Using PCA Scatterplots to Determine Functional Uniqueness for Crop Wild Relatives

Weston Kirk

March 2025

1 Introduction

Crop Wild Relatives (CWR) are the phylogenetically close relatives of common crop species, and represent sources of genetic material for adaptive traits often lost in cultivated species, acting as a safeguard against catastrophic loss [1]. The form and function of crop wild relatives can be measured by functional traits, or quantifiable features of a particular species. In particular, 6 functional traits capture plant form and function, based on conclusions from [2]:

- **ln** – nitrogen content per unit mass (mg/g)
- **sla** – specific leaf area (mm^2/mg)
- **la** – leaf area (mm^2)
- **sm** – seed mass (mg)
- **ph** – plant height (m)
- **ssd** – specific stem density (g/m^3)

2 PCA for Plant Trait Data

The researcher used an existing dataset, `Cleaned_Trait_Data.csv` [1], which contains imputed functional trait measurements for 2790 CWR species. Analysis was done in the R language, using R Studio [2]. Thus, the trait dataset is considered as a matrix $T \in \mathbb{R}^{6 \text{ x } 2790}$. At this point, the researcher de-meaned each row in T to center the data (average of each trait across species), then scaled T using,

$$\frac{1}{\sqrt{n-1}}T = \frac{1}{\sqrt{2789}}T$$

[1] This dataset was from the researchers BSURP Project last summer, where the majority of the non-imputed data came from plant height (**ph**).

[2] Full code can be found in Section 7 at the bottom of the document

Then, the researcher used the `prcomp()` function in the built-in `stats` R library. The `prcomp()` function uses SVD composition, (similar methods to those discussed in lecture 16), to calculate the PCA. The researcher specified the following arguments to the function,

```
pca_result <- prcomp(data_matrix, center = FALSE, scale. = TRUE)
```

specifically, the center was set to false because we already computed the data, and scale was set to true to account for differences in measurement types/units across traits. This is the SVD for the plant trait data:

$$\underbrace{\begin{bmatrix} 0.3834 & 0.4606 & 0.2849 & 0.5939 & -0.3625 & -0.2744 \\ -0.3646 & 0.5217 & -0.4424 & 0.0480 & 0.3854 & -0.4984 \\ 0.4484 & 0.2528 & 0.3661 & -0.1255 & 0.7447 & 0.1755 \\ -0.3039 & 0.5980 & 0.3366 & -0.5178 & -0.3303 & 0.2442 \\ 0.4897 & -0.0914 & -0.0732 & -0.5994 & -0.1891 & -0.5928 \\ 0.4323 & 0.2930 & -0.6859 & -0.0412 & -0.1437 & 0.4842 \end{bmatrix}}_{\mathbf{U}\ (6 \times 6)} \underbrace{\begin{bmatrix} 1.8111 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 1.1861 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0.6835 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0.5993 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0.5654 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0.4086 & \cdots \end{bmatrix}}_{\mathbf{\Sigma}\ (6 \times 2790,\ \text{only top rows shown})} \underbrace{\begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}}_{\mathbf{V}^{\mathbf{T}}\ (2790 \times 2790,\ \text{too large to display})}$$

3 Explained Variance in 2D and 3D PCA-Space

The researcher intended to visualize functional space using a PCA Scatter Plot, but debated whether or not to do this in 2D or 3D PCA space. Once both were plotted, the researcher created a Scree plot (Figure 1) to find the "eblow" to see if this aligned with key thresholds for PCA. Recall, the variance captured by each PC can be denoted as:

$$\text{Variance Captured by PC}_i = \frac{\sigma_i^2}{\sum_{j=1}^{6} \sigma_j^2}$$

Traditionally, capturing anywhere from 70% to 90% of the explained variance are well-established cutoffs [3].

It turns out that the first two PC capture roughly 78.1% of the variance, while the first three PC capture roughly 85.9% of the variance. While the elbow clearly appears with PC 2, the further PC contributions are not negligible, (i.e. they do not plateau significantly), so the researcher plotted in both 2D (Figure 2) and 3D (Figure 3) PCA Space.

