phenotypes

2023-10-10

Plotting the distributions of each trait

K. Uckele October 13, 2023

Load phenotypic data

```r
morphometric <- read_csv(
  "~/Dropbox/Costus/costus-genetic-mapping/phenotype/results/processed_data/phenotypic_data_no_outliers
  col_names = TRUE)

morphometric <- morphometric %>%
  mutate(VEFN = factor(VEFN, levels = c("0", "1"),
                           labels = c("Absent", "Present")))

morphometric <- morphometric %>%
  mutate(VNG = factor(VNG, levels = