' ,
                                         binary.meth = 'TSS' ,
                                         compress = 'xz' ,
                                         clamping.mask = F,
                                         output.format = '.grd' )

myBiomodProj.future

myBiomodProj.future.plot <- myBiomodProj.future
myBiomodProj.future.plot@models.projected <- c("GLM","GAM","ANN","RF","MAXENT")
```

```
plot(myBiomodProj.future.plot)
```



Finally, to make a single prediction using our ensemble model:

```
myBiomodEF <- BIOMOD_EnsembleForecasting(EM.output = myBiomodEM,
                                         projection.output = myBiomodProj.future)

plot(myBiomodEF)
```



In summary, we have now read in species occurrence data and environmental data, and then used these data to construct a variety of SDMs. We used these models to predict future habitati suitability. Please note that workshop was oriented towards improving spatial data skills in R, and that there are many important nuances to SDM that could not be covered in 2 hrs. For example, it is extremely important to appreciate the limitations of SDM outputs which need to be interpreted within the context of the specific method, sampling biases, response type (e.g., Merow et al. 2013), community, biotic interactions, variable interactions, or other factors/theories (further reading: Guillera-Arroita et al. 2015; Guisan and Rahbek 2011; Kramer-Schadt et al. 2013; Maguire et al. 2015; Thuiller et al. 2013).

Suggested directions

- A particularly active area of research: joint species distribution modeling (JSDM)
- The package used by biomod to create several of their plots: RasterVis
- FedData package: e.g., get weather station data in R
- sf, a new way to manage vector data in R
- Landscape genetics and SDMs (see Gotelli and Staton-Geddes 2015; Razgour 2015)

- ggmap: google map imagery
- rgl or plot_ly for 3d plots
- Many, many more!

Please let me know if you have questions in the future!

Daniel M. Griffith Postdoctoral Research Associate 321 Richardson Hall Forest Ecosystems and Society Oregon State University https://danielmgriffith.wordpress.com/ griffith.dan@gmail.com

Special thanks to Lila Leatherman for helping to prepare and communicate this workshop.

Related References

Allouche, Omri, Asaf Tsoar, and Ronen Kadmon. (2006) Assessing the accuracy of species distribution models: prevalence, kappa and the true skill statistic (TSS). Journal of applied ecology 43, 1223-1232.

Bechtold, W.A. & Patterson, P.L. (2015) The Enhanced Forest Inventory and Analysis Program National Sampling Design and Estimation Procedures.

Case, M.J. & Lawler, J.J. (2017) Integrating mechanistic and empirical model projections to assess climate impacts on tree species distributions in northwestern North America. Global Change Biology, 23, 2005-2015.

Coops, N.C., Waring, R.H., & Hilker, T. (2012) Prediction of soil properties using a process-based forest growth model to match satellite-derived estimates of leaf area index. Remote Sensing of Environment, 126, 160-173.

Gotelli, N.J. & Stanton-Geddes, J. (2015) Climate change, genetic markers and species distribution modelling. Journal of Biogeography, 42, 1577-1585.

Guillera-Arroita, G., Lahoz-Monfort, J.J., Elith, J., Gordon, A., Kujala, H., Lentini, P.E., McCarthy, M.A., Tingley, R., & Wintle, B.A. (2015) Is my species distribution model fit for purpose? Matching data and models to applications: Matching distribution models to applications. Global Ecology and Biogeography, 24, 276-292.

Guisan, A. & Rahbek, C. (2011) SESAM - a new framework integrating macroecological and species distribution models for predicting spatio-temporal patterns of species assemblages: Predicting spatio-temporal patterns of species assemblages. Journal of Biogeography, 38, 1433-1444.

Hijmans, R.J. & van Etten, J. (2012) raster: Geographic analysis and modeling with raster data. R package version, 1, 9-92.

Kramer-Schadt, S., Niedballa, J., Pilgrim, J.D., et al. (2013) The importance of correcting for sampling bias in MaxEnt species distribution models. Diversity and Distributions, 19, 1366-1379.

Maguire, K.C., Nieto-Lugilde, D., Fitzpatrick, M.C., Williams, J.W., & Blois, J.L. (2015) Modeling Species and Community Responses to Past, Present, and Future Episodes of Climatic and Ecological Change. Annual Review of Ecology, Evolution, and Systematics, 46, 343-368.

Mathys, A.S., Coops, N.C., & Waring, R.H. (2017) An ecoregion assessment of projected tree species vulnerabilities in western North America through the 21st century. Global Change Biology, 23, 920-932.

Merow, C., Smith, M.J., & Silander, J.A. (2013) A practical guide to MaxEnt for modeling species' distributions: what it does, and why inputs and settings matter. Ecography, 36, 1058-1069.

Razgour, O. (2015) Beyond species distribution modeling: A landscape genetics approach to investigating range shifts under future climate change. Ecological Informatics, 30, 250-256.

Schroeder, T.A., Hamann, A., Wang, T., & Coops, N.C. (2010) Occurrence and dominance of six Pacific Northwest conifer species. Journal of Vegetation Science, 21, 586-596.

Thuiller, W., Georges, D., Engler, R., Breiner, F., Georges, M.D., & Thuiller, C.W. (2016) Package 'biomod2.'

Thuiller, W., Lafourcade, B., Engler, R., & Araújo, M.B. (2009) BIOMOD - a platform for ensemble forecasting of species distributions. Ecography, 32, 369-373.

Thuiller, W., Münkemüller, T., Lavergne, S., Mouillot, D., Mouquet, N., Schiffers, K., & Gravel, D. (2013) A road map for integrating eco-evolutionary processes into biodiversity models. Ecology Letters, 16, 94-105.