Seed mass Model Workflow

This analysis was created for a study about what leads to domestication of some grasses but not others. We attempt to see if selection for domestication relied on large seeds in the first place, and which abiotic factors influenced the evolution of seed mass even before human domestication.

In this workflow, we will be taking seedmass data of various grasses from EoL and combine it with data on abiotic factors per species from a previous study by Esther de Regt.

This workflow was created in R version 4.10, and the following packages were used:

package	version
ape	5.5
ggplot2	3.3.5
ggtree	3.0.2
phylobase	0.8.10
phylolm	2.6.2
phylosignal	1.3
raster	3.4-13
sp	0.3.4
tidytree	1.4-5
treeio	1.16.1
usdm	1.1-18

Seed Mass Preprocessing

To start, the trait data gathered from EoL is imported. This data is described more in the README of the scripts directory, and in the shell script used to download the data. For this step, make sure the path leads to the correct location of these data files. In the first code block, please enter the path you have saved the trait-functional-poaceae directory of this project.

```
project_path = '/path/to/project/'
```

For running the analysis on Poales data instead of Poaceae data, comment out the first line, and uncomment the second line of this block.

```
# Import the poaceae seed mass from poaceae_seedmasses.csv.
raw_traits = read.csv(paste(project_path, 'trait-functional-poaceae/data/',
                            'poaceae_seedmasses.csv', sep=''),
                      stringsAsFactors = FALSE)

# Import the poales seed mass from poales_seedmasses.csv.
#raw_traits = read.csv('/trait-functional-poaceae/data/
#                       poales_seedmasses.csv',
```

```
#                       stringsAsFactors = FALSE)
```

Because a few entries are incorrectly saved in EOL, they need to be adjusted. First, in the Poales data, at page ID 50552493, EOL has no organism name stored. Looking up the name on the website of EOL shows it belongs to *Juncus pallescens*. Because this species is already present in the data under another ID (631017) with the same measurement, it was removed. A similar issue occurred at ID 49909676. This species was looked up to be *Lepyrodia cryptica*, which was not in the data yet. Therefore the organism name was manually added. Finally, *Uniola paniculata* was removed because it was a big outlier in this data that does not seem to be correct.

```
# Remove id 50552493 from the raw_traits data frame.
raw_traits = raw_traits[!raw_traits$id==50552493,]

# Change the organism for id 49909676 to 'Lepyrodia cryptica'.
raw_traits[raw_traits$id==49909676, 'organism'] = "Lepyrodia cryptica"

# Remove organism 'Uniola paniculata' from the raw_traits data frame.
raw_traits = raw_traits[!raw_traits$organism=='Uniola paniculata',]
```

Because multiple seed mass measurements can be recorded for one species, these measurements need to be aggregated so that only one measurement is recorded on one species. This was done by averaging all the measurements of each unique species.

```
# Merge the seed mass measurements for species that have multiple measurements
# by taking the average of these measurements.
agg_traits = merge(
             aggregate(list(measurement=raw_traits$measurement), FUN=mean,
                       by=list(id=raw_traits$id)),
                       unique(subset(raw_traits, select=-measurement)))
```

Since the seed mass is log-normally distributed, the natural log of the seed mass measurements were added in a new column.

```
# Add a column log_measurement to the agg_traits data frame to store
# the natural log of the seed mass measurements.
agg_traits$log_measurement = log(agg_traits$measurement)
```

Because this workflow aims to study the relation between domesticated species, wild ancestors, also called crop progenitors, and other wild species, lists were made to store domesticated and progenitor species. Next, subsets of the seed mass data were made. One subset for only domesticated species, one for only crop progenitor species, and one for all wild species including crop progenitors.

```
# Create a list of domesticated species.
domesticated = c(
  'Triticum aestivum', 'Triticum aestivum spelta', 'Triticum turgidum dicoccum',
  'Aegilops speltoides',  'Hordeum vulgare', 'Sorghum bicolor', 'Zea mays',
  'Oryza sativa', 'Oryza glaberrima', 'Secale cereale', 'Eragrostis tef',
  'Digitaria exilis', 'Briza maxima')

# Create a list of crop progenitor species.
progenitors = c('Triticum dicoccoides', 'Hordeum spontaneum',
                'Sorghum arundinaceum', 'Zea mexicana', 'Oryza rufipogon',
                'Oryza barthii', 'Secale montanum', 'Eragrostis pilosa',
                'Digitaria longiflora', 'Briza media')

# Create a data frame dom_traits by taking the entries of the agg_traits
# data frame where the organism name is in the list of domesticated species.
```

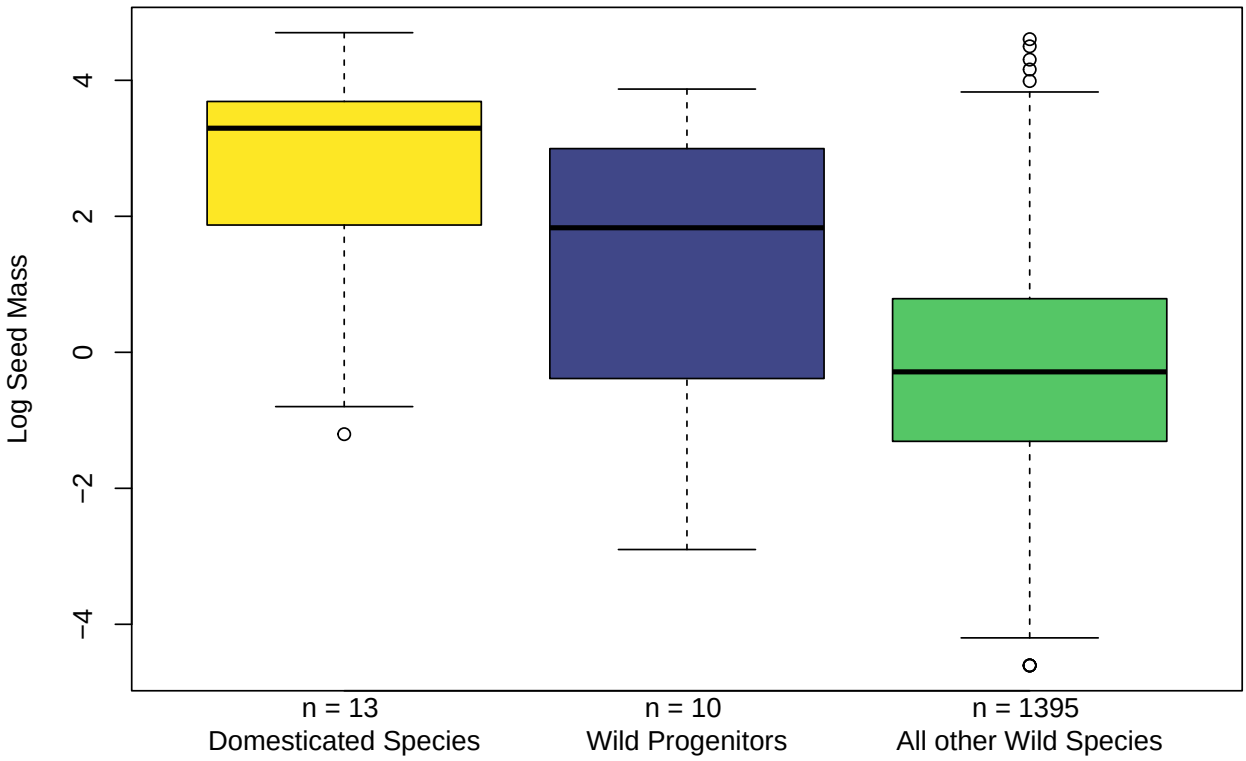
```
dom_traits = agg_traits[agg_traits$organism %in% domesticated,]
# Create a data frame prog_traits by taking the entries of the agg_traits
# data frame where the organism name is in the list of crop progenitor species.
prog_traits = agg_traits[agg_traits$organism %in% progenitors,]
# Create a data frame wild_traits by filtering the domesticated species out of
# the agg_traits data frame.
wild_traits = agg_traits[!agg_traits$organism %in% domesticated,]
```

Selection on seed mass for domestication

The log of the seed mass from the subsets made were put in a boxplot. Note that for this boxplot, the wild species subset was also filtered on the crop progenitors, as not to have these observations in the plot for both wild progenitors and wild species. Because the default x-axis for a boxplot has ticks for each plot and these were interfering with the text labels, the x-axis was turned off in the box plot and later added separately without tick marks, but with the number of observations and the name of the group.

```
# Create a boxplot with the log measurement of domesticated, progenitor,
# and other wild species. "xaxt='n'" disables the x-axis since ticks were
# interfering with the labels.
boxplot(dom_traits$log_measurement, prog_traits$log_measurement,
        wild_traits[!wild_traits$organism %in% progenitors,]$log_measurement, xaxt='n',
        ylab="Log Seed Mass",
        col=c('#FDE725', '#404788', '#55C666'))

# Manually create another x-axis with the number of observations, the name of
# the subsets, and with the ticks disabled.
axis(side = 1, at = 1:3,
     labels = c(paste('n =', nrow(dom_traits), '\nDomesticated Species'),
                paste('n =', nrow(prog_traits), '\nWild Progenitors'),
                paste('n =', nrow(wild_traits[!wild_traits$organism %in% progenitors,]),
                      '\nAll other Wild Species')),
        lwd.ticks = FALSE)
```



Environmental Data Preprocessing

Environmental data about angiosperms were collected by Esther de Regt for a similar study attempting to find a correlation between several angiosperm traits and their environmental variables. This data was imported and merged with the data frame of wild species' seed mass. This merge created a new data frame of the organisms that exist in both the environmental data and the seed mass data. The organism names were changed to replace spaces with underscores for compatability reasons later.

```
layers = read.csv(paste(project_path, 'trait-functional-poaceae/data/',
                        'raw_means.csv', sep=''))
with_layer = merge(wild_traits, layers, by.x='organism', by.y='X')

with_layer$organism = gsub(" ", "_", with_layer$organism)
```

Some of these environmental variables are collinear with eachother. This can cause an unstable regression in the phylogenetic linear model later. For this reason, from the usdm package `vifstep()` was used to find the variable with the highest collinearity and remove it if its VIF was greater than the threshold 10. This function repeats this process until no variables exist with a VIF greater than the threshold any longer. If the usdm package is not yet installed on your system, uncomment the first line in this block to install it.

```
# install.packages('usdm')
library(usdm)

vif = vifstep(with_layer[8:length(with_layer)])@results$Variables
```

Tree Preprocessing

Because the seed mass data is not purely influenced by environmental variables, but also by the species' ancestry, a phylogenetic tree is needed to account for this signal. The treeio package was used to load the tree. If this package is not installed on your system, uncomment the four lines at the start of this codeblock

to install it. Make sure that the path to the tree file is correct. After the tree was important, tips that were not present in the data were dropped.

```
#if (!requireNamespace("BiocManager", quietly = TRUE))
#  install.packages("BiocManager")#BiocManager::install("treeio")
#
#BiocManager::install("treeio")

library(treeio)

# Import the tree
tree = read.tree(paste(project_path, 'trait-functional-poaceae/data/Smith_Brown/v0.1/',
                       'ALLMB.tre', sep=''))

# Prune the tree to only species available in the filtered data.
tree = drop.tip(tree, tree$tip.label[!tree$tip.label %in% with_layer$organism])
```

Because a few of the next steps require the organisms to be in the column names, this was changed in the data frame.

```
# Convert the data frame rownames to be the name of the organism instead.
rownames(with_layer) = with_layer$organism
```

Phylogenetic Signal

To determine the phylogenetic signal in the seed mass data, the phylosignal function `phyloSignal()` was used. To convert the tree to a format that is accepted by this function, the phylobase function `phylo4d()` was used. The phyloSignal function calculates several measures that indicate how much phylogenetic signal is present in a certain dataset following a given tree. The Pagel's lambda variable was chosen for this study. If the phylosignal and phylobase packages are not installed on your system, uncomment the first line of this codeblock to install them.

```
#install.packages('phylosignal')
library(phylosignal)
library(phylobase)

# Convert the treeio tree to a phylo4d object with the log seed mass data.
tree4d = phylo4d(tree, with_layer[, c("log_measurement"), drop=FALSE])
# The phylo4d object can be used by phyloSignal() to calculate phylogenetic signal.
sig = phyloSignal(tree4d)
```

```
# The Pagel's lambda and p-value of is calculation are displayed.
sig$stat$Lambda
```

```
## [1] 0.8456199
```

```
sig$pvalue$Lambda
```

```
## [1] 0.001
```

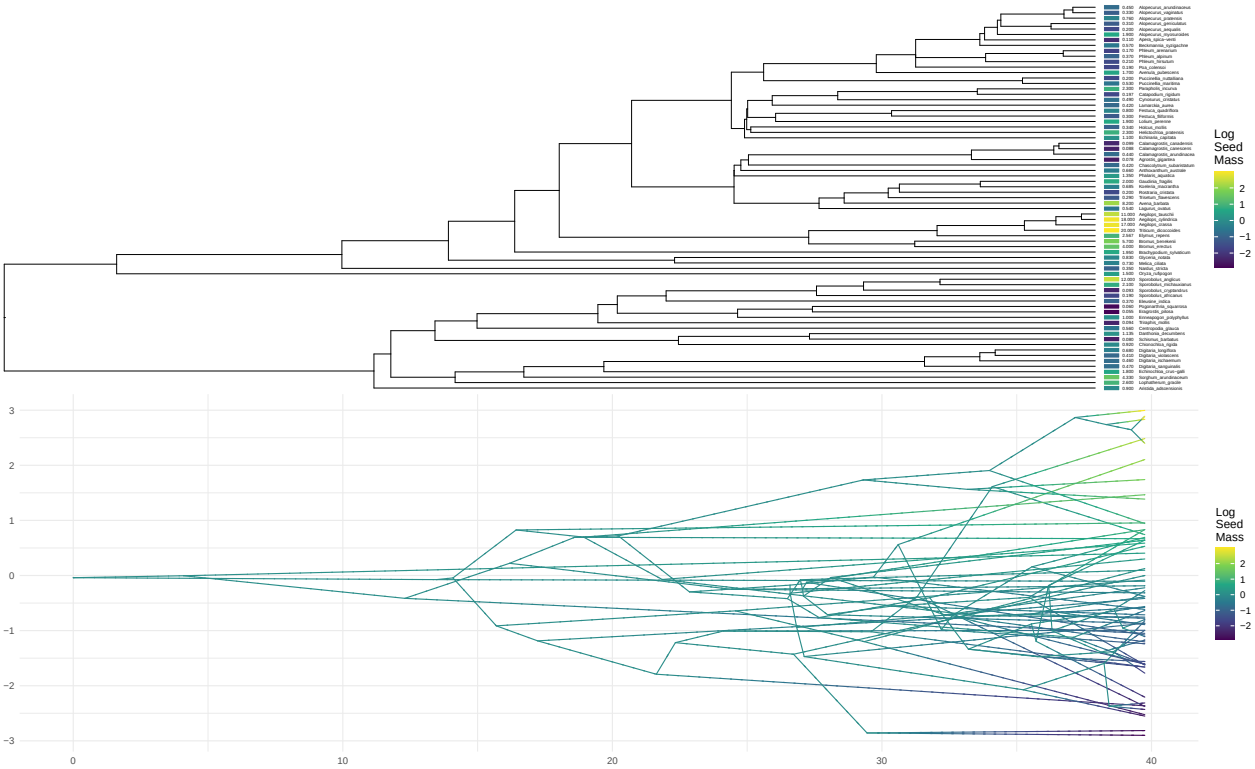
To supplement the Pagel's lambda measure of phylogenetic signal, two trees were also visualised. To do this the packages ggtree and ggplot2 were used. If these are not installed on your system, uncomment the first line of this codeblock to install them. The two figures made here both show the tree and seed mass data in different ways. The first shows the tree dendogram, followed by a one-dimensional heatmap showing the log seed mass. Next to this is the absolute seed mass as a number, followed by the species name these belong to. The second figure shows the tree with the splits relative to each other in time on the horizontal axis, and seed mass on the vertical axis. This shows how each species evolved using estimates of the common ancestor's seed mass.

```
#BiocManager::install("ggtree")
library(ggtree)
library(ggplot2)
```

```
# Join the phylogenetic tree with the data frame containing seed mass.
annotated_tree = full_join(tree, with_layer, by = c("label" = "organism"))

# Create a tree with a heatmap at the tips.
gheatmap(ggtree(annotated_tree, size=0.25) +
           # Add a label at the tips containing the absolute seed mass.
           geom_tiplab(aes(label=format(round(measurement, 3), nsmall=3)),
                       size=1.5, offset=0.75) +
           # Add a label at the tips with the name of the organisms.
           geom_tiplab(aes(label=label), size=1.5, offset=1.4, width=5),
         # Use the log_measurement from the with_layer data frame to create a
         # one-dimensional heatmap.
         with_layer[, c("log_measurement"), drop=FALSE],
         # Define the width of the heatmap and use the viridis colourmap.
         width = 0.015, colnames = FALSE) + scale_fill_viridis_c(name = "Log\nSeed\nMass")

# Create a tree that is coloured according to the seed mass of tips and
# estimated seed mass of internal nodes.
ggtree(annotated_tree, aes(colour=log_measurement), continuous = "colour",
       yscale = "log_measurement", size=0.5) +
         scale_color_viridis_c(name = "Log\nSeed\nMass") +
         theme_minimal()
```



Phylogenetic Linear Model

To create a linear model while keeping in mind the phylogenetic relation of the species, the `phylostep()` function from phylolm was used. If you do not have this package installed, uncomment the first line of this codeblock to install it. The `phylostep` function was used to create a linear model from the VIF selected variables, while keeping the phylogeny in mind. It does so by assuming a model of evolution to compare the tree and seed mass to. The default model is Brownian Motion.

```
#install.packages('phylolm')

library(phylolm)

# Construct a formula using log_measurement, and the VIF selected variables.
formula = as.formula(paste('log_measurement ~ ', paste(vif, collapse='+')))
# Call the phylostep function with the formula, environmental variables, and tree.
lm = phylostep(formula, data=with_layer, phy = tree)
```

Finally, display a summary of the phylogenetic linear model to see the variables and correlation estimates.

```
# Display a summary of the phylogenetic linear model.
summary(lm)
```

```
##
## Call:
## phylolm(formula = create.formula(plm), data = data, phy = phy,
##     model = model, lower.bound = lower.bound, upper.bound = upper.bound,
##     starting.value = starting.value)
##
##    AIC logLik
##  237.3 -112.6
##
## Raw residuals:
##     Min      1Q  Median      3Q     Max
## -2.7554 -0.9955 -0.2998  0.7283  3.7052
##
## Mean tip height: 39.75106
## Parameter estimate(s) using ML:
## sigma2: 0.1224835
##
## Coefficients:
##                      Estimate      StdErr t.value p.value
## (Intercept)        -0.2035054   1.2427576 -0.1638 0.87043
## PETWettestQuarter  -0.0130690   0.0051411 -2.5421 0.01338 *
## Slope             -15.5864569   7.6360632 -2.0412 0.04524 *
## bio15               0.0168504   0.0086868  1.9398 0.05668 .
## bio18               0.0031315   0.0013369  2.3423 0.02219 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.124 Adjusted R-squared: 0.07093
```