R Assignment 3: Species Delimitation using Morphometrics and Machine Learning

#### Motivation

Add section…

#### Prerequisites

Before starting this week’s tutorial you will have:

1. downloaded the program [Image\_J](https://imagej.nih.gov/ij/download.htm) for making morphometric measurements,
2. downloaded the class google sheet in the file format .csv. Note: if you can’t use imageJ for any reason, don’t panic. Display the digital image of the herbarium sheet on any device and use a ruler held up to the screen to make measurements,
3. **MB**: probably need to get students to do some pre-assignment reading on machine learning (decision trees, random forests). Also, some reading on MDS and interpreting MDS plots.

#### Outcomes

By the end of this tutorial you will:

1. have experience taking morphometric measurements from herbarium specimens,
2. understand some core concepts in machine learning and how it can be used in species delimitation,
3. the tools and knowledge needed to implement machine learning in R as part of a taxonomic research programme,
4. More?

#### Activity 1: Taking Morphometrics Measurements from Herbarium Specimens using ImageJ

ImageJ is an image processing software that was developed by the NIH. It is used to analyze different types of images but this tutorial will cover how to measure certain morphological features from photographs of herbarium specimens.

To use ImageJ you have to import the photographs you want to work with.

1. Drag the file to the lower grey bar. As you are doing this the words <> should appear. Once the image has been dropped it will open and now it is ready to analyze in ImageJ.

There are many measurements that can be taken with ImageJ. It might be helpful to click through the different buttons at the top of the bar to explore all of the different ways ImageJ can make measurements. This tutorial will use the line button for all of the measurements.

1. Zoom in on the ruler (Mac: Command shift =, Windows: CTRL + ). Most digitized herbarium specimens will have a ruler in the photograph. This is to help standardized it and is useful for measuring features in ImageJ.
2. Use the line button (5th button from the left) and draw a line by clicking and dragging along the bottom of the ruler so that the line goes from the 0 tick to the 5 cm tick. This needs to be very accurate because we are going to use this length to calibrate the rest of the measurements taken in ImageJ.
3. Go to the top bar and hit “analyze”. Then go to “set scale…”. Enter 5 into the Known distance box and enter cm into the Unit of Length box. Hit the box “Global”. By checking “Global” you are ensuring that this is applied to all photos in the current session of ImageJ. Then hit OK. ImageJ is now calibrated using the ruler from your herbarium photo.

Now you can go on to take measurements.

1. Use the line or the segmented line (you can change between them by right clicking on the line button) draw lines across what you want to measure. Once you have the right length hit (Mac: Command M, Windows: CTRL M) and ImageJ will measure and record the length in a convenient table for you. This table should pop up once you take your first measurement. When you have all of the measurements you need you can either download that table that ImageJ made or just copy and input the data in your own table. You will have to eventually input your data in a group google sheet so keep that in mind as you are recording data.
2. Once you have taken and recorded all of the measurements necessary put your findings into the class google spreadsheet so that other people will be able to access, download, and use your data for the next step.

Note: [Here is a link](https://imagej.nih.gov/ij/docs/guide/user-guide.pdf) to the ImageJ user guide. All the information you will need is in this document, but if you are interested in the program and the other features available, the user guide can provide more information.

#### Activity 2: Delimiting Species using Morphometrics and Machine Learning

randomForest is an R package that uses machine learning for classification, regression, and clustering. It combines the simplicity of decision trees with flexibility which makes the result more accurate than simply using decision trees alone. There are many tutorials and videos online about randomForest but [this video](https://www.youtube.com/watch?v=J4Wdy0Wc_xQ) is useful if you want more background.

**MB**: maybe this video (or other online learning resources) should be a ‘pre-assignment’?

There are a few ways to use randomForest but this tutorial will cover using the package in unsupervised mode. This means that the program will have to find the structure of the data itself, without us training it first.

1. Open R and set your working directory. You want the working directory to be wherever the data set you are going to input is located.
2. Install the packages needed for randomForest to run properly. These are randomForest, tidyverse, and caret. Installation only needs to be done once so if you have already downloaded any of these (install.packages()), attach the packages using library().
3. By highlighting the script from lines 11-91 and pressing **Ctrl+Enter** (or clicking **Run** at the top of the Source Pane), the functions will execute consecutively, storing objects in the **Environment Pane**.

**MB**: Might need some additional instruction and R code to help with importing the morphometric data. If this tutorial is done on RStudio Cloud, we can load the data in automatically when they open the assignment.

#### Activity 3: Decyphering some code

Let’s step through the R script in order to understand what is happening here.

**Lines 12-15**: The library() function loads various packages designed specifically for machine learning (caret, randomForest), and filtering and plotting data (tidyverse).

**Line 23**: Set the seed for the random number generator. This can be any number.

**MB**: might need to explain the signficance of this step.

**Lines 33-42**: The randomForest() function generates 30000 decision trees (ntree) that attempt to classify the specimens based on binary questions posed to the morphometrics data (columbine\_data). Each question (‘node’) in the decision tree consideres three morphometric variables (mtry). Setting proximity = TRUE tells the function to calculate a similarity metric between specimens. This will come in handy in the following step.

**Line 51**: Create a distance matrix to describing how similar or dissimilar each sample is from one another.

**Line 54**: The cmdscale() function finds the axes of greatest variation in our morphometrics dataset. For example, if corolla width was the trait that varied the most between all specimens, cmdscale() would assign this variable as *MDS axis 1*. In reality, it’s unlikely that a single trait will explain the greatest variation in morphology, instead, MDS axes tend to be combinations of traits.

**MB**: maybe this should be part of pre-assignment reading?

**Lines 57-64**: At this part of the script we will extract information from our MDS object mds.analysis and store this information into various objects. These objects help us to choose which information is most useful to us, and make it easier to create our MDS plot later on.

**Line 57**: Let’s extract the percentage variation described by each of *MDS axis 1* and *MDS axis 2*. Recall from previous tutorials that the $ operator is a way to subset data (in this case mds.analysis) in order to extract information. We’ll round the percent variation to the nearest tenth using round().

**Line 59**: Here, we’re using the $ operator to extract information about where each specimen lies in the morphometric ‘shape space’.

**Lines 61-64**: We’ll create our own spreadsheet of data for plotting using the data.frame() function. Each parameter in this function specifies a column in the spreadsheet (data.frame).

**Lines 62-94**: The ggplot() function takes our mds.data object and plots it. *MDS1* is the axis explaining the most morphometric variation in our herbarium dataset. Notice that this axis separates *A. flavescens* and *A. formosa*.

**MB**: Opportunity here to pose some questions about the ‘V’ distribution of specimens in the MDS plot.

**The result**:

