Color Analysis of Floral Tissues

K. Uckele

2025-02-06

Contents

Introduction

This analysis explores the color properties of different floral tissues — bract, petal, and labellum — using spectral data sourced from Google Sheets. The analysis includes data loading, quality control, spectral processing, visualization, and the computation of summary descriptors.

Setup

Load Required Libraries

```
## libraries
library(googlesheets4)
library(cowplot)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tibble)
library(knitr)
library(rmarkdown)
library(pavo)
```

Set working directory

```
# Set the root directory to the project root
knitr::opts_knit$set(root.dir = normalizePath("../"))
```

Define Custom Operators

```
# Define a 'not in' operator
`%notin%` = Negate(`%in%`)
```

Authenticate Google Sheets Access

Authenticate access to Google Sheets using your email. Ensure that the email has the necessary permissions to access the sheets.

```
# Authenticate with Google Sheets
gs4_auth(email = "kuckele@ucsc.edu")
```

Quality Control

Load Spectral Data from Google Sheets

We load the spectral data for bract, petal, and labellum from the specified Google Sheets.

```
# Specify the Google Sheets ID
sheet_id <- "12NWtlqKbLPAxU-rVBTqMQjY_6jH1DulVvNHS_QXAvFI"

# Load data for different floral tissues
bract <- read_sheet(ss = sheet_id, sheet = "Bract")
```

```
## v Reading from "Color".
```

```
## v Range ''Bract''.
```

```
petal <- read_sheet(ss = sheet_id, sheet = "Petal")
```

```
## v Reading from "Color".
```

```
## v Range ''Petal''.
```

```
labellum <- read_sheet(ss = sheet_id, sheet = "Labellum")
```

```
## v Reading from "Color".
```

```
## v Range ''Labellum''.
```

Full Spectrum Analyses

Spectral Data Processing

Convert Data to rspec Objects

The spectral data is converted into rspec objects using the pavo package for further analysis.

```
# Set seed for reproducibility
set.seed(1612217)

# Convert datasets to rspec objects with wavelength limits
bract_spec <- as.rspec(bract, lim = c(300, 700), whichwl = 1)
```

```
## The spectral data contain 555 negative value(s),
##  which may produce unexpected results if used in models.
## Consider using procspec() to correct them.
```

```
petal_spec <- as.rspec(petal, lim = c(300, 700), whichwl = 1)
```

```
## The spectral data contain 272 negative value(s),
##  which may produce unexpected results if used in models.
## Consider using procspec() to correct them.
```

```
labellum_spec <- as.rspec(labellum, lim = c(300, 700), whichwl = 1)

# change column names
colnames(bract_spec) <- gsub("-", "x", colnames(bract_spec))
colnames(petal_spec) <- gsub("-", "x", colnames(petal_spec))
colnames(labellum_spec) <- gsub("-", "x", colnames(labellum_spec))
```

Note: The conversion may produce warnings about negative values in spectral data, which are addressed in subsequent steps.

Average the Spectra

Aggregate the spectral data by sample names, averaging replicates.

```
# Extract sample names by removing trailing numbers in parentheses
bract_samples <- gsub("\\([0-9]+\\)$", "", names(bract_spec))[-1]
petal_samples <- gsub("\\([0-9]+\\)$", "", names(petal_spec))[-1]
labellum_samples <- gsub("\\([0-9]+\\)$", "", names(labellum_spec))[-1]

# Verify sample counts
table(bract_samples)
```

```
## bract_samples
##    125    126     39  39x10 39x109 39x110 39x115 39x116 39x117  39x12 39x122
##      2      3      1      2      1      2      2      1      3      2      2
## 39x123 39x125 39x126  39x13 39x130 39x136  39x14  39x15  39x16  39x17   39x2
##      2      2      2      1      3      2      3      2      4      2      2
##  39x21  39x23  39x25  39x27  39x34  39x39   39x4  39x40  39x41  39x44  39x46
##      2      2      2      2      2      2      2      2      2      2      2
##  39x49  39x50  39x51  39x55  39x56  39x57   39x6  39x60  39x65  39x67  39x68
##      2      2      2      2      2      2      2      2      2      2      1
##  39x75  39x77  39x78  39x79   39x8  39x81  39x82  39x86  39x87  39x89  39x92
##      2      2      2      2      1      2      3      1      1      2      2
##  39x93  39x95  39x96  39x98  62x10 62x103 62x105 62x109 62x116 62x119 62x122
```

```
##      2      2      2      2      1      2      2      1      1      2      1
## 62x125 62x128 62x129  62x13 62x130 62x131 62x134 62x135 62x136 62x137 62x138
##      1      1      2      2      1      2      1      2      2      1      1
## 62x139  62x14 62x140 62x143 62x144 62x147 62x151 62x152 62x153 62x154 62x157
##      1      1      2      1      2      1      1      3      2      2      2
## 62x158  62x16 62x162 62x167 62x168 62x169  62x17 62x172 62x173 62x175  62x18
##      2      2      2      2      1      1      2      2      2      2      1
## 62x182 62x185 62x188 62x189  62x19 62x190 62x197 62x200 62x206  62x21 62x214
##      1      2      1      1      2      2      2      2      2      2      1
## 62x218 62x219  62x22 62x220 62x225 62x233 62x234 62x240 62x251 62x252 62x258
##      2      2      2      1      2      1      1      2      2      1      1
## 62x261 62x264 62x265 62x267 62x268 62x276 62x282 62x293 62x296 62x300 62x302
##      2      2      1      2      2      1      1      2      1      1      1
## 62x303 62x304 62x305 62x306 62x307 62x308 62x309 62x310 62x313 62x315 62x322
##      2      1      1      1      2      1      2      1      1      1      1
## 62x324 62x327 62x332 62x337 62x338 62x351 62x355 62x359 62x363 62x370   62x5
##      1      1      2      1      2      1      2      2      2      1      1
##  62x57  62x58  62x59  62x60  62x63  62x65  62x66  62x69  62x73  62x74  62x75
##      2      2      2      1      1      1      1      2      2      1      1
##  62x77  62x83  62x87  62x91  62x92  62x94  62x96   BRAC
##      1      2      1      2      2      1      1      2
```

```
table(petal_samples)
```

```
## petal_samples
##    125    126     39  39x10 39x109 39x110 39x115 39x116 39x117  39x12 39x122
##      2      2      1      2      1      2      2      1      3      2      2
## 39x123 39x125 39x126  39x13 39x130 39x136  39x14  39x15  39x16  39x17   39x2
##      2      2      2      1      3      2      3      2      4      2      2
##  39x21  39x23  39x25  39x27  39x34  39x39   39x4  39x40  39x41  39x44  39x46
##      2      2      2      2      2      2      2      2      2      2      2
##  39x49  39x50  39x51  39x55  39x56  39x57   39x6  39x60  39x65  39x67  39x68
##      2      2      2      2      2      2      2      2      2      2      1
##  39x75  39x77  39x78  39x79   39x8  39x81  39x82  39x86  39x87  39x89  39x92
##      2      2      2      2      1      2      3      1      1      2      2
##  39x93  39x95  39x96  39x98  62x10 62x103 62x105 62x109 62x116 62x119 62x122
##      2      2      2      2      1      2      2      1      1      2      1
## 62x125 62x128 62x129  62x13 62x130 62x131 62x134 62x135 62x136 62x137 62x138
##      1      1      2      2      1      2      1      2      2      1      1
## 62x139  62x14 62x140 62x143 62x144 62x147 62x151 62x152 62x153 62x154 62x157
##      1      1      2      1      2      1      1      3      2      2      2
## 62x158  62x16 62x162 62x167 62x168 62x169  62x17 62x172 62x173 62x175  62x18
##      2      2      2      2      1      1      2      2      2      2      1
## 62x182 62x185 62x188 62x189  62x19 62x190 62x197 62x200 62x206  62x21 62x214
##      1      2      1      1      2      2      2      2      2      2      1
## 62x218 62x219  62x22 62x220 62x225 62x233 62x234 62x240 62x251 62x252 62x258
##      2      2      2      1      2      1      1      2      2      1      1
## 62x261 62x262 62x264 62x265 62x268 62x276 62x282 62x293 62x296 62x300 62x302
##      2      2      2      1      2      1      1      2      1      1      1
## 62x303 62x304 62x305 62x306 62x307 62x308 62x309 62x310 62x313 62x315 62x322
##      2      1      1      1      2      1      2      1      1      1      1
## 62x324 62x327 62x332 62x337 62x338 62x351 62x355 62x359 62x363 62x370   62x5
##      1      1      2      1      2      1      2      2      2      1      1
##  62x57  62x58  62x59  62x60  62x63  62x65  62x66  62x69  62x73  62x74  62x75
##      2      2      2      1      1      1      1      2      2      1      1
```

```
##  62x77  62x83  62x87  62x91  62x92  62x94  62x96   BRAC
##      1      2      1      2      2      1      1      2
```

```
table(labellum_samples)
```

```
## labellum_samples
##    125    126     39  39x10 39x109 39x110 39x115 39x116 39x117  39x12 39x122
##      2      2      1      2      1      2      2      1      3      2      2
## 39x123 39x125 39x126  39x13 39x130 39x136  39x14  39x15  39x16  39x17   39x2
##      2      2      2      1      3      2      3      2      4      2      2
##  39x21  39x23  39x25  39x27  39x34  39x39   39x4  39x40  39x41  39x44  39x46
##      2      2      2      2      2      2      2      2      2      2      2
##  39x49  39x50  39x51  39x55  39x56  39x57   39x6  39x60  39x65  39x67  39x68
##      2      2      2      2      2      2      2      2      2      2      1
##  39x75  39x77  39x78  39x79   39x8  39x81  39x82  39x86  39x87  39x89  39x92
##      2      2      2      2      1      2      3      1      1      2      2
##  39x93  39x95  39x96  39x98  62x10 62x103 62x105 62x109 62x116 62x119 62x122
##      2      2      2      2      1      2      2      1      1      2      1
## 62x125 62x128 62x129  62x13 62x130 62x131 62x134 62x135 62x136 62x137 62x138
##      1      1      2      2      1      2      1      2      2      1      1
## 62x139  62x14 62x140 62x143 62x144 62x147 62x151 62x152 62x153 62x154 62x157
##      1      1      2      1      2      1      1      3      2      2      2
## 62x158  62x16 62x162 62x167 62x168 62x169  62x17 62x172 62x173 62x175  62x18
##      2      2      2      2      1      1      2      2      2      2      1
## 62x182 62x185 62x188 62x189  62x19 62x190 62x197 62x200 62x206  62x21 62x214
##      1      2      1      1      2      2      2      2      2      2      1
## 62x218 62x219  62x22 62x220 62x225 62x233 62x234 62x240 62x251 62x252 62x258
##      2      2      2      1      2      1      1      2      2      1      1
## 62x261 62x262 62x264 62x265 62x268 62x276 62x282 62x293 62x296 62x300 62x302
##      2      2      2      1      2      1      1      2      1      1      1
## 62x303 62x304 62x305 62x306 62x307 62x308 62x309 62x310 62x313 62x315 62x322
##      2      1      1      1      2      1      2      1      1      1      1
## 62x324 62x327 62x332 62x337 62x338 62x351 62x355 62x359 62x363 62x370   62x5
##      1      1      2      2      2      1      2      2      2      1      1
##  62x57  62x58  62x59  62x60  62x63  62x65  62x66  62x69  62x73  62x74  62x75
##      2      2      2      1      1      1      1      2      2      1      1
##  62x77  62x83  62x87  62x91  62x92  62x94  62x96   BRAC
##      1      2      1      2      2      1      1      2
```

```
# Aggregate spectra by sample names using mean
bract_spec_avg <- aggspec(bract_spec, by = bract_samples, FUN = mean)
petal_spec_avg <- aggspec(petal_spec, by = petal_samples, FUN = mean)
labellum_spec_avg <- aggspec(labellum_spec, by = labellum_samples, FUN = mean)
```

Fix Negative Reflectance Values

Negative reflectance values are corrected by adding the minimum reflectance.

```
# Fix negative values by adding the minimum reflectance
bract_spec_avg <- procspec(bract_spec_avg, fixneg = "addmin")
```

```
## processing options applied:
## Negative value correction: added min to all reflectance
```

```
petal_spec_avg <- procspec(petal_spec_avg, fixneg = "addmin")
```

```
## processing options applied:
```

```
## Negative value correction: added min to all reflectance
```

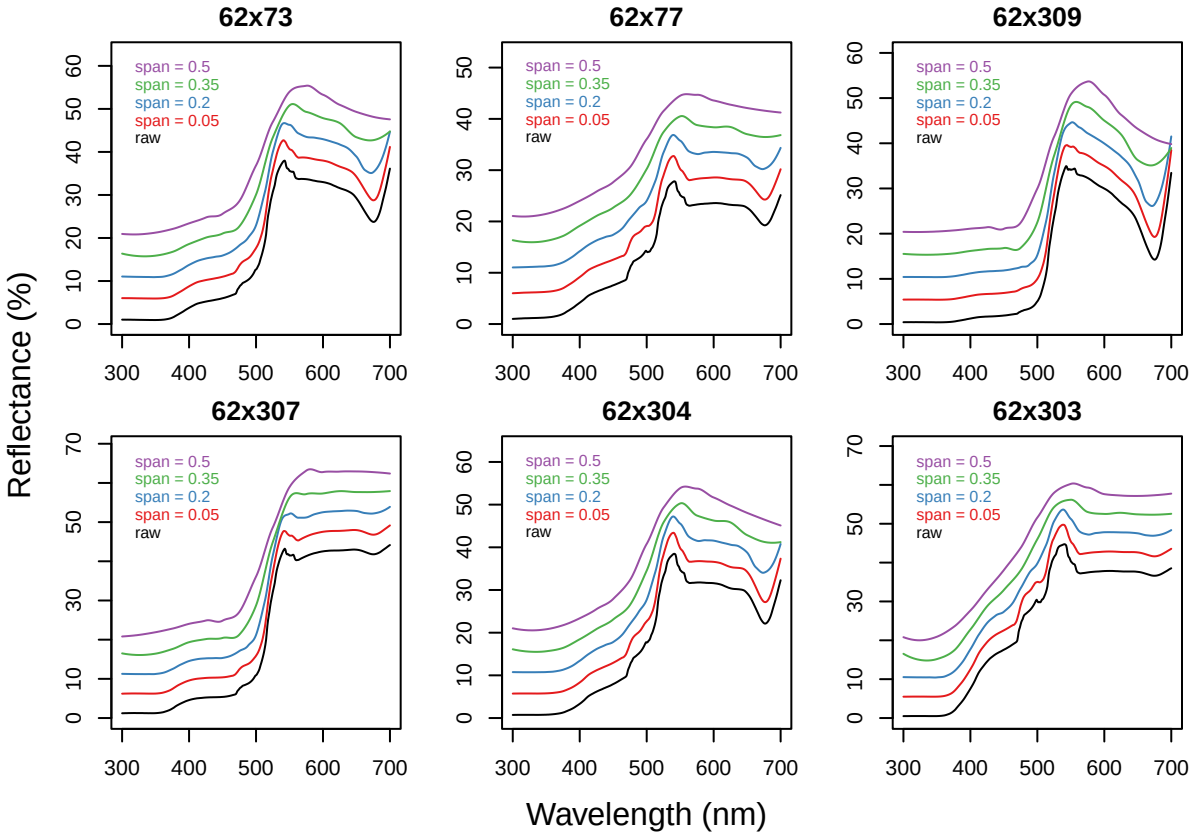
```
labellum_spec_avg <- procspec(labellum_spec_avg, fixneg = "addmin")
```

```
## processing options applied:
## Negative value correction: added min to all reflectance
```

Determine Smoothing Parameter

Use plotsmooth to visualize and decide on an appropriate smoothing span.

```
# Plot to determine suitable smoothing span
plotsmooth(bract_spec_avg[,1:7],
           minsmooth = 0.05,
           maxsmooth = 0.5,
           curves = 4,
           ask = FALSE)
```



Choose a span (e.g., 0.2) based on the plot to balance smoothness and data fidelity.

Smooth the Spectral Data

Apply smoothing to the spectral data using the chosen span.

```
# Apply smoothing with span = 0.2
bract_spec_sm <- procspec(bract_spec_avg, opt = "smooth", span = 0.2)
```

```
## processing options applied:
## smoothing spectra with a span of 0.2
```

```
petal_spec_sm <- procspec(petal_spec_avg, opt = "smooth", span = 0.2)
```

```
## processing options applied:
## smoothing spectra with a span of 0.2
```

```
labellum_spec_sm <- procspec(labellum_spec_avg, opt = "smooth", span = 0.2)
```

```
## processing options applied:
## smoothing spectra with a span of 0.2
```

Scale the Spectral Data

Scale the spectral data to different reference points for comparative analysis.

```
# Scale spectra to both minimum and maximum reflectance
bract_spec_scaleminmax <- procspec(bract_spec_sm, opt = c("min", "max"))
```

```
## processing options applied:
## Scaling spectra to a minimum value of zero
## Scaling spectra to a maximum value of 1
```

```
petal_spec_scaleminmax <- procspec(petal_spec_sm, opt = c("min", "max"))
```

```
## processing options applied:
## Scaling spectra to a minimum value of zero
## Scaling spectra to a maximum value of 1
```

```
labellum_spec_scaleminmax <- procspec(labellum_spec_sm, opt = c("min", "max"))
```

```
## processing options applied:
## Scaling spectra to a minimum value of zero
## Scaling spectra to a maximum value of 1
```

Plot Processed Spectra

Visualize the processed spectral data for each floral tissue, highlighting specific samples.

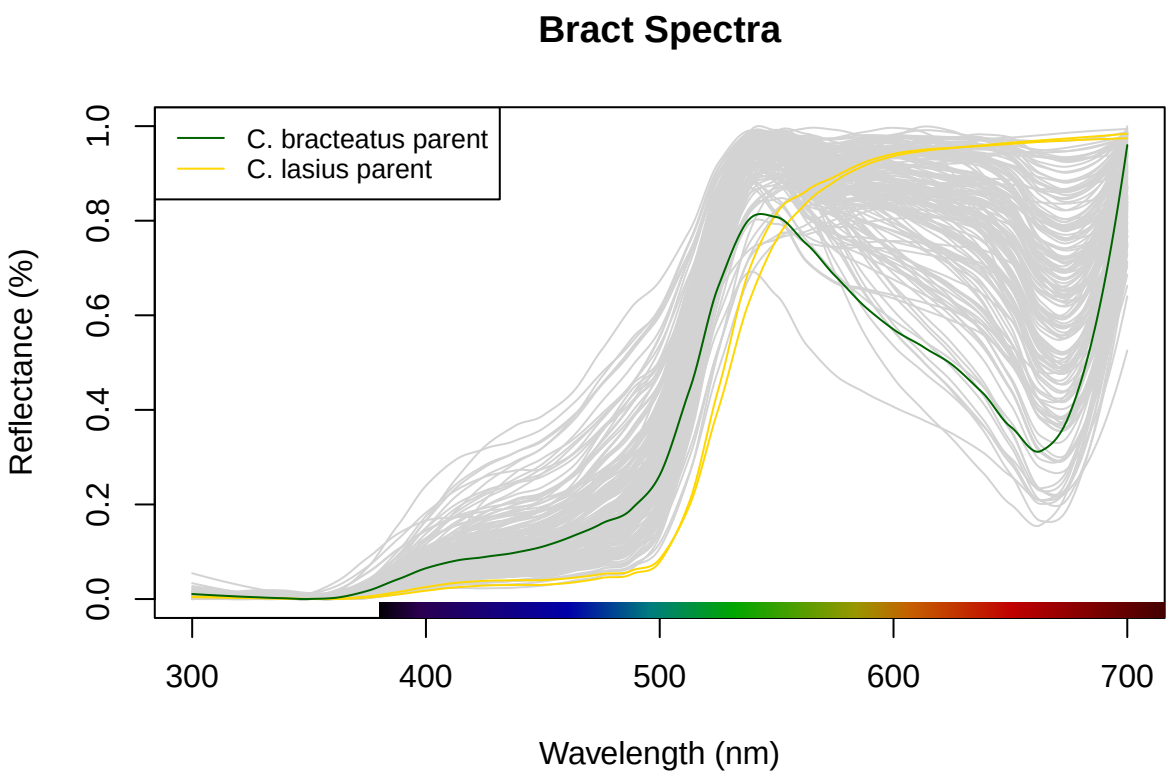
Define Common Plotting Parameters

```
# Define a color palette for highlighting specific samples
highlight_colors <- c(rep("lightgrey", 170), "gold1", "gold1", "darkgreen")
```

Bract Spectra Plot

```
# Order samples to highlight specific ones
order_spec_bract <- c(
  which(names(bract_spec_scaleminmax) == "125"),
  which(names(bract_spec_scaleminmax) == "126"),
  which(names(bract_spec_scaleminmax) == "BRAC")
)

# Reorder columns to place highlighted samples at the end
columns_bract <- 1:ncol(bract_spec_scaleminmax)
order_spec_bract <- c(columns_bract[columns_bract %notin% order_spec_bract], order_spec_bract)

# Plot the spectra
```

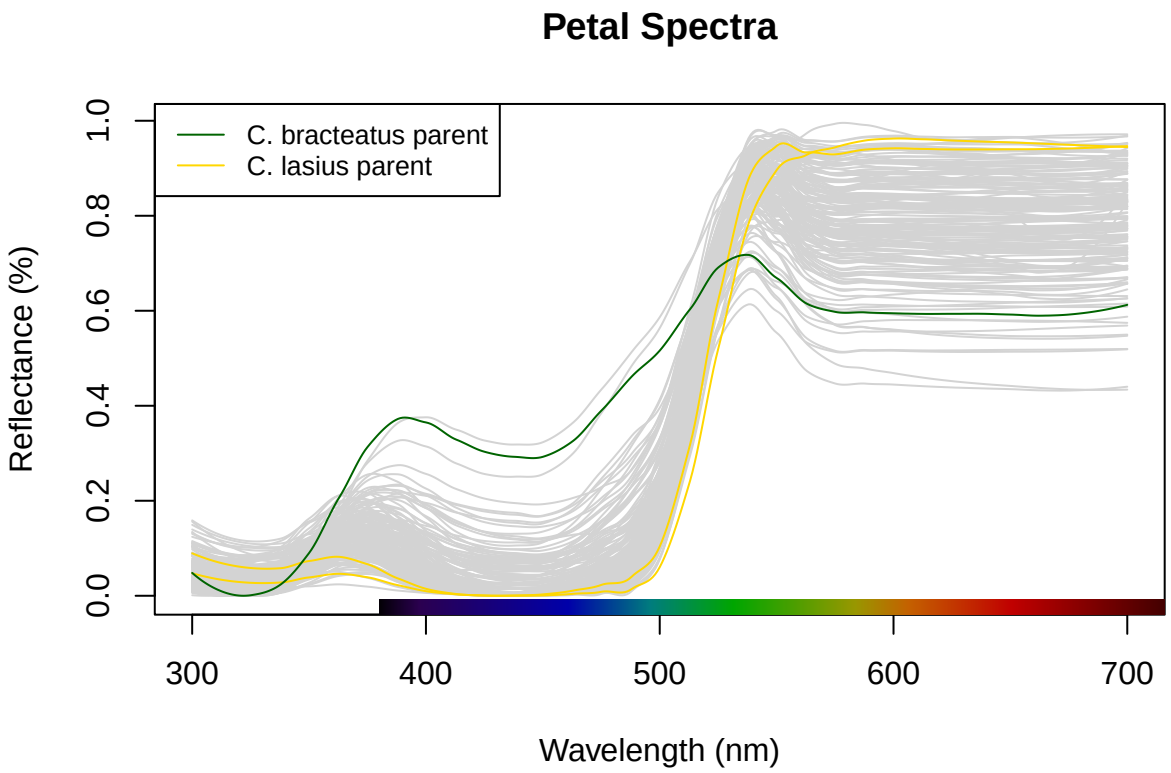
```
plot(bract_spec_scaleminmax[order_spec_bract], type = "o",
     col = highlight_colors, main = "Bract Spectra",
     xlab = "Wavelength (nm)", ylab = "Reflectance (%)")

# Add a legend
legend("topleft", legend = c("C. bracteatus parent", "C. lasius parent"),
       col = c("darkgreen", "gold1"), lty = 1, cex = 0.8)
```



Petal Spectra Plot

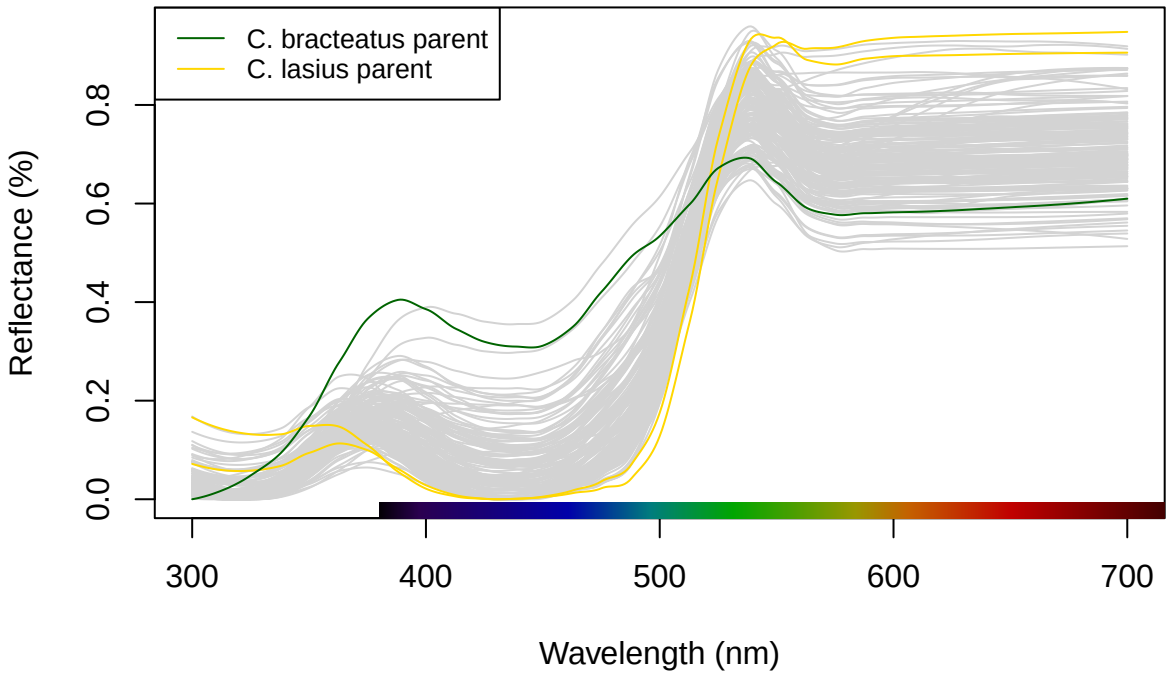
```
# Order samples to highlight specific ones
order_spec_petal <- c(
  which(names(petal_spec_scaleminmax) == "125"),
  which(names(petal_spec_scaleminmax) == "126"),
  which(names(petal_spec_scaleminmax) == "BRAC")
)

# Reorder columns to place highlighted samples at the end
columns_petal <- 1:ncol(petal_spec_scaleminmax)
order_spec_petal <- c(columns_petal[columns_petal %notin% order_spec_petal], order_spec_petal)

# Plot the spectra
plot(petal_spec_scaleminmax[order_spec_petal], type = "o",
     col = highlight_colors, main = "Petal Spectra",
     xlab = "Wavelength (nm)", ylab = "Reflectance (%)")

# Add a legend
legend("topleft", legend = c("C. bracteatus parent", "C. lasius parent"),
```

```
       col = c("darkgreen", "gold1"), lty = 1, cex = 0.8)
```



Labellum Spectra Plot

```
# Order samples to highlight specific ones
order_spec_labellum <- c(
  which(names(labellum_spec_scaleminmax) == "125"),
  which(names(labellum_spec_scaleminmax) == "126"),
  which(names(labellum_spec_scaleminmax) == "BRAC")
)

# Reorder columns to place highlighted samples at the end
columns_labellum <- 1:ncol(labellum_spec_scaleminmax)
order_spec_labellum <- c(columns_labellum[columns_labellum %notin% order_spec_labellum], order_spec_lab

# Plot the spectra
plot(labellum_spec_scaleminmax[order_spec_labellum], type = "o",
     col = highlight_colors, main = "Labellum Spectra",
     xlab = "Wavelength (nm)", ylab = "Reflectance (%)")

# Add a legend
legend("topleft", legend = c("C. bracteatus parent", "C. lasius parent"),
       col = c("darkgreen", "gold1"), lty = 1, cex = 0.8)
```



Combine and Save Spectral Plots with Highlighted Parents

```
# Save combined spectra plots with highlighted parental samples
pdf("./results/figures/Combined_Spectra_Bract_Petal_Labellum_Cbracteatus_Clasius_hybrids.pdf", width = 3
par(mfrow = c(1, 3))
# Increase left margin to provide more space for y-axis labels
par(mar = c(5, 5, 4, 2) + 0.1)  # c(bottom, left, top, right)
plot(bract_spec_scaleminmax[order_spec_bract], type = "o", col = highlight_colors,
     main = "Bract spectra", xlab = "Wavelength (nm)", ylab = "Reflectance (%)",
     cex.main = 3, cex.lab = 2.5, cex.axis = 2, lwd = 2)
legend("topleft", legend = c("C. bracteatus parent", "C. lasius parent"),
       col = c("darkgreen", "gold1"), lty = 1, cex = 2.5, bty = "n")
plot(petal_spec_scaleminmax[order_spec_petal], type = "o", col = highlight_colors,
     main = "Petal spectra", xlab = "Wavelength (nm)", ylab = "Reflectance (%)",
     cex.main = 3, cex.lab = 2.5, cex.axis = 2, lwd = 2)
legend("topleft", legend = c("C. bracteatus parent", "C. lasius parent"),
       col = c("darkgreen", "gold1"), lty = 1, cex = 2.5, bty = "n")
plot(labellum_spec_scaleminmax[order_spec_labellum], type = "o", col = highlight_colors,
     main = "Labellum spectra", xlab = "Wavelength (nm)", ylab = "Reflectance (%)",
     cex.main = 3, cex.lab = 2.5, cex.axis = 2, lwd = 2)
legend("topleft", legend = c("C. bracteatus parent", "C. lasius parent"),
       col = c("darkgreen", "gold1"), lty = 1, cex = 2.5, bty = "n")
dev.off()
```

```
## pdf
##   2
```

Principal component analysis - Bract

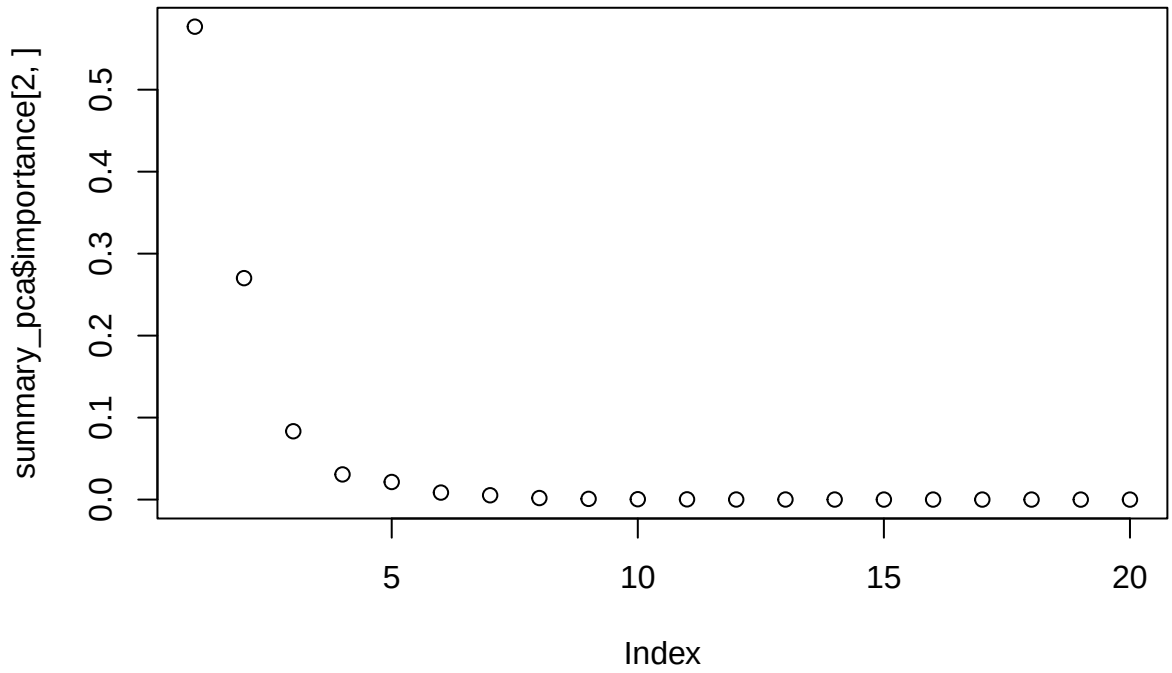
```
# opt = 'center' centers the spectra to have a mean reflectance of zero (thus removing brightness as a
# opt = 'bin' bins the spectra into user-defined bins
spec.bin <- procspec(bract_spec_scaleminmax, opt = c("bin", "center"))
```

```
## processing options applied:
## Centering spectra to a mean of zero
## binned spectra to 21-nm intervals
```

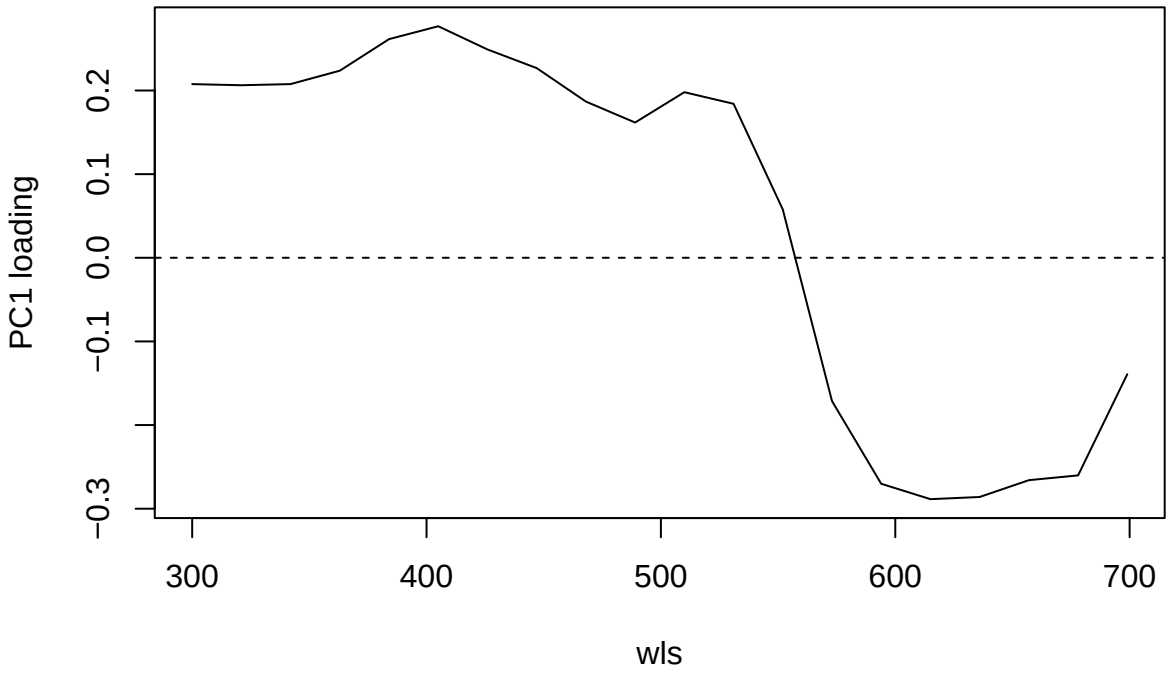
```
# Transpose so wavelength are variables for the PCA
spec.bin <- t(spec.bin)

# Names variables as wavelength bins
colnames(spec.bin) <- spec.bin[1, ]
spec.bin <- spec.bin[-1, ] # remove 'wl' row

# Run PCA on the processed spectra data
pca1 <- prcomp(spec.bin, scale. = TRUE)
(summary_pca <- summary(pca1))
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     3.3966 2.3243 1.29094 0.78343 0.65511 0.41299 0.32437
## Proportion of Variance 0.5768 0.2701 0.08333 0.03069 0.02146 0.00853 0.00526
## Cumulative Proportion  0.5768 0.8470 0.93030 0.96099 0.98244 0.99097 0.99623
##                            PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.19439 0.13133 0.09227 0.07497 0.04561 0.04302 0.03454
## Proportion of Variance 0.00189 0.00086 0.00043 0.00028 0.00010 0.00009 0.00006
## Cumulative Proportion  0.99812 0.99898 0.99941 0.99969 0.99979 0.99989 0.99995
##                           PC15    PC16    PC17    PC18     PC19     PC20
## Standard deviation     0.02078 0.01617 0.01300 0.01077 0.006633 0.005973
## Proportion of Variance 0.00002 0.00001 0.00001 0.00001 0.000000 0.000000
## Cumulative Proportion  0.99997 0.99998 0.99999 1.00000 1.000000 1.000000
```

```
plot(summary_pca$importance[2,])
```

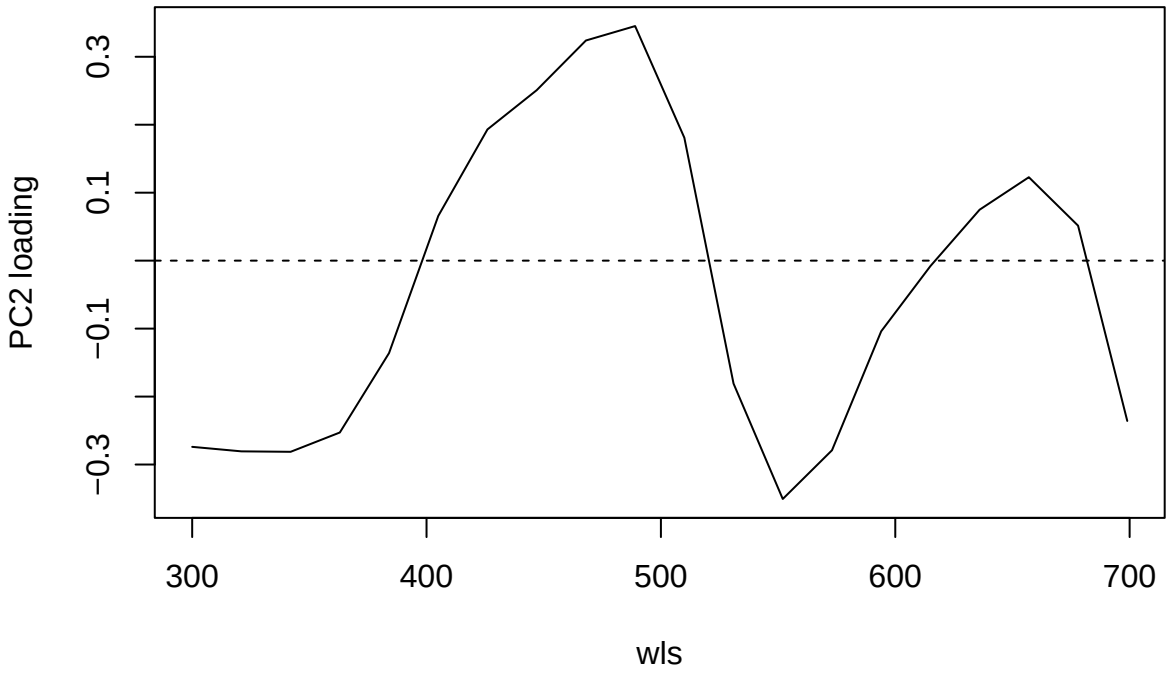



```
# Convert column names (wavelength bins) to numeric values and assign to 'wls'
wls <- as.numeric(colnames(spec.bin))

# Plot loadings of the first principal component (PC1)
plot(pca1$rotation[, 1] ~ wls, type = "l", ylab = "PC1 loading")
abline(h = 0, lty = 2)
```

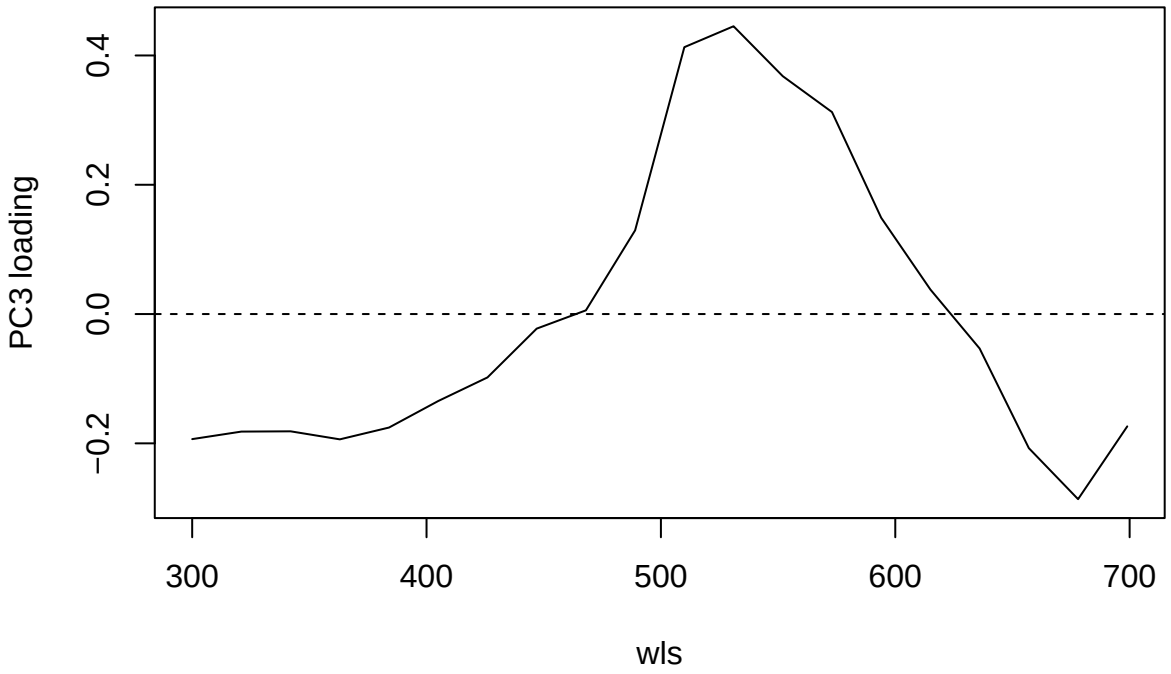



```
## appears to contrast shorter wavelengths (300-570 nm) with long wavelengths
## (570-700 nm)

# Plot loadings of the second principal component (PC2)
```

```r
plot(pca1$rotation[, 2] ~ wls, type = "l", ylab = "PC2 loading")
abline(h = 0, lty = 2)
```




```r
## confusing. UV wavelengths (300-400) are correlated with warm colors
## (520-630 nm -- green, yellow, orange, red)
## cool wavelengths (400-520 nm) are correlated with hot colors (630-700 nm -- red)

# Plot loadings of the third principal component (PC2)
plot(pca1$rotation[, 3] ~ wls, type = "l", ylab = "PC3 loading")
abline(h = 0, lty = 2)
```



```
## Confusing. wavelengths 490-610 nm (huge range including most of visible light)
## are contrasted with <490 nm (UV and dark blue/violet) and >610 nm (orange/red)

# Extracting PCA Scores
pca_bract <- data.frame(PC1_bract=pca1$x[,1], PC2_bract=pca1$x[,2], PC3_bract=pca1$x[,3])
```

Principal component analysis - Petal

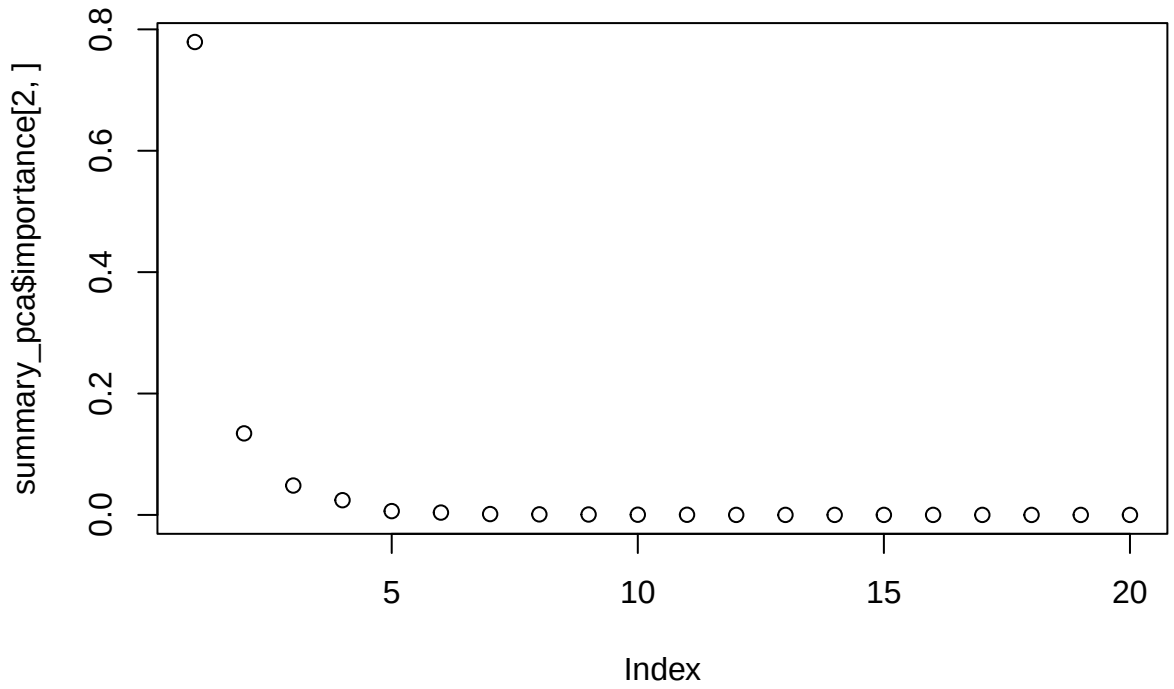
```
# opt = 'center' centers the spectra to have a mean reflectance of zero (thus removing brightness as a
# opt = 'bin' bins the spectra into user-defined bins
spec.bin <- procspec(petal_spec_scaleminmax, opt = c("bin", "center"))
```

```
## processing options applied:
## Centering spectra to a mean of zero
## binned spectra to 21-nm intervals
```

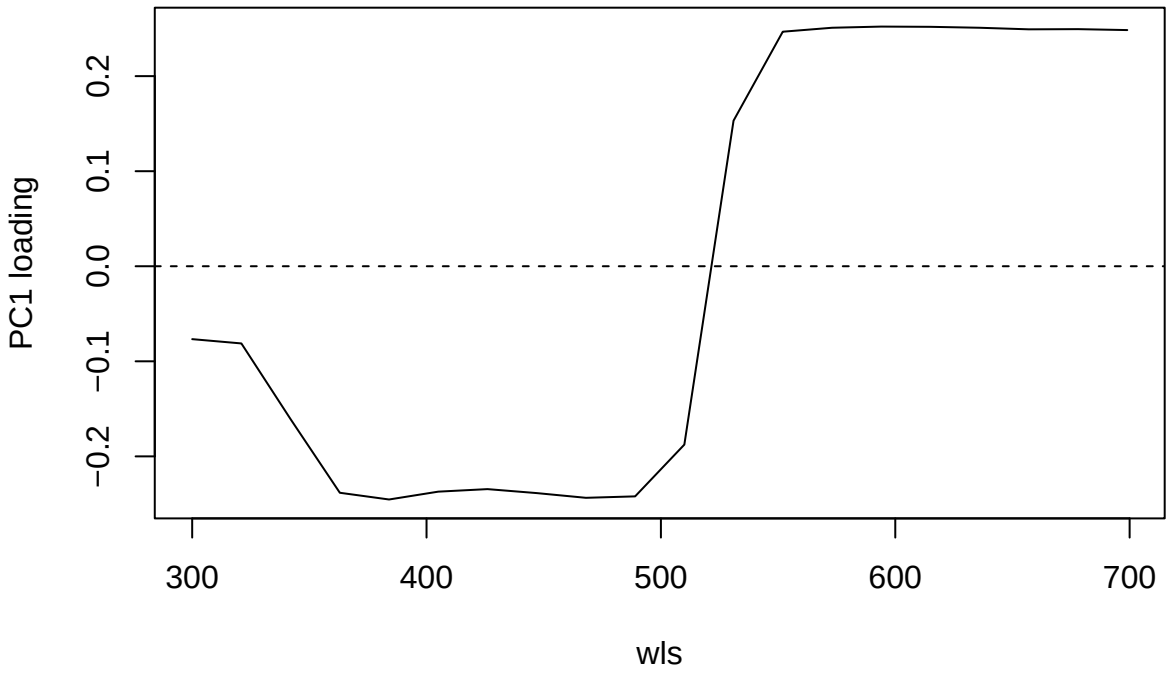
```
# Transpose so wavelength are variables for the PCA
spec.bin <- t(spec.bin)

# Names variables as wavelength bins
colnames(spec.bin) <- spec.bin[1, ]
spec.bin <- spec.bin[-1, ] # remove 'wl' row

# Run PCA on the processed spectra data
pca1 <- prcomp(spec.bin, scale. = TRUE)
(summary_pca <- summary(pca1))
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     3.9476 1.6392 0.98410 0.69632 0.35431 0.27714 0.16604
## Proportion of Variance 0.7792 0.1343 0.04842 0.02424 0.00628 0.00384 0.00138
## Cumulative Proportion  0.7792 0.9135 0.96193 0.98617 0.99245 0.99629 0.99767
##                           PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.1345 0.11777 0.08335 0.06943 0.03063 0.02532 0.02234
## Proportion of Variance 0.0009 0.00069 0.00035 0.00024 0.00005 0.00003 0.00002
## Cumulative Proportion  0.9986 0.99927 0.99961 0.99985 0.99990 0.99993 0.99996
##                           PC15    PC16    PC17     PC18     PC19     PC20
## Standard deviation     0.01865 0.01602 0.01141 0.007871 0.005009 0.001636
## Proportion of Variance 0.00002 0.00001 0.00001 0.000000 0.000000 0.000000
## Cumulative Proportion  0.99998 0.99999 1.00000 1.000000 1.000000 1.000000
```

```
plot(summary_pca$importance[2,])
```

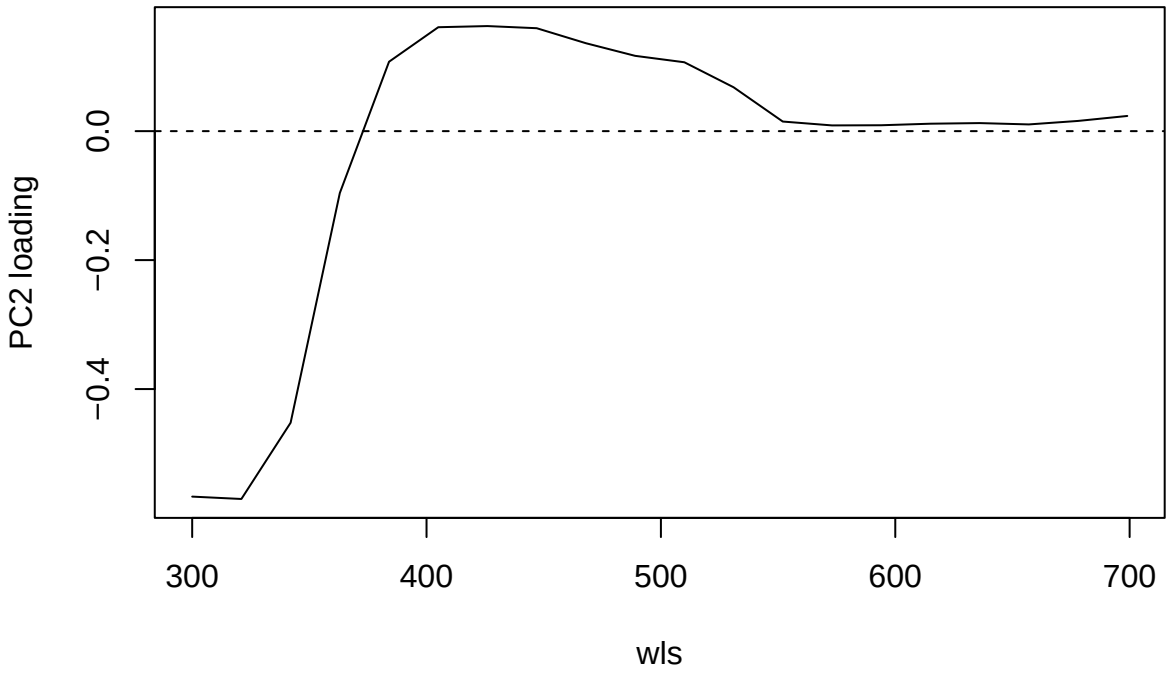



```
# Convert column names (wavelength bins) to numeric values and assign to 'wls'
wls <- as.numeric(colnames(spec.bin))

# Plot loadings of the first principal component (PC1)
plot(pca1$rotation[, 1] ~ wls, type = "l", ylab = "PC1 loading")
abline(h = 0, lty = 2)
```

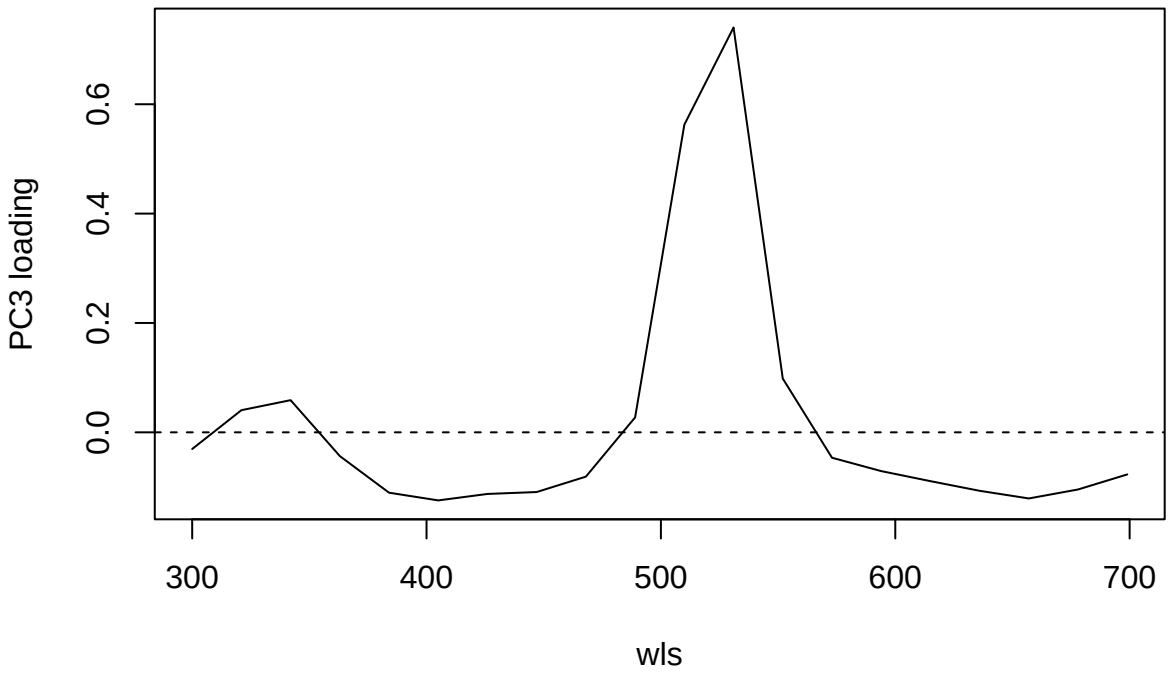



```
## appears to contrast short wavelengths (310-510 nm) with long wavelengths
## (510-700 nm)

# Plot loadings of the second principal component (PC2)
```

```r
plot(pca1$rotation[, 2] ~ wls, type = "l", ylab = "PC2 loading")
abline(h = 0, lty = 2)
```




```r
## appears to be associated with the UV range (300-400 nm)

# Plot loadings of the third principal component (PC2)
plot(pca1$rotation[, 3] ~ wls, type = "l", ylab = "PC3 loading")
abline(h = 0, lty = 2)
```




```r
## appears to be associated with wavelengths 500-550 nm (green)
```

```r
# Extracting PCA Scores
pca_petal <- data.frame(PC1_petal=pca1$x[,1], PC2_petal=pca1$x[,2], PC3_petal=pca1$x[,3])
```

Principal component analysis - Labellum

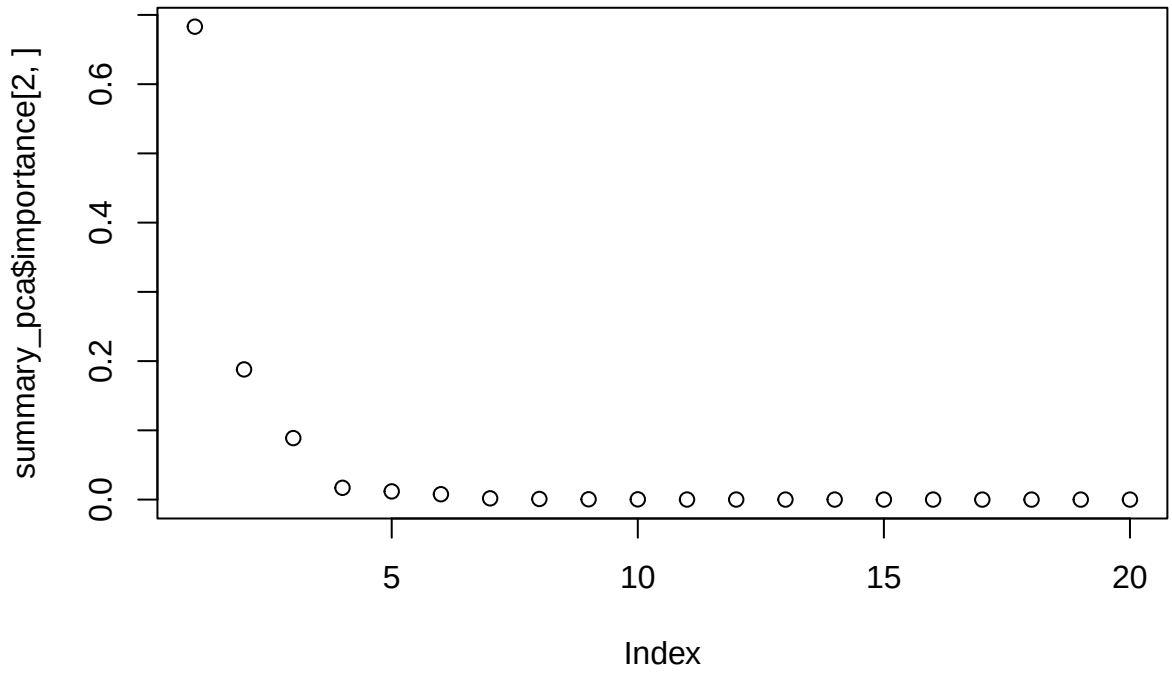
```r
# opt = 'center' centers the spectra to have a mean reflectance of zero (thus removing brightness as a
# opt = 'bin' bins the spectra into user-defined bins
spec.bin <- procspec(labellum_spec_scaleminmax, opt = c("bin", "center"))
```

```
## processing options applied:
## Centering spectra to a mean of zero
## binned spectra to 21-nm intervals
```

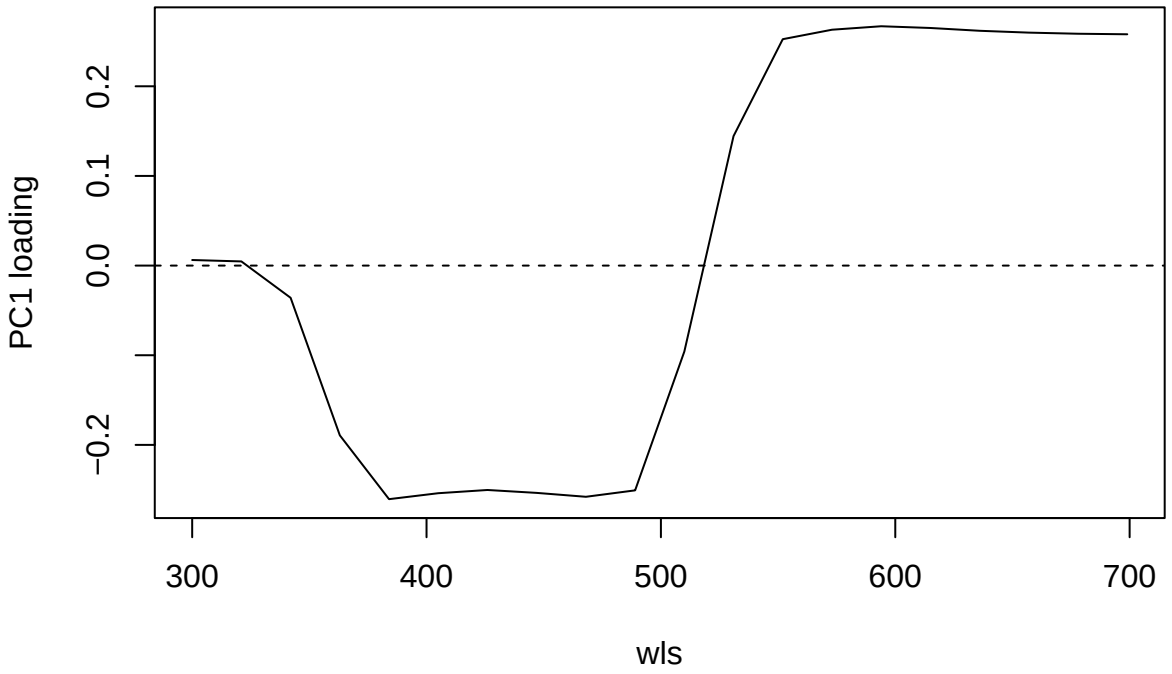
```r
# Transpose so wavelength are variables for the PCA
spec.bin <- t(spec.bin)

# Names variables as wavelength bins
colnames(spec.bin) <- spec.bin[1, ]
spec.bin <- spec.bin[-1, ] # remove 'wl' row

# Run PCA on the processed spectra data
pca1 <- prcomp(spec.bin, scale. = TRUE)
(summary_pca <- summary(pca1))
```

```
## Importance of components:
##                           PC1   PC2     PC3    PC4     PC5     PC6    PC7
## Standard deviation     3.696 1.939 1.33257 0.5831 0.48625 0.39272 0.1896
## Proportion of Variance 0.683 0.188 0.08879 0.0170 0.01182 0.00771 0.0018
## Cumulative Proportion  0.683 0.871 0.95981 0.9768 0.98863 0.99634 0.9981
##                           PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.13245 0.09023 0.08153 0.03977 0.03711 0.02527 0.02280
## Proportion of Variance 0.00088 0.00041 0.00033 0.00008 0.00007 0.00003 0.00003
## Cumulative Proportion  0.99902 0.99942 0.99975 0.99983 0.99990 0.99993 0.99996
##                           PC15    PC16     PC17     PC18     PC19     PC20
## Standard deviation     0.01880 0.01617 0.009696 0.007808 0.003989 0.001307
## Proportion of Variance 0.00002 0.00001 0.000000 0.000000 0.000000 0.000000
## Cumulative Proportion  0.99998 0.99999 1.000000 1.000000 1.000000 1.000000
```

```r
plot(summary_pca$importance[2,])
```

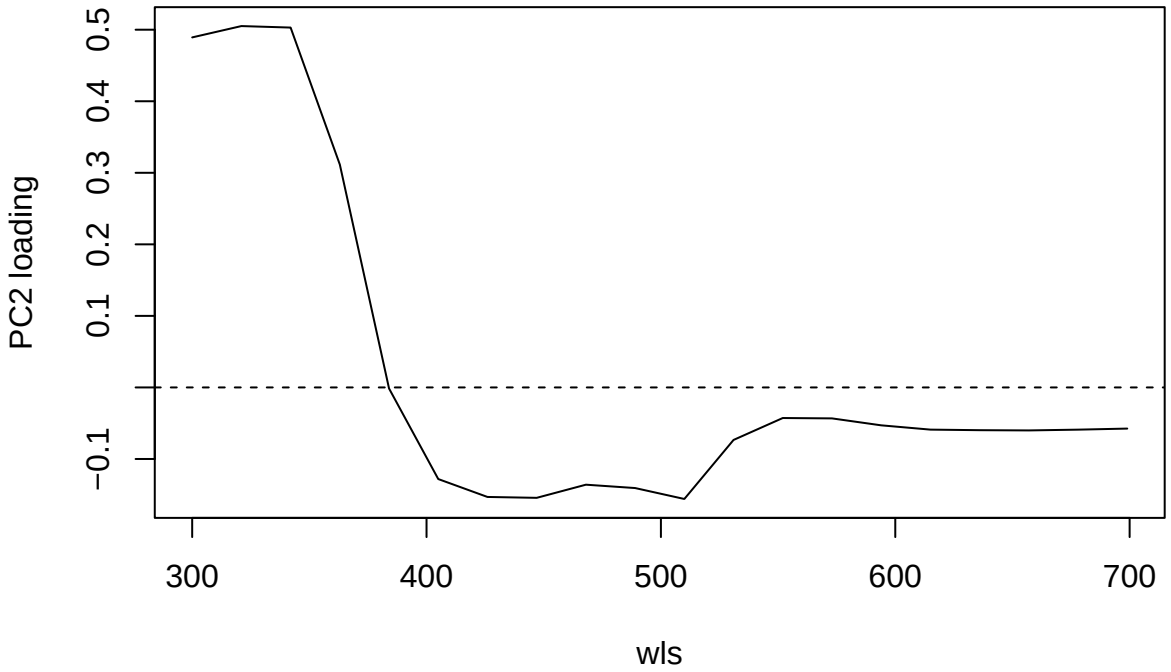



```
# Convert column names (wavelength bins) to numeric values and assign to 'wls'
wls <- as.numeric(colnames(spec.bin))

# Plot loadings of the first principal component (PC1)
plot(pca1$rotation[, 1] ~ wls, type = "l", ylab = "PC1 loading")
abline(h = 0, lty = 2)
```

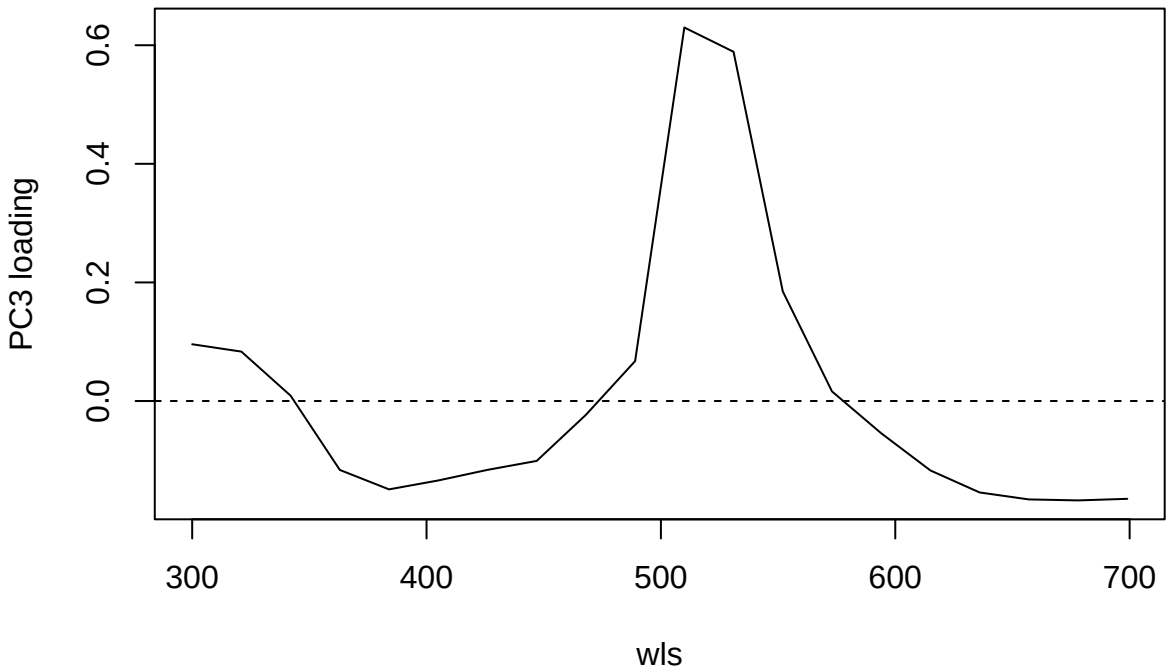



```
## appears to contrast shorter wavelengths (300-520 nm) with long wavelengths
## (520-700 nm)

# Plot loadings of the second principal component (PC2)
```

```r
plot(pca1$rotation[, 2] ~ wls, type = "l", ylab = "PC2 loading")
abline(h = 0, lty = 2)
```




```r
## appears to be associated with UV wavelengths (300-390 nm)

# Plot loadings of the third principal component (PC2)
plot(pca1$rotation[, 3] ~ wls, type = "l", ylab = "PC3 loading")
abline(h = 0, lty = 2)
```




```r
## appears to be associated with wavelengths 500-570 nm (green, yellow)
```

```r
# Extracting PCA Scores
pca_labellum <- data.frame(PC1_labellum=pca1$x[,1], PC2_labellum=pca1$x[,2], PC3_labellum=pca1$x[,3])
```

Format PC Scores

```r
# Replace 'x' with '_' in row names
rownames(pca_bract) <- gsub("x", "_", rownames(pca_bract))
rownames(pca_petal) <- gsub("x", "_", rownames(pca_petal))
rownames(pca_labellum) <- gsub("x", "_", rownames(pca_labellum))

# Convert row names to a column named 'id'
pca_bract <- pca_bract %>% rownames_to_column(var = "id")
pca_petal <- pca_petal %>% rownames_to_column(var = "id")
pca_labellum <- pca_labellum %>% rownames_to_column(var = "id")
```

Spectral Descriptors

We compute various spectral descriptors to quantify the color properties of the floral tissues.

Define spectral descriptors

B1: Total brightness B2: Mean brightness B3: Intensity (Rmax) S1U to S1R: Relative contributions of UV, Violet, Blue, Green, Yellow, and Red spectral ranges to total brightness S2: Spectral saturation (Rmax/Rmin) S3: Chroma S4: Spectral purity S5: Chroma S6: Contrast (Rmax - Rmin) S7: Spectral saturation S8: Chroma ((Rmax - Rmin)/B2) S9: Carotenoid chroma ((R700 - R450)/R700) S10: Peaky chroma H1 to H5: Hue metrics (e.g., peak wavelength) Note: Some metrics may be sensitive to spectral noise.

Calculate Spectral Descriptors

Bract

```r
# Calculate spectral descriptors for bract
summary_bract <- summary(bract_spec_scaleminmax)

# Remove S2 due to infinite values
summary_bract <- summary_bract %>% select(-S2)

# Extract metrics for parents
bract125 <- round(summary_bract["125", ], 3)
bract126 <- round(summary_bract["126", ], 3)
bractBRAC <- round(summary_bract["BRAC", ], 3)
```

Plot Bract Spectral Descriptors Histograms

```r
pdf("./results/figures/bract_descriptors_histograms.pdf", width = 14, height = 10)

# Set up plotting area: 4 rows x 6 columns for histograms
par(mfrow = c(4, 6),        # 4 rows, 6 columns
    mar = c(5, 5, 4, 2) + 0.1,  # Margins for each plot: bottom, left, top, right
    oma = c(0, 0, 0, 5))    # Outer margins: bottom, left, top, right

# Loop through each metric and plot histogram with parental lines
for (i in 1:ncol(summary_bract)) {
```

```
  hist(summary_bract[, i],
       xlab = colnames(summary_bract)[i],
       main = colnames(summary_bract)[i],
       col = "lightgrey",
       border = "white")

  # Add vertical lines for parental samples
  abline(v = bract125[i], col = 'gold1', lwd = 3)      # C. lasius parent
  abline(v = bract126[i], col = 'gold1', lwd = 3)      # C. lasius parent
  abline(v = bractBRAC[i], col = 'darkgreen', lwd = 3) # C. bracteatus parent
}

# Allow drawing in the outer margin
par(xpd = TRUE)

plot.new()

# Add a shared legend in the outer right margin
legend("topright",
       inset = c(0, 0),  # Adjusts the position of the legend
       legend = c("C. lasius parent", "C. bracteatus parent"),
       col = c("gold1", "darkgreen"),
       lty = 1,              # Line type: solid
       lwd = 3,              # Line width
       cex = 1.5,            # Text size
       bty = "n")            # No box around the legend

# Close the PDF device to save the file
dev.off()
```

```
## pdf
##   2
```

Petal Summary

```
# Calculate summary statistics for petal
summary_petal <- summary(petal_spec_scaleminmax)

# Remove S2 due to infinite values
summary_petal <- summary_petal %>% select(-S2)

# Extract metrics for specific samples
petal125 <- round(summary_petal["125", ], 3)
petal126 <- round(summary_petal["126", ], 3)
petalBRAC <- round(summary_petal["BRAC", ], 3)
```

Plot Petal Spectral Descriptors Histograms

```
pdf("./results/figures/petal_descriptors_histograms.pdf", width = 14, height = 10)

# Set up plotting area: 4 rows x 6 columns for histograms
par(mfrow = c(4, 6),        # 4 rows, 6 columns
    mar = c(5, 5, 4, 2) + 0.1,  # Margins for each plot: bottom, left, top, right
```

```
    oma = c(0, 0, 0, 5))    # Outer margins: bottom, left, top, right

# Loop through each metric and plot histogram with parental lines
for (i in 1:ncol(summary_petal)) {
  hist(summary_petal[, i],
       xlab = colnames(summary_petal)[i],
       main = colnames(summary_petal)[i],
       col = "lightgrey",
       border = "white")

  # Add vertical lines for parental samples
  abline(v = petal125[i], col = 'gold1', lwd = 3)       # C. lasius parent
  abline(v = petal126[i], col = 'gold1', lwd = 3)       # C. lasius parent
  abline(v = petalBRAC[i], col = 'darkgreen', lwd = 3) # C. bracteatus parent
}

# Allow drawing in the outer margin
par(xpd = TRUE)

plot.new()

# Add a shared legend in the outer right margin
legend("topright",
       inset = c(0, 0),  # Adjusts the position of the legend
       legend = c("C. lasius parent", "C. bracteatus parent"),
       col = c("gold1", "darkgreen"),
       lty = 1,              # Line type: solid
       lwd = 3,              # Line width
       cex = 1.5,            # Text size
       bty = "n")            # No box around the legend

# Close the PDF device to save the file
dev.off()
```

```
## pdf
##   2
```

Labellum Summary

```
# Calculate summary statistics for labellum
summary_labellum <- summary(labellum_spec_scaleminmax)

# Remove S2 due to infinite values
summary_labellum <- summary_labellum %>% select(-S2)

# Extract metrics for specific samples
labellum125 <- round(summary_labellum["125", ], 3)
labellum126 <- round(summary_labellum["126", ], 3)
labellumBRAC <- round(summary_labellum["BRAC", ], 3)
```

Plot Labellum Spectral Descriptors Histograms

```r
pdf("./results/figures/labellum_descriptors_histograms.pdf", width = 14, height = 10)

# Set up plotting area: 4 rows x 6 columns for histograms
par(mfrow = c(4, 6),        # 4 rows, 6 columns
    mar = c(5, 5, 4, 2) + 0.1,  # Margins for each plot: bottom, left, top, right
    oma = c(0, 0, 0, 5))    # Outer margins: bottom, left, top, right

# Loop through each metric and plot histogram with parental lines
for (i in 1:ncol(summary_labellum)) {
  hist(summary_labellum[, i],
       xlab = colnames(summary_labellum)[i],
       main = colnames(summary_labellum)[i],
       col = "lightgrey",
       border = "white")

  # Add vertical lines for parental samples
  abline(v = labellum125[i], col = 'gold1', lwd = 3)      # C. lasius parent
  abline(v = labellum126[i], col = 'gold1', lwd = 3)      # C. lasius parent
  abline(v = labellumBRAC[i], col = 'darkgreen', lwd = 3) # C. bracteatus parent
}

# Allow drawing in the outer margin
par(xpd = TRUE)

plot.new()

# Add a shared legend in the outer right margin
legend("topright",
       inset = c(0, 0),  # Adjusts the position of the legend
       legend = c("C. lasius parent", "C. bracteatus parent"),
       col = c("gold1", "darkgreen"),
       lty = 1,              # Line type: solid
       lwd = 3,              # Line width
       cex = 1.5,            # Text size
       bty = "n")            # No box around the legend

# Close the PDF device to save the file
dev.off()
```

```
## pdf
##   2
```

Data Formatting and Export

Reformat Rownames

Replace 'x' with '_' in row names for consistency.

```r
# Replace 'x' with '_' in row names
rownames(summary_bract) <- gsub("x", "_", rownames(summary_bract))
rownames(summary_petal) <- gsub("x", "_", rownames(summary_petal))
rownames(summary_labellum) <- gsub("x", "_", rownames(summary_labellum))
```

Subset and Rename Columns

Select relevant metrics and rename columns to indicate their corresponding floral tissue.

```
# Subset relevant columns based on analysis needs
summary_bract <- summary_bract[, c("S1U", "S1V", "S1B", "S1G", "S1R", "S5", "S9", "H4")]
summary_petal <- summary_petal[, c("B3", "S1U", "S1V", "S1B", "S1Y", "S1R", "S5", "S9", "H4")]
summary_labellum <- summary_labellum[, c("B3", "S1B", "S1Y", "S1R", "S5", "S6", "S9", "H3", "H4")]

# Rename columns to include tissue type
colnames(summary_bract) <- c("S1U_bract", "S1V_bract", "S1B_bract", "S1G_bract",
                             "S1R_bract", "S5_bract", "S9_bract", "H4_bract")
colnames(summary_petal) <- c("B3_petal", "S1U_petal", "S1V_petal", "S1B_petal",
                             "S1Y_petal", "S1R_petal", "S5_petal", "S9_petal", "H4_petal")
colnames(summary_labellum) <- c("B3_labellum", "S1B_labellum", "S1Y_labellum",
                                "S1R_labellum", "S5_labellum", "S6_labellum",
                                "S9_labellum", "H3_labellum", "H4_labellum")
```

Convert Rownames to a Column

Add the row names as a new column id to facilitate merging.

```
# Convert row names to a column named 'id'
summary_bract <- summary_bract %>% rownames_to_column(var = "id")
summary_petal <- summary_petal %>% rownames_to_column(var = "id")
summary_labellum <- summary_labellum %>% rownames_to_column(var = "id")
```

UV Analyses

Spectral Data Processing

Convert Data to rspec Objects

The spectral data is converted into rspec objects using the pavo package for further analysis.

```
# Set seed for reproducibility
set.seed(1612217)

# Convert datasets to rspec objects with wavelength limits
bract_spec <- as.rspec(bract, lim = c(300, 400), whichwl = 1)
```

```
## The spectral data contain 456 negative value(s),
##  which may produce unexpected results if used in models.
## Consider using procspec() to correct them.
```

```
petal_spec <- as.rspec(petal, lim = c(300, 400), whichwl = 1)
```

```
## The spectral data contain 86 negative value(s),
##  which may produce unexpected results if used in models.
## Consider using procspec() to correct them.
```

```
labellum_spec <- as.rspec(labellum, lim = c(300, 400), whichwl = 1)

# change column names
colnames(bract_spec) <- gsub("-", "x", colnames(bract_spec))
colnames(petal_spec) <- gsub("-", "x", colnames(petal_spec))
colnames(labellum_spec) <- gsub("-", "x", colnames(labellum_spec))
```

Note: The conversion may produce warnings about negative values in spectral data, which are addressed in subsequent steps.

Average the Spectra

Aggregate the spectral data by sample names, averaging replicates.

```
# Extract sample names by removing trailing numbers in parentheses
bract_samples <- gsub("\\([0-9]+\\)$", "", names(bract_spec))[-1]
petal_samples <- gsub("\\([0-9]+\\)$", "", names(petal_spec))[-1]
labellum_samples <- gsub("\\([0-9]+\\)$", "", names(labellum_spec))[-1]

# Verify sample counts
table(bract_samples)
```

```
## bract_samples
##    125    126     39  39x10 39x109 39x110 39x115 39x116 39x117  39x12 39x122
##      2      3      1      2      1      2      2      1      3      2      2
## 39x123 39x125 39x126  39x13 39x130 39x136  39x14  39x15  39x16  39x17   39x2
##      2      2      2      1      3      2      3      2      4      2      2
##  39x21  39x23  39x25  39x27  39x34  39x39   39x4  39x40  39x41  39x44  39x46
##      2      2      2      2      2      2      2      2      2      2      2
##  39x49  39x50  39x51  39x55  39x56  39x57   39x6  39x60  39x65  39x67  39x68
##      2      2      2      2      2      2      2      2      2      2      1
##  39x75  39x77  39x78  39x79   39x8  39x81  39x82  39x86  39x87  39x89  39x92
##      2      2      2      2      1      2      3      1      1      2      2
##  39x93  39x95  39x96  39x98  62x10 62x103 62x105 62x109 62x116 62x119 62x122
##      2      2      2      2      1      2      2      1      1      2      1
## 62x125 62x128 62x129  62x13 62x130 62x131 62x134 62x135 62x136 62x137 62x138
##      1      1      2      2      1      2      1      2      2      1      1
## 62x139  62x14 62x140 62x143 62x144 62x147 62x151 62x152 62x153 62x154 62x157
##      1      1      2      1      2      1      1      3      2      2      2
## 62x158  62x16 62x162 62x167 62x168 62x169  62x17 62x172 62x173 62x175  62x18
##      2      2      2      2      1      1      2      2      2      2      1
## 62x182 62x185 62x188 62x189  62x19 62x190 62x197 62x200 62x206  62x21 62x214
##      1      2      1      1      2      2      2      2      2      2      1
## 62x218 62x219  62x22 62x220 62x225 62x233 62x234 62x240 62x251 62x252 62x258
##      2      2      2      1      2      1      1      2      2      1      1
## 62x261 62x264 62x265 62x267 62x268 62x276 62x282 62x293 62x296 62x300 62x302
##      2      2      1      2      2      1      1      2      1      1      1
## 62x303 62x304 62x305 62x306 62x307 62x308 62x309 62x310 62x313 62x315 62x322
##      2      1      1      1      2      1      2      1      1      1      1
## 62x324 62x327 62x332 62x337 62x338 62x351 62x355 62x359 62x363 62x370   62x5
##      1      1      2      1      2      1      2      2      2      1      1
##  62x57  62x58  62x59  62x60  62x63  62x65  62x66  62x69  62x73  62x74  62x75
##      2      2      2      1      1      1      1      2      2      1      1
##  62x77  62x83  62x87  62x91  62x92  62x94  62x96   BRAC
##      1      2      1      2      2      1      1      2
```

```
table(petal_samples)
```

```
## petal_samples
##    125    126     39  39x10 39x109 39x110 39x115 39x116 39x117  39x12 39x122
##      2      2      1      2      1      2      2      1      3      2      2
## 39x123 39x125 39x126  39x13 39x130 39x136  39x14  39x15  39x16  39x17   39x2
##      2      2      2      1      3      2      3      2      4      2      2
```

```
##  39x21  39x23  39x25  39x27  39x34  39x39   39x4  39x40  39x41  39x44  39x46
##      2      2      2      2      2      2      2      2      2      2      2
##  39x49  39x50  39x51  39x55  39x56  39x57   39x6  39x60  39x65  39x67  39x68
##      2      2      2      2      2      2      2      2      2      2      1
##  39x75  39x77  39x78  39x79   39x8  39x81  39x82  39x86  39x87  39x89  39x92
##      2      2      2      2      1      2      3      1      1      2      2
##  39x93  39x95  39x96  39x98  62x10 62x103 62x105 62x109 62x116 62x119 62x122
##      2      2      2      2      1      2      2      1      1      2      1
## 62x125 62x128 62x129  62x13 62x130 62x131 62x134 62x135 62x136 62x137 62x138
##      1      1      2      2      1      2      1      2      2      1      1
## 62x139  62x14 62x140 62x143 62x144 62x147 62x151 62x152 62x153 62x154 62x157
##      1      1      2      1      2      1      1      3      2      2      2
## 62x158  62x16 62x162 62x167 62x168 62x169  62x17 62x172 62x173 62x175  62x18
##      2      2      2      2      1      1      2      2      2      2      1
## 62x182 62x185 62x188 62x189  62x19 62x190 62x197 62x200 62x206  62x21 62x214
##      1      2      1      1      2      2      2      2      2      2      1
## 62x218 62x219  62x22 62x220 62x225 62x233 62x234 62x240 62x251 62x252 62x258
##      2      2      2      1      2      1      1      2      2      1      1
## 62x261 62x262 62x264 62x265 62x268 62x276 62x282 62x293 62x296 62x300 62x302
##      2      2      2      1      2      1      1      2      1      1      1
## 62x303 62x304 62x305 62x306 62x307 62x308 62x309 62x310 62x313 62x315 62x322
##      2      1      1      1      2      1      2      1      1      1      1
## 62x324 62x327 62x332 62x337 62x338 62x351 62x355 62x359 62x363 62x370   62x5
##      1      1      2      1      2      1      2      2      2      1      1
##  62x57  62x58  62x59  62x60  62x63  62x65  62x66  62x69  62x73  62x74  62x75
##      2      2      2      1      1      1      1      2      2      1      1
##  62x77  62x83  62x87  62x91  62x92  62x94  62x96   BRAC
##      1      2      1      2      2      1      1      2
```

```
table(labellum_samples)
```

```
## labellum_samples
##    125    126     39  39x10 39x109 39x110 39x115 39x116 39x117  39x12 39x122
##      2      2      1      2      1      2      2      1      3      2      2
## 39x123 39x125 39x126  39x13 39x130 39x136  39x14  39x15  39x16  39x17   39x2
##      2      2      2      1      3      2      3      2      4      2      2
##  39x21  39x23  39x25  39x27  39x34  39x39   39x4  39x40  39x41  39x44  39x46
##      2      2      2      2      2      2      2      2      2      2      2
##  39x49  39x50  39x51  39x55  39x56  39x57   39x6  39x60  39x65  39x67  39x68
##      2      2      2      2      2      2      2      2      2      2      1
##  39x75  39x77  39x78  39x79   39x8  39x81  39x82  39x86  39x87  39x89  39x92
##      2      2      2      2      1      2      3      1      1      2      2
##  39x93  39x95  39x96  39x98  62x10 62x103 62x105 62x109 62x116 62x119 62x122
##      2      2      2      2      1      2      2      1      1      2      1
## 62x125 62x128 62x129  62x13 62x130 62x131 62x134 62x135 62x136 62x137 62x138
##      1      1      2      2      1      2      1      2      2      1      1
## 62x139  62x14 62x140 62x143 62x144 62x147 62x151 62x152 62x153 62x154 62x157
##      1      1      2      1      2      1      1      3      2      2      2
## 62x158  62x16 62x162 62x167 62x168 62x169  62x17 62x172 62x173 62x175  62x18
##      2      2      2      2      1      1      2      2      2      2      1
## 62x182 62x185 62x188 62x189  62x19 62x190 62x197 62x200 62x206  62x21 62x214
##      1      2      1      1      2      2      2      2      2      2      1
## 62x218 62x219  62x22 62x220 62x225 62x233 62x234 62x240 62x251 62x252 62x258
##      2      2      2      1      2      1      1      2      2      1      1
## 62x261 62x262 62x264 62x265 62x268 62x276 62x282 62x293 62x296 62x300 62x302
```

```
##      2      2      2      1      2      1      1      2      1      1      1
## 62x303 62x304 62x305 62x306 62x307 62x308 62x309 62x310 62x313 62x315 62x322
##      2      1      1      1      2      1      2      1      1      1      1
## 62x324 62x327 62x332 62x337 62x338 62x351 62x355 62x359 62x363 62x370   62x5
##      1      1      2      2      2      1      2      2      2      1      1
##  62x57  62x58  62x59  62x60  62x63  62x65  62x66  62x69  62x73  62x74  62x75
##      2      2      2      1      1      1      1      2      2      1      1
##  62x77  62x83  62x87  62x91  62x92  62x94  62x96   BRAC
##      1      2      1      2      2      1      1      2
```

```
# Aggregate spectra by sample names using mean
bract_spec_avg <- aggspec(bract_spec, by = bract_samples, FUN = mean)
petal_spec_avg <- aggspec(petal_spec, by = petal_samples, FUN = mean)
labellum_spec_avg <- aggspec(labellum_spec, by = labellum_samples, FUN = mean)
```

Fix Negative Reflectance Values

Negative reflectance values are corrected by adding the minimum reflectance.

```
# Fix negative values by adding the minimum reflectance
bract_spec_avg <- procspec(bract_spec_avg, fixneg = "addmin")
```

```
## processing options applied:
## Negative value correction: added min to all reflectance
```

```
petal_spec_avg <- procspec(petal_spec_avg, fixneg = "addmin")
```

```
## processing options applied:
## Negative value correction: added min to all reflectance
```

```
labellum_spec_avg <- procspec(labellum_spec_avg, fixneg = "addmin")
```

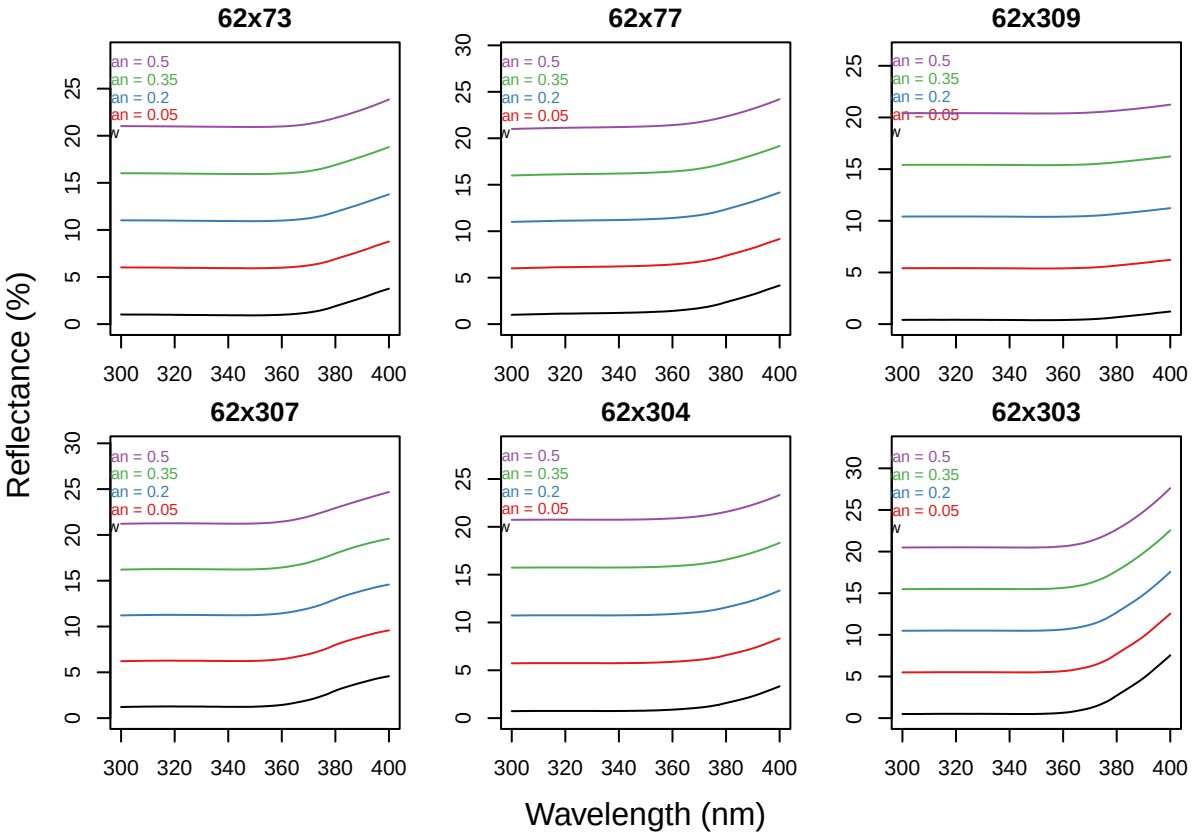
```
## processing options applied:
## Negative value correction: added min to all reflectance
```

Determine Smoothing Parameter

Use plotsmooth to visualize and decide on an appropriate smoothing span.

```
# Plot to determine suitable smoothing span
plotsmooth(bract_spec_avg[,1:7],
           minsmooth = 0.05,
           maxsmooth = 0.5,
           curves = 4,
           ask = FALSE)
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FALSE, :
## k-d tree limited by memory. ncmax= 200
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FALSE, :
## k-d tree limited by memory. ncmax= 200
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FALSE, :
## k-d tree limited by memory. ncmax= 200
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FALSE, :
## k-d tree limited by memory. ncmax= 200
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FALSE, :
## k-d tree limited by memory. ncmax= 200
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FALSE, :
## k-d tree limited by memory. ncmax= 200
```



Choose a span (e.g., 0.2) based on the plot to balance smoothness and data fidelity.

Smooth the Spectral Data

Apply smoothing to the spectral data using the chosen span.

```
# Apply smoothing with span = 0.2
bract_spec_sm <- procspec(bract_spec_avg, opt = "smooth", span = 0.2)
```

```
## processing options applied:
## smoothing spectra with a span of 0.2
```

```
petal_spec_sm <- procspec(petal_spec_avg, opt = "smooth", span = 0.2)
```

```
## processing options applied:
## smoothing spectra with a span of 0.2
```

```
labellum_spec_sm <- procspec(labellum_spec_avg, opt = "smooth", span = 0.2)
```

```
## processing options applied:
## smoothing spectra with a span of 0.2
```

Scale the Spectral Data

Scale the spectral data to different reference points for comparative analysis.

```
# Scale spectra to both minimum and maximum reflectance
bract_spec_scaleminmax_UV <- procspec(bract_spec_sm, opt = c("min", "max"))
```

```
## processing options applied:
## Scaling spectra to a minimum value of zero
```

```
## Scaling spectra to a maximum value of 1
```

```
petal_spec_scaleminmax_UV <- procspec(petal_spec_sm, opt = c("min", "max"))
```

```
## processing options applied:
## Scaling spectra to a minimum value of zero
## Scaling spectra to a maximum value of 1
```

```
labellum_spec_scaleminmax_UV <- procspec(labellum_spec_sm, opt = c("min", "max"))
```

```
## processing options applied:
## Scaling spectra to a minimum value of zero
## Scaling spectra to a maximum value of 1
```

Plot Processed Spectra

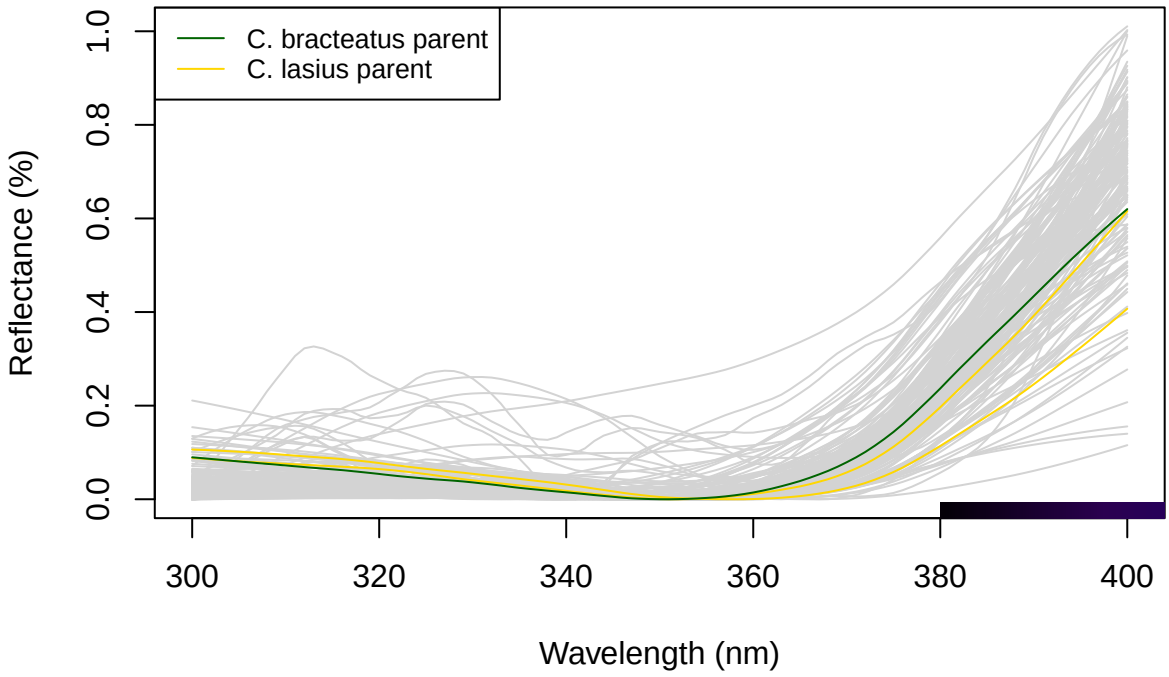
Visualize the processed spectral data for each floral tissue, highlighting specific samples.

Define Common Plotting Parameters

```
# Define a color palette for highlighting specific samples
highlight_colors <- c(rep("lightgrey", 170), "gold1", "gold1", "darkgreen")
```

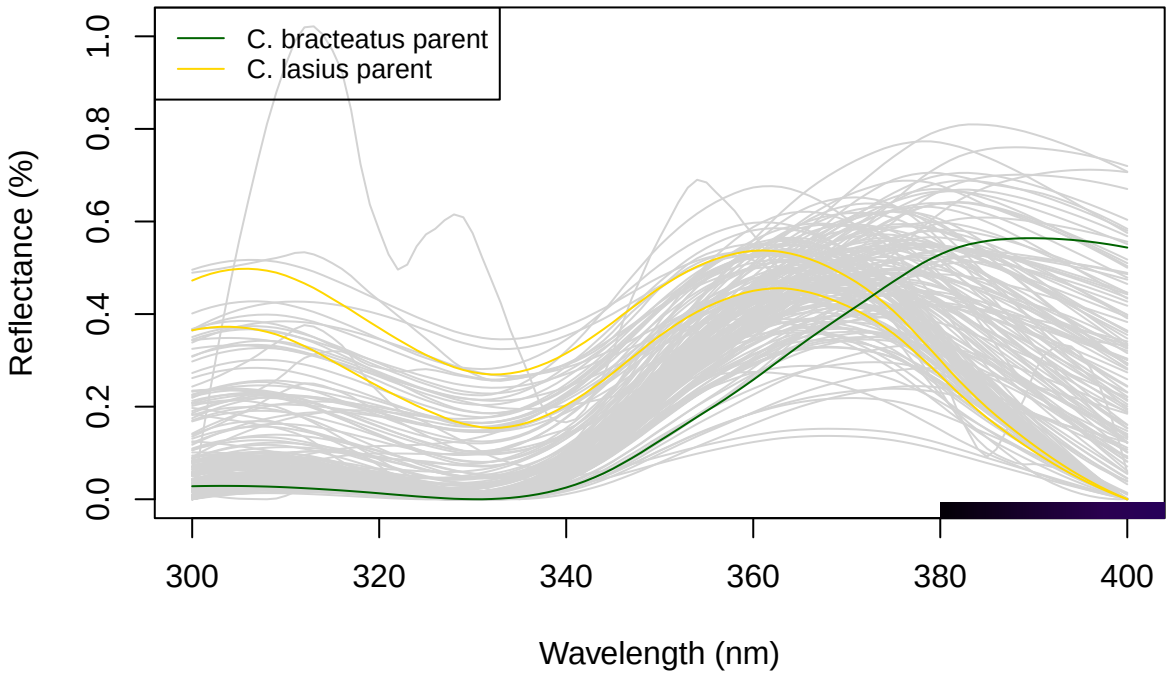
Bract Spectra Plot

```
# Order samples to highlight specific ones
order_spec_bract <- c(
  which(names(bract_spec_scaleminmax_UV) == "125"),
  which(names(bract_spec_scaleminmax_UV) == "126"),
  which(names(bract_spec_scaleminmax_UV) == "BRAC")
)

# Reorder columns to place highlighted samples at the end
columns_bract <- 1:ncol(bract_spec_scaleminmax_UV)
order_spec_bract <- c(columns_bract[columns_bract %notin% order_spec_bract], order_spec_bract)

# Plot the spectra
plot(bract_spec_scaleminmax_UV[order_spec_bract], type = "o",
     col = highlight_colors, main = "Bract Spectra",
     xlab = "Wavelength (nm)", ylab = "Reflectance (%)")

# Add a legend
legend("topleft", legend = c("C. bracteatus parent", "C. lasius parent"),
       col = c("darkgreen", "gold1"), lty = 1, cex = 0.8)
```



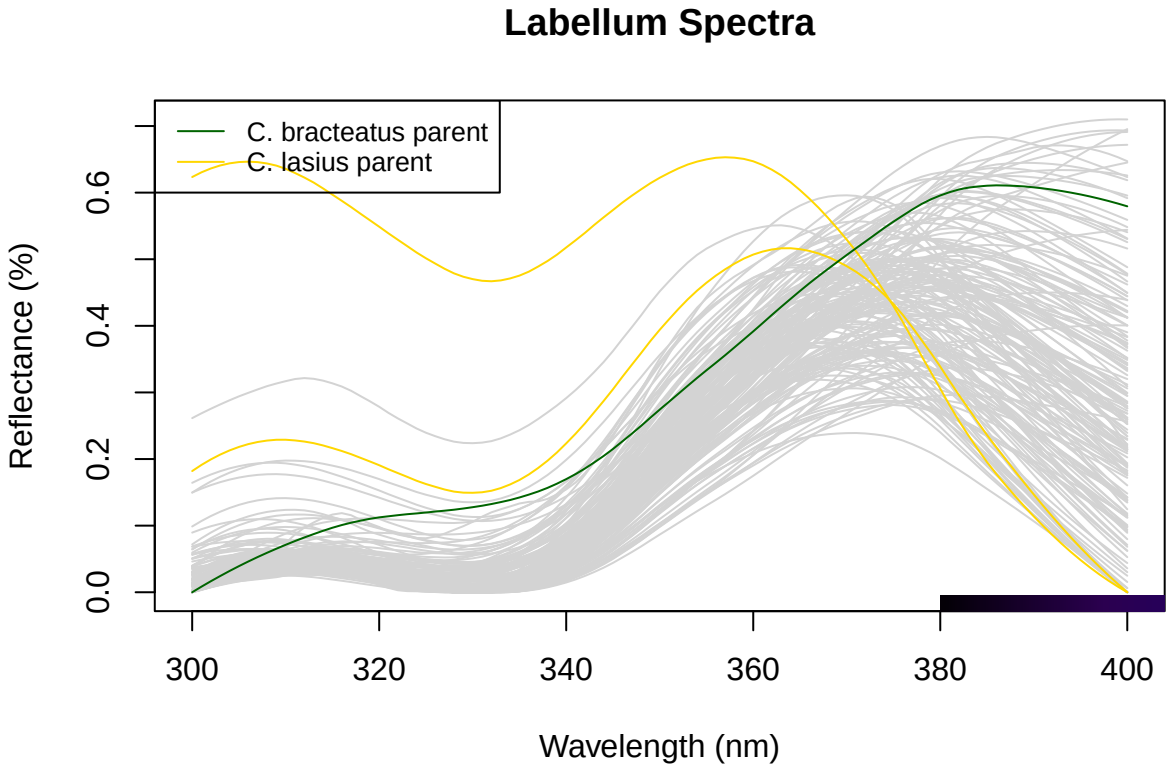
Petal Spectra Plot

```
# Order samples to highlight specific ones
order_spec_petal <- c(
  which(names(petal_spec_scaleminmax_UV) == "125"),
  which(names(petal_spec_scaleminmax_UV) == "126"),
  which(names(petal_spec_scaleminmax_UV) == "BRAC")
)

# Reorder columns to place highlighted samples at the end
columns_petal <- 1:ncol(petal_spec_scaleminmax_UV)
order_spec_petal <- c(columns_petal[columns_petal %notin% order_spec_petal], order_spec_petal)

# Plot the spectra
plot(petal_spec_scaleminmax_UV[order_spec_petal], type = "o",
     col = highlight_colors, main = "Petal Spectra",
     xlab = "Wavelength (nm)", ylab = "Reflectance (%)")

# Add a legend
legend("topleft", legend = c("C. bracteatus parent", "C. lasius parent"),
       col = c("darkgreen", "gold1"), lty = 1, cex = 0.8)
```



Labellum Spectra Plot

```
# Order samples to highlight specific ones
order_spec_labellum <- c(
  which(names(labellum_spec_scaleminmax_UV) == "125"),
  which(names(labellum_spec_scaleminmax_UV) == "126"),
  which(names(labellum_spec_scaleminmax_UV) == "BRAC")
)

# Reorder columns to place highlighted samples at the end
columns_labellum <- 1:ncol(labellum_spec_scaleminmax_UV)
order_spec_labellum <- c(columns_labellum[columns_labellum %notin% order_spec_labellum], order_spec_labe

# Plot the spectra
plot(labellum_spec_scaleminmax_UV[order_spec_labellum], type = "o",
     col = highlight_colors, main = "Labellum Spectra",
     xlab = "Wavelength (nm)", ylab = "Reflectance (%)")

# Add a legend
legend("topleft", legend = c("C. bracteatus parent", "C. lasius parent"),
       col = c("darkgreen", "gold1"), lty = 1, cex = 0.8)
```



Combine and Save Spectral Plots with Highlighted Parents

```
# Save combined spectra plots with highlighted parental samples
pdf("./results/figures/Combined_UVspectra_Bract_Petal_Labellum_Cbracteatus_Clasius_hybrids.pdf", width =
par(mfrow = c(1, 3))
# Increase left margin to provide more space for y-axis labels
par(mar = c(5, 5, 4, 2) + 0.1)  # c(bottom, left, top, right)
plot(bract_spec_scaleminmax_UV[order_spec_bract], type = "o", col = highlight_colors,
     main = "Bract spectra", xlab = "Wavelength (nm)", ylab = "Reflectance (%)",
     cex.main = 3, cex.lab = 2.5, cex.axis = 2, lwd = 2)
legend("topleft", legend = c("C. bracteatus parent", "C. lasius parent"),
       col = c("darkgreen", "gold1"), lty = 1, cex = 2.5, bty = "n")
plot(petal_spec_scaleminmax_UV[order_spec_petal], type = "o", col = highlight_colors,
     main = "Petal spectra", xlab = "Wavelength (nm)", ylab = "Reflectance (%)",
     cex.main = 3, cex.lab = 2.5, cex.axis = 2, lwd = 2)
legend("topleft", legend = c("C. bracteatus parent", "C. lasius parent"),
       col = c("darkgreen", "gold1"), lty = 1, cex = 2.5, bty = "n")
plot(labellum_spec_scaleminmax_UV[order_spec_labellum], type = "o", col = highlight_colors,
     main = "Labellum spectra", xlab = "Wavelength (nm)", ylab = "Reflectance (%)",
     cex.main = 3, cex.lab = 2.5, cex.axis = 2, lwd = 2)
legend("topleft", legend = c("C. bracteatus parent", "C. lasius parent"),
       col = c("darkgreen", "gold1"), lty = 1, cex = 2.5, bty = "n")
dev.off()
```

```
## pdf
##   2
```

Spectral Descriptors

We compute various spectral descriptors to quantify the color properties of the floral tissues.

Define spectral descriptors

B1: Total brightness B2: Mean brightness B3: Intensity (Rmax) S1U to S1R: Relative contributions of UV, Violet, Blue, Green, Yellow, and Red spectral ranges to total brightness S2: Spectral saturation (Rmax/Rmin) S3: Chroma S4: Spectral purity S5: Chroma S6: Contrast (Rmax - Rmin) S7: Spectral saturation S8: Chroma ((Rmax - Rmin)/B2) S9: Carotenoid chroma ((R700 - R450)/R700) S10: Peaky chroma H1 to H5: Hue metrics (e.g., peak wavelength) Note: Some metrics may be sensitive to spectral noise.

Calculate Spectral Descriptors

Bract

```
# Calculate spectral descriptors for bract
summary_bract_UV <- summary(bract_spec_scaleminmax_UV)
```

```
## Warning: cannot calculate violet chroma; wavelength below 415 nm
```

```
## Warning: cannot calculate blue chroma; wavelength range not between 400 and 510
## nm
```

```
## Warning: cannot calculate green chroma; wavelength range not between 510 and
## 605 nm
```

```
## Warning: cannot calculate yellow chroma; wavelength range not between 550 and
## 625 nm
```

```
## Warning: cannot calculate red chroma; wavelength range not between 605 and 700
## nm
```

```
# Remove S2 due to infinite values
summary_bract_UV <- summary_bract_UV %>% select(-S2)

# Extract metrics for parents
bract125 <- round(summary_bract_UV["125", ], 3)
bract126 <- round(summary_bract_UV["126", ], 3)
bractBRAC <- round(summary_bract_UV["BRAC", ], 3)
```

Plot Bract Spectral Descriptors Histograms

```
pdf("./results/figures/bract_UV_descriptors_histograms.pdf", width = 14, height = 10)

# Set up plotting area: 4 rows x 6 columns for histograms
par(mfrow = c(4, 6),        # 4 rows, 6 columns
    mar = c(5, 5, 4, 2) + 0.1,  # Margins for each plot: bottom, left, top, right
    oma = c(0, 0, 0, 5))    # Outer margins: bottom, left, top, right

# Loop through each metric and plot histogram with parental lines
for (i in 1:ncol(summary_bract_UV)) {

  if (!any(!is.na(summary_bract_UV[, i]))) next  # Skip columns that are entirely NA

  hist(summary_bract_UV[, i],
       xlab = colnames(summary_bract_UV)[i],
```

```
       main = colnames(summary_bract_UV)[i],
       col = "lightgrey",
       border = "white")

  # Add vertical lines for parental samples
  abline(v = bract125[i], col = 'gold1', lwd = 3)      # C. lasius parent
  abline(v = bract126[i], col = 'gold1', lwd = 3)      # C. lasius parent
  abline(v = bractBRAC[i], col = 'darkgreen', lwd = 3) # C. bracteatus parent
}

# Allow drawing in the outer margin
par(xpd = TRUE)

plot.new()

# Add a shared legend in the outer right margin
legend("topright",
       inset = c(0, 0),  # Adjusts the position of the legend
       legend = c("C. lasius parent", "C. bracteatus parent"),
       col = c("gold1", "darkgreen"),
       lty = 1,              # Line type: solid
       lwd = 3,              # Line width
       cex = 1.5,            # Text size
       bty = "n")            # No box around the legend

# Close the PDF device to save the file
dev.off()
```

```
## pdf
##   2
```

Petal Summary

```
# Calculate summary statistics for petal
summary_petal_UV <- summary(petal_spec_scaleminmax_UV)
```

```
## Warning: cannot calculate violet chroma; wavelength below 415 nm
```

```
## Warning: cannot calculate blue chroma; wavelength range not between 400 and 510
## nm
```

```
## Warning: cannot calculate green chroma; wavelength range not between 510 and
## 605 nm
```

```
## Warning: cannot calculate yellow chroma; wavelength range not between 550 and
## 625 nm
```

```
## Warning: cannot calculate red chroma; wavelength range not between 605 and 700
## nm
```

```
# Remove S2 due to infinite values
summary_petal_UV <- summary_petal_UV %>% select(-S2)

# Extract metrics for specific samples
petal125 <- round(summary_petal_UV["125", ], 3)
petal126 <- round(summary_petal_UV["126", ], 3)
```

```r
petalBRAC <- round(summary_petal_UV["BRAC", ], 3)
```

Plot Petal Spectral Descriptors Histograms

```r
pdf("./results/figures/petal_UV_descriptors_histograms.pdf", width = 14, height = 10)

# Set up plotting area: 4 rows x 6 columns for histograms
par(mfrow = c(4, 6),        # 4 rows, 6 columns
    mar = c(5, 5, 4, 2) + 0.1,  # Margins for each plot: bottom, left, top, right
    oma = c(0, 0, 0, 5))    # Outer margins: bottom, left, top, right

# Loop through each metric and plot histogram with parental lines
for (i in 1:ncol(summary_petal_UV)) {

  if (!any(!is.na(summary_petal_UV[, i]))) next  # Skip columns that are entirely NA

  hist(summary_petal_UV[, i],
       xlab = colnames(summary_petal_UV)[i],
       main = colnames(summary_petal_UV)[i],
       col = "lightgrey",
       border = "white")

  # Add vertical lines for parental samples
  abline(v = petal125[i], col = 'gold1', lwd = 3)      # C. lasius parent
  abline(v = petal126[i], col = 'gold1', lwd = 3)      # C. lasius parent
  abline(v = petalBRAC[i], col = 'darkgreen', lwd = 3) # C. bracteatus parent
}

# Allow drawing in the outer margin
par(xpd = TRUE)

plot.new()

# Add a shared legend in the outer right margin
legend("topright",
       inset = c(0, 0),  # Adjusts the position of the legend
       legend = c("C. lasius parent", "C. bracteatus parent"),
       col = c("gold1", "darkgreen"),
       lty = 1,              # Line type: solid
       lwd = 3,              # Line width
       cex = 1.5,            # Text size
       bty = "n")            # No box around the legend

# Close the PDF device to save the file
dev.off()
```

```
## pdf
##   2
```

Labellum Summary

```r
# Calculate summary statistics for labellum
summary_labellum_UV <- summary(labellum_spec_scaleminmax_UV)
```

```
## Warning: cannot calculate violet chroma; wavelength below 415 nm

## Warning: cannot calculate blue chroma; wavelength range not between 400 and 510
## nm

## Warning: cannot calculate green chroma; wavelength range not between 510 and
## 605 nm

## Warning: cannot calculate yellow chroma; wavelength range not between 550 and
## 625 nm

## Warning: cannot calculate red chroma; wavelength range not between 605 and 700
## nm
```

```r
# Remove S2 due to infinite values
summary_labellum_UV <- summary_labellum_UV %>% select(-S2)

# Extract metrics for specific samples
labellum125 <- round(summary_labellum_UV["125", ], 3)
labellum126 <- round(summary_labellum_UV["126", ], 3)
labellumBRAC <- round(summary_labellum_UV["BRAC", ], 3)
```

Plot Labellum Spectral Descriptors Histograms

```r
pdf("./results/figures/labellum_UV_descriptors_histograms.pdf", width = 14, height = 10)

# Set up plotting area: 4 rows x 6 columns for histograms
par(mfrow = c(4, 6),        # 4 rows, 6 columns
    mar = c(5, 5, 4, 2) + 0.1,  # Margins for each plot: bottom, left, top, right
    oma = c(0, 0, 0, 5))    # Outer margins: bottom, left, top, right

# Loop through each metric and plot histogram with parental lines
for (i in 1:ncol(summary_labellum_UV)) {

  if (!any(!is.na(summary_labellum_UV[, i]))) next  # Skip columns that are entirely NA

  hist(summary_labellum_UV[, i],
       xlab = colnames(summary_labellum_UV)[i],
       main = colnames(summary_labellum_UV)[i],
       col = "lightgrey",
       border = "white")

  # Add vertical lines for parental samples
  abline(v = labellum125[i], col = 'gold1', lwd = 3)      # C. lasius parent
  abline(v = labellum126[i], col = 'gold1', lwd = 3)      # C. lasius parent
  abline(v = labellumBRAC[i], col = 'darkgreen', lwd = 3) # C. bracteatus parent
}

# Allow drawing in the outer margin
par(xpd = TRUE)

plot.new()

# Add a shared legend in the outer right margin
legend("topright",
       inset = c(0, 0),  # Adjusts the position of the legend
```

```
       legend = c("C. lasius parent", "C. bracteatus parent"),
       col = c("gold1", "darkgreen"),
       lty = 1,               # Line type: solid
       lwd = 3,               # Line width
       cex = 1.5,             # Text size
       bty = "n")             # No box around the legend

# Close the PDF device to save the file
dev.off()
```

```
## pdf
##   2
```

Data Formatting and Export

Reformat Rownames

Replace 'x' with '_' in row names for consistency.

```
# Replace 'x' with '_' in row names
rownames(summary_bract_UV) <- gsub("x", "_", rownames(summary_bract_UV))
rownames(summary_petal_UV) <- gsub("x", "_", rownames(summary_petal_UV))
rownames(summary_labellum_UV) <- gsub("x", "_", rownames(summary_labellum_UV))
```

Convert Rownames to a Column

Add the row names as a new column id to facilitate merging.

```
# Convert row names to a column named 'id'
summary_petal_UV <- summary_petal_UV %>% rownames_to_column(var = "id")
summary_labellum_UV <- summary_labellum_UV %>% rownames_to_column(var = "id")
```

Subset and Rename Columns

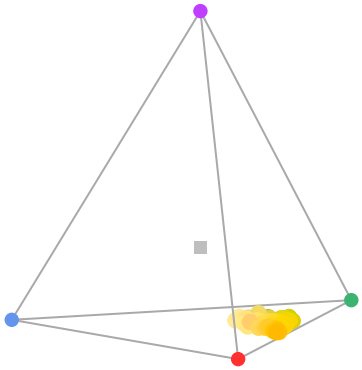
Select relevant metrics and rename columns to indicate their corresponding floral tissue.

```
# Subset relevant columns based on analysis needs
summary_petal_UV <- summary_petal_UV[, c("id", "H1")]
summary_labellum_UV <- summary_labellum_UV[, c("id", "H1")]

# Rename columns to include tissue type
colnames(summary_petal_UV) <- c("id", "H1_UV_petal")
colnames(summary_labellum_UV) <- c("id", "H1_UV_labellum")
```

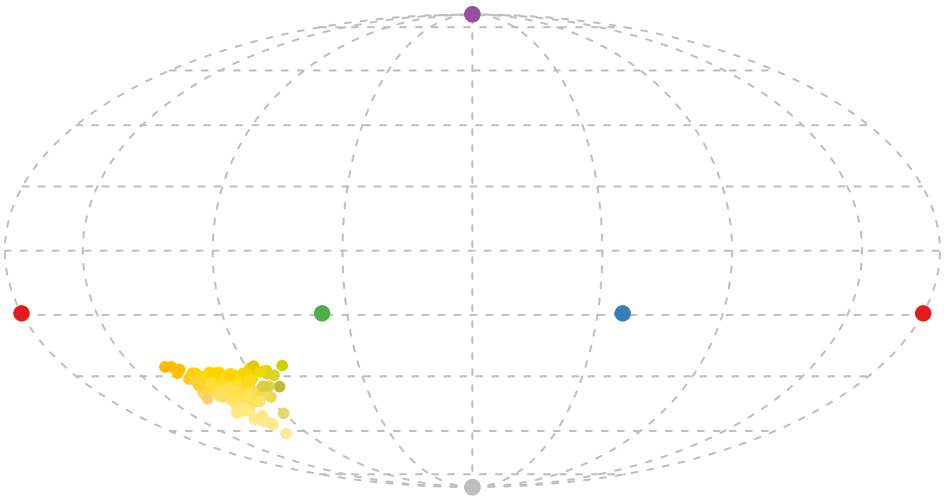
Vision Models

```
# Compute Photoreceptor Quantum Catches for Birds and Bees
## Bract
bract_bird_vis <- vismodel(bract_spec_scaleminmax, visual = "avg.uv", relative = TRUE)
bract_bee_vis <- vismodel(bract_spec_scaleminmax, visual = "apis", relative = TRUE)

## Petal
petal_bird_vis <- vismodel(petal_spec_scaleminmax, visual = "avg.uv", relative = TRUE)
petal_bee_vis <- vismodel(petal_spec_scaleminmax, visual = "apis", relative = TRUE)
```

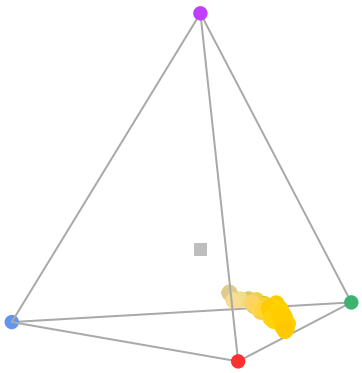
```
## Labellum
labellum_bird_vis <- vismodel(labellum_spec_scaleminmax, visual = "avg.uv", relative = TRUE)
labellum_bee_vis <- vismodel(labellum_spec_scaleminmax, visual = "apis", relative = TRUE)

# Convert Bird Vision Data into a Color Space
bract_bird_colspace <- colspace(bract_bird_vis, space = "auto")
petal_bird_colspace <- colspace(petal_bird_vis, space = "auto")
labellum_bird_colspace <- colspace(labellum_bird_vis, space = "auto")

# Plot the Avian Color Space Representation
plot(bract_bird_colspace, col = spec2rgb(bract_spec_scaleminmax))
```



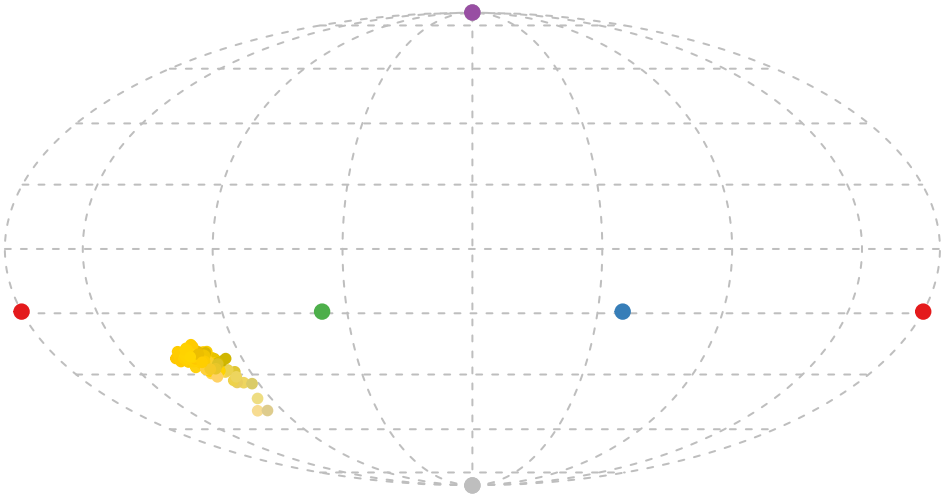
```
projplot(bract_bird_colspace, pch = 20, col = spec2rgb(bract_spec_scaleminmax))
```



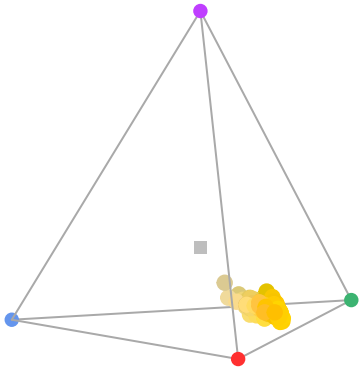
```
plot(petal_bird_colspace, col = spec2rgb(petal_spec_scaleminmax))
```



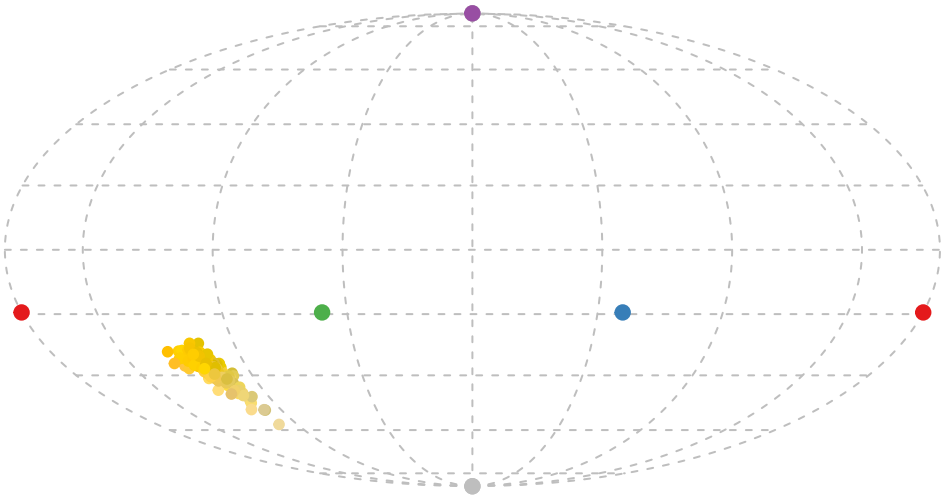
```
projplot(petal_bird_colspace, pch = 20, col = spec2rgb(petal_spec_scaleminmax))
```



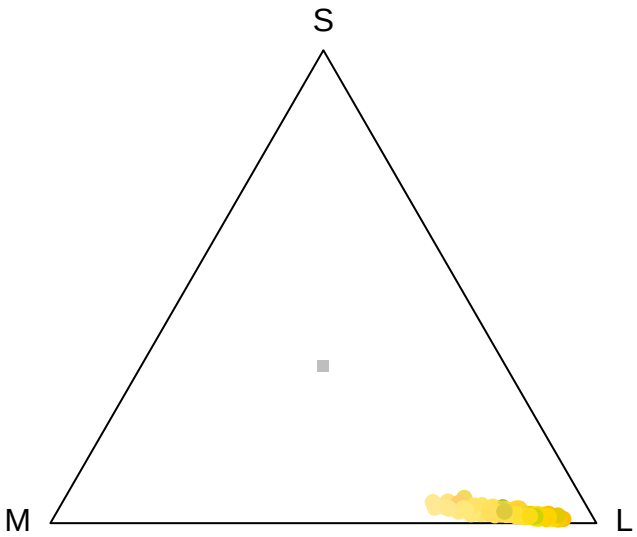
```
plot(labellum_bird_colspace, col = spec2rgb(labellum_spec_scaleminmax))
```



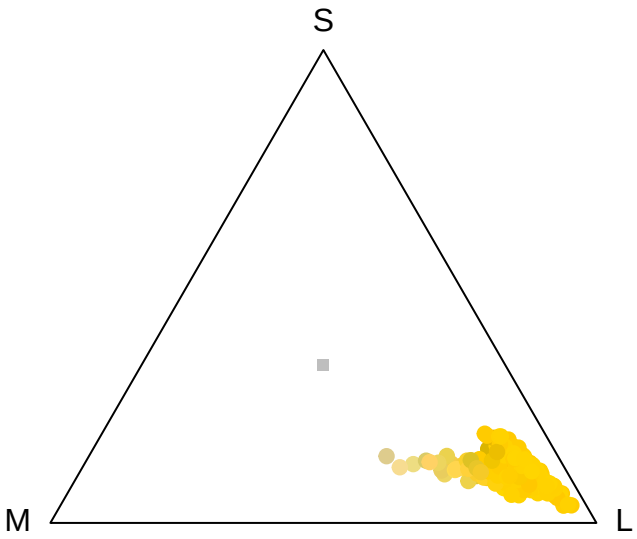
```
projplot(labellum_bird_colspace, pch = 20, col = spec2rgb(labellum_spec_scaleminmax))
```



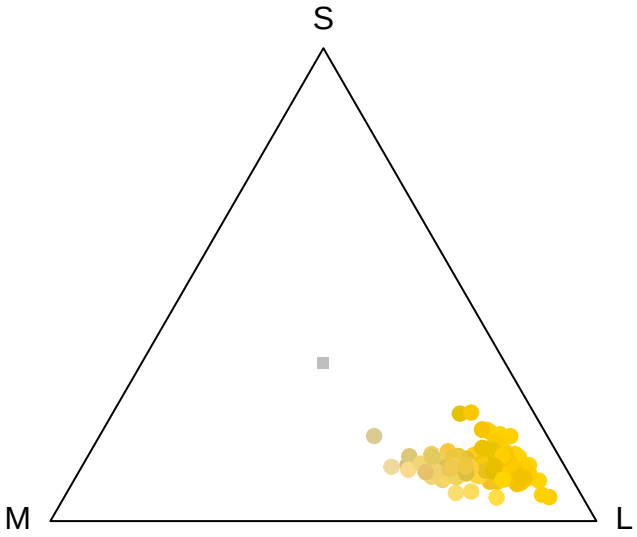
```
# Convert Bee Vision Data into a Color Space
bract_bee_colspace <- colspace(bract_bee_vis, space = "tri")
petal_bee_colspace <- colspace(petal_bee_vis, space = "tri")
labellum_bee_colspace <- colspace(labellum_bee_vis, space = "tri")

# Plot the Bee Color Space Representation
plot(bract_bee_colspace, col = spec2rgb(bract_spec_scaleminmax))
```




```
plot(petal_bee_colspace, col = spec2rgb(petal_spec_scaleminmax))
```




```
plot(labellum_bee_colspace, col = spec2rgb(labellum_spec_scaleminmax))
```




```
# Replace 'x' with '_' in row names
rownames(bract_bird_colspace) <- gsub("x", "_", rownames(bract_bird_colspace))
rownames(bract_bee_colspace) <- gsub("x", "_", rownames(bract_bee_colspace))

rownames(petal_bird_colspace) <- gsub("x", "_", rownames(petal_bird_colspace))
rownames(petal_bee_colspace) <- gsub("x", "_", rownames(petal_bee_colspace))

rownames(labellum_bird_colspace) <- gsub("x", "_", rownames(labellum_bird_colspace))
rownames(labellum_bee_colspace) <- gsub("x", "_", rownames(labellum_bee_colspace))

# Convert row names to a column named 'id'
bract_bird_colspace <- bract_bird_colspace %>% rownames_to_column(var = "id")
bract_bee_colspace <- bract_bee_colspace %>% rownames_to_column(var = "id")

petal_bird_colspace <- petal_bird_colspace %>% rownames_to_column(var = "id")
petal_bee_colspace <- petal_bee_colspace %>% rownames_to_column(var = "id")

labellum_bird_colspace <- labellum_bird_colspace %>% rownames_to_column(var = "id")
```

```r
labellum_bee_colspace <- labellum_bee_colspace %>% rownames_to_column(var = "id")

# Subset relevant columns based on analysis needs
bract_bird_colspace <- bract_bird_colspace[, c("id", "h.theta", "r.vec")]
bract_bee_colspace <- bract_bee_colspace[, c("id", "h.theta", "r.vec")]

petal_bird_colspace <- petal_bird_colspace[, c("id", "h.theta", "r.vec")]
petal_bee_colspace <- petal_bee_colspace[, c("id", "h.theta", "r.vec")]

labellum_bird_colspace <- labellum_bird_colspace[, c("id", "h.theta", "r.vec")]
labellum_bee_colspace <- labellum_bee_colspace[, c("id", "h.theta", "r.vec")]

# Rename columns to include tissue type
colnames(bract_bird_colspace) <- c("id", "h.theta.bird.bract", "r.vec.bird.bract")
colnames(bract_bee_colspace) <- c("id", "h.theta.bee.bract", "r.vec.bee.bract")

colnames(petal_bird_colspace) <- c("id", "h.theta.bird.petal", "r.vec.bird.petal")
colnames(petal_bee_colspace) <- c("id", "h.theta.bee.petal", "r.vec.bee.petal")

colnames(labellum_bird_colspace) <- c("id", "h.theta.bird.labellum", "r.vec.bird.labellum")
colnames(labellum_bee_colspace) <- c("id", "h.theta.bee.labellum", "r.vec.bee.labellum")
```

Merge Dataframes

```r
# Merge all summaries by 'id'
joined_df <- pca_bract %>%
  full_join(pca_petal, by = "id") %>%
  full_join(pca_labellum, by = "id") %>%
  full_join(summary_bract, by = "id") %>%
  full_join(summary_petal, by = "id") %>%
  full_join(summary_labellum, by = "id") %>%
  full_join(summary_petal_UV, by = "id") %>%
  full_join(summary_labellum_UV, by = "id") %>%
  full_join(bract_bird_colspace, by = "id") %>%
  full_join(bract_bee_colspace, by = "id") %>%
  full_join(petal_bird_colspace, by = "id") %>%
  full_join(petal_bee_colspace, by = "id") %>%
  full_join(labellum_bird_colspace, by = "id") %>%
  full_join(labellum_bee_colspace, by = "id")
```

Export Summary Descriptors to CSV

Save the combined summary descriptors to a CSV file for further analysis or reporting.

```r
# Write the combined summary dataframe to a CSV file
write.csv(joined_df, "./results/processed_data/color_traits.csv",
          quote = FALSE, row.names = FALSE)
```