4 Relative Ranking and Spearmans Rho

The researcher is looking for the most functionally unique species in 2D and 3D PCA space. To compute this, the researcher used the Euclidian norm for vectors to compute the average distance from all pairwise species, using distance as a proxy for functional uniqueness. This is denoted as the functional uniqueness for each species ($\bar{d}_i$),

$$\bar{f}_i = \frac{1}{2789} \sum_{\substack{j=1 \\ j \neq i}}^{2790} \sqrt{\sum_{k=1}^{d} (s_{ik} - s_{jk})^2}$$

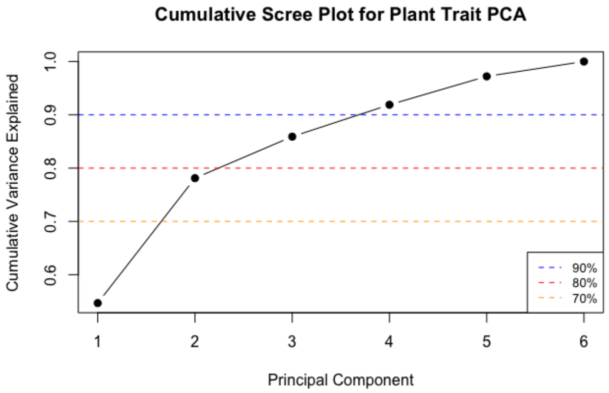

Figure 1: Scree Plot for Plant Trait Data

where $d = 2$ for 2D and $d = 3$ for 3D. From this, a relative functional uniqueness ranking was created for each of the 2790 species. In order to compare the relative rankings of the computations in 2D and 3D PCA space, the researcher used Spearmans Rho (ρ) to measure,

$$\rho = 1 - \frac{6 \sum d_i^2}{2790((2790)^2 - 1)}$$

where d_i represents the difference in species i ranking between each space, (i.e. if species i is ranked 3rd in 2D PCA space, and 7th in 3D PCA space, then $d_i = 4$). For these two datasets, Spearmans Rho (ρ) was 0.954, meaning that these 2 measures of trait space are well aligned.

5 Functionally Unique Species

Adjacent literature looking examined global protected areas in relation to the in-situ conservation of crop wild relatives, namely, creating a "top 10" list of clusters where CWR live inside and outside of protected areas [4]. The researcher took inspiration from this approach on a species-level, presenting the top 10 most "functionally unique" species, seen in Table 1.
From this analysis, it is clear that functional uniqueness occurs across a variety of conservation statuses. It is also true that 4 out of the 10 are Data Deficient (DD), which signifies that there is simply no data available. This means that we are completely unsure of their conservation status by IUCN criteria, but in reality, could be Extinct in the Wild or Least Concern. Identifying DD species represents an active area of research [5], and could serve as a next step for this work. That being said, a high functional uniqueness, as demonstrated here, provides conservation justification for the DD species listed above. Namely, the

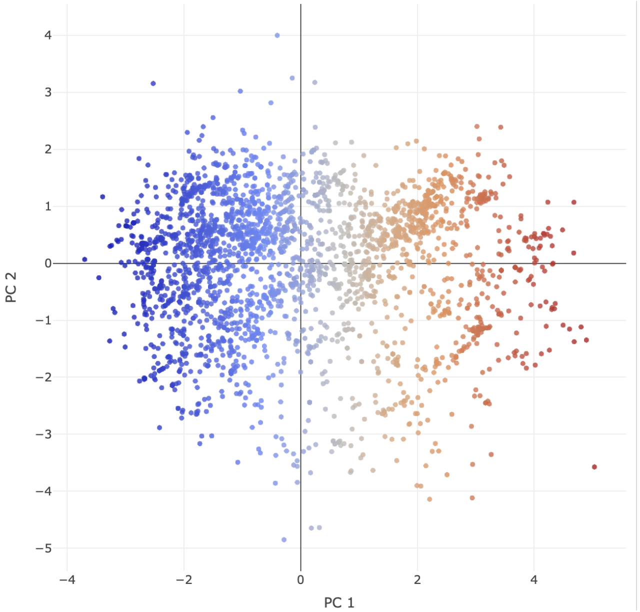

Figure 2: 2D PCA Scatterplot

isolation of *Prunus cerasifera* is notable, being by far the most ”functionally unique” species. For reference, the isolated dark red point in the top left of the 3D PCA Scatterplot is *Prunus cerasifera*. Furthermore, 3 out of the 10 species the researcher considered are Vulnerable or Endangered, including *Manilkara huberi*[3], the only Endangered species in the top 10. These three species not only are at-risk by traditional conservation measures, but if they were to go extinct, a unique niche in functional space would be lost, one which would be less likely to be filled by an adjacent species. This emphasizes the notion that functional space should be viewed as a conservation measure–as it represents a different way to quantify biodiversity.

[3] Here is an interesting excerpt from Wikipedia about the use of *Manilkara huberi* latex in golf balls, one of the many uses of CWR: ”The latex from M. huberi was used to make golf ball covers, along with that the better known and more widespread M. bidentata (”Balata” or ”Massaranduba.”) Latex products from a number of Manilkara species being interchangeably called ”balata”, ”Gutta Balata”, or more generally ”Chicle.” It was considered a good quality, but short-lived, cover, requiring frequent recoating or replacement. Balls with Balata coatings had a high spin rate. Yet it was popular in tournaments among professionals and low handicap players. Modern materials such as Polyurethane Elastomer and Methacrylic Acid copolymers have made Balata golf balls largely obsolete by the late 20th century, as they have much better abrasion resistance and generally lower air drag.”

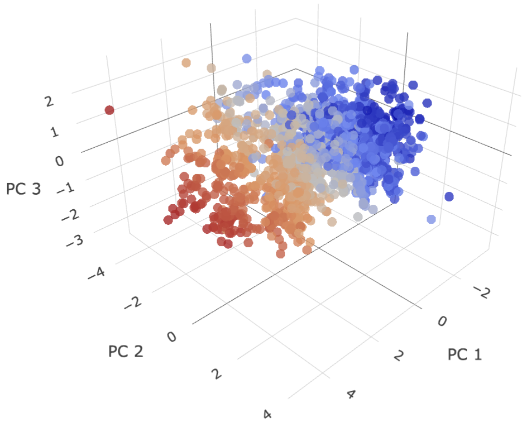

Figure 3: 3D PCA Scatterplot

Species	**Avg. Distance**	**Rank**	**IUCN Status**
Prunus cerasifera	6.689872	1	**Data Deficient**
Pinus heldreichii	5.465169	2	**Least Concern**
Manilkara huberi	5.380211	3	**Endangered**
Cyperus articulatus	5.359989	4	**Least Concern**
Ambrosia deltoidea	5.292286	5	**Data Deficient**
Vaccinium myrsinites	5.249545	6	**Data Deficient**
Madhuca hainanensis	5.201356	7	**Vulnerable**
Manilkara bidentata	5.160287	8	**Vulnerable**
Pinus densiflora	5.090327	9	**Least Concern**
Atriplex lampa	5.076923	10	**Data Deficient**

Table 1: Top 10 Species Ranked by Average Distance

6 Use of LLMs

I used LLMs to assist in generating some of the code due to complex syntax in R. LLMs were used for translation purposes, not conceptual reasoning.

7 R Code

```
library(readr)
library(stats)
library(dplyr)
library(tibble)
library(viridis)
```

Figure 4: *Manilkara huberi*, a large tree native to South and Central America used for its edible fruit, lumber, and most notably latex. Fev, Public domain, via Wikimedia Commons.

```
library(plotly)

# Load in cleaned trait data from previous analysis
file_path <- "Cleaned_Trait_Data.csv"
traits_data <- read_csv(file_path)

# Removing extraneous value from the sm column
traits_data <- traits_data %>%
  mutate(sm = as.numeric(sub(" .*", "", sm)))

# Transpose the data in loaded-in format
# This is done to ensure A is m x n, where:
# we have n samples (species in this case, where n = 2790)
# and m measurements (functional traits in this case, where n = 6)
T <- as.data.frame(t(as.matrix(traits_data)))
view(T)

# Fixing a formatting issue, move species names to top row
colnames(T) <- T[1, ]
T <- T[-1, ]
view(T)

# Convert to numeric data
T_numeric <- as.matrix(T)
mode(T_numeric) <- "numeric"

# Row means computation for de-meaning
row_avg <- rowMeans(T_numeric, na.rm = TRUE)
print("\nRow Averages:")
print(row_avg)

# Subtract respective row means
# (these are averaged functional trait measurements across all species)
T_centered <- T_numeric - row_avg

# Convert to data frame
T_centered_df <- as.data.frame(T_centered)
view(T_centered_df)

# Assign de-meaned data back to T
T <- T_centered_df
view(T)

# A is an m x n matrix, so n is number of rows (number of species)
n <- ncol(T_centered)
print(n)

# Scale T by 1/sqrt(n - 1) to match SVD
T_scaled <- T_centered / sqrt(n - 1)

# Convert back to a data frame
T_scaled_df <- as.data.frame(T_scaled)
view(T_scaled_df)

# Assign scaled data to T before transposing
T <- T_scaled_df
view(T)

# Transpose T
```

```
T_transpose <- as.data.frame(t(as.matrix(T)))
view(T_transpose)

# Convert T and T^T to numeric matrices for computation
T_numeric <- as.matrix(T)
mode(T_numeric) <- "numeric"

T_transpose_numeric <- as.matrix(T_transpose)
mode(T_transpose_numeric) <- "numeric"

# Ensure correct input orientation: species as rows, traits as columns
data_matrix <- as.data.frame(t(T_numeric))

# PCA using built in prcomp() function
pca_result <- prcomp(data_matrix, center = FALSE, scale. = TRUE)

# Examining SVD components from prcomp()
U <- pca_result$rotation

# To visualize singular values along diagonal
Sigma_values <- pca_result$sdev
Sigma <- diag(Sigma_values)

# Issue displaying with species names, dont print
Vt <- t(pca_result$x)  # V^T (Right Singular Vectors - Principal Component Scores Transposed

# Print individual SVD matrices
cat("\nU (Left Singular Vectors - Principal Component Directions):\n")
print(U)

cat("\n (Singular Value Matrix - Diagonal):\n")
print(Sigma)

# Extract 2D PCA scores (PC1 vs PC2)
pca_scores_2D <- as.data.frame(pca_result$x[, 1:2])
pca_scores_2D$species <- rownames(pca_scores_2D)

# 3D PCA scores for species
pca_scores <- as.data.frame(pca_result$x[, 1:3])
pca_scores$species <- rownames(pca_scores)

# 2D PCA Scatter Plot
p2 <- plot_ly(pca_scores_2D,
              x = ~PC1,
              y = ~PC2,
              text = ~paste(species, "<br>PC1:", round(PC1, 2), "<br>PC2:", round(PC2, 2)),
              type = "scatter",
              mode = "markers",
              marker = list(size = 5, opacity = 0.8, color = ~PC1, colorscale = "RdBu"),
              hoverinfo = "text",
              showlegend = FALSE) %>%
  layout(title = "2D PCA Scatter Plot for Plant Trait Data",
         xaxis = list(title = "PC 1"),
         yaxis = list(title = "PC 2"))


# This line actually renders the plot in the plot window
p2

# 3D PCA Scatter Plot
p3 <- plot_ly(pca_scores,
              x = ~PC1,
              y = ~PC2,
              z = ~PC3,
              text = ~paste(species,
                            "<br>PC1:", round(PC1, 2),
                            "<br>PC2:", round(PC2, 2),
                            "<br>PC3:", round(PC3, 2)),
              type = "scatter3d",
              mode = "markers",
              marker = list(size = 5,
                            opacity = 0.8,
                            color = ~PC1,
                            colorscale = "RdBu"),
              hoverinfo = "text",
              showlegend = FALSE) %>%
  layout(title = "3D PCA Scatter Plot for Plant Trait Data",
         scene = list(
           xaxis = list(title = "PC 1"),
           yaxis = list(title = "PC 2"),
           zaxis = list(title = "PC 3")
         ))


p3


# Singular values from sigma
Sigma_values <- diag(Sigma)

# Use Sigma_values to compute eigenvalues and explain variance from each PC
explained_variance <- (Sigma_values^2) / sum(Sigma_values^2)
```

```
# Print above result
cat("\nVariance Explained by Each Principal Component:\n")
print(explained_variance)

# How much each PC contributes to overall variance as we add more up to all 6
cumulative_variance <- cumsum(explained_variance)

# Print above result
cat("\nCumulative Variance Explained:\n")
print(cumulative_variance)

# Scree plot to visualize cumulative_variance relationship
plot(cumulative_variance, type = "b",
     main = "Cumulative Scree Plot for Plant Trait PCA",
     xlab = "Principal Component",
     ylab = "Cumulative Variance Explained",
     col = "black", pch = 19, lty = 1)

# Threshold lines at 70%, 80%, and 90%
abline(h = 0.7, col = "orange", lty = 2)  # 70% line
abline(h = 0.8, col = "red", lty = 2)  # 80% line
abline(h = 0.9, col = "blue", lty = 2)  # 90% line

# Legend
legend("bottomright", legend = c("90%", "80%", "70%"),
       col = c("blue", "red", "orange"), lty = 2, cex = 0.8)

# Average Euclidean distance for each species in both 2D and 3D
compute_avg_distance_vectorized <- function(matrix) {
  # Capture all 2790 species
  n <- nrow(matrix)

  # Pairwise differences for each coordinate
  x_diff <- outer(matrix[,1], matrix[,1], "-")  # x_i - x_j
  y_diff <- outer(matrix[,2], matrix[,2], "-")  # y_i - y_j

  if (ncol(matrix) == 3) {
    z_diff <- outer(matrix[,3], matrix[,3], "-")  # z_i - z_j (for 3D)
    # Store in a distance matrix, where the rows are pairwise distances for each species
    dist_matrix <- sqrt(x_diff^2 + y_diff^2 + z_diff^2)
  } else {
    dist_matrix <- sqrt(x_diff^2 + y_diff^2)
  }

  diag(dist_matrix) <- NA  # when i = j, distance from a coordinate to itself irrelevant
  avg_distances <- rowMeans(dist_matrix, na.rm = TRUE)  # Compute average per row

  return(avg_distances)
}

# Average distance for 2D PCA call w/matrix conversion
coords_2D <- as.matrix(pca_scores_2D[, c("PC1", "PC2")])
pca_scores_2D$avg_distance <- compute_avg_distance_vectorized(coords_2D)
# Higher avgerage distance means higher rank,
# as we are looking for the most functionally unique species
pca_scores_2D$rank <- rank(-pca_scores_2D$avg_distance)

# Same as above, but for 3D PCA
coords_3D <- as.matrix(pca_scores[, c("PC1", "PC2", "PC3")])  # Convert to matrix
pca_scores$avg_distance <- compute_avg_distance_vectorized(coords_3D)
pca_scores$rank <- rank(-pca_scores$avg_distance)  # Higher avg dist → Higher rank

# Print above result
cat("\nTop 10 Species in 2D PCA by Avg Distance:\n")
print(head(pca_scores_2D[order(pca_scores_2D$rank), c("species", "avg_distance", "rank")], 1

cat("\nTop 10 Species in 3D PCA by Avg Distance:\n")
print(head(pca_scores[order(pca_scores$rank), c("species", "avg_distance", "rank")], 10))

# Spearmans rank correlation between 2D and 3D PCA relative rankings
spearman_rho <- cor(pca_scores_2D$rank, pca_scores$rank, method = "spearman")

# Print above result
cat("\nSpearmans rho:", spearman_rho, "\n")
```

References

[1] Emily Warschefsky, R Varma Penmetsa, Douglas R Cook, and Eric JB Von Wettberg. Back to the wilds: tapping evolutionary adaptations for resilient crops through systematic hybridization with crop wild relatives. *American journal of botany*, 101(10):1791–1800, 2014.

[2] Carlos P Carmona, Riin Tamme, Meelis Pärtel, Francesco de Bello, Sébastien Brosse, Pol Capdevila, Roy González-M, Manuela González-Suárez, Roberto Salguero-Gómez, Maribel Vásquez-Valderrama, et al. Erosion of global functional diversity across the tree of life. *Science advances*, 7(13):eabf2675, 2021.

[3] Anshul Verma, Pierpaolo Vivo, and Tiziana Di Matteo. A memory-based method to select the number of relevant components in principal compo-

nent analysis. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(9):093408, 2019.

[4] Holly Vincent, Ahmed Amri, Nora P Castañeda-Álvarez, Hannes Dempewolf, Ehsan Dulloo, Luigi Guarino, David Hole, Chikelu Mba, Alvaro Toledo, and Nigel Maxted. Modeling of crop wild relative species identifies areas globally for in situ conservation. *Communications biology*, 2(1):136, 2019.

[5] Victor Cazalis, Luca Santini, Pablo M Lucas, Manuela González-Suárez, Michael Hoffmann, Ana Benítez-López, Michela Pacifici, Aafke M Schipper, Monika Böhm, Alexander Zizka, et al. Prioritizing the reassessment of data-deficient species on the iucn red list. *Conservation Biology*, 37(6):e14139, 2023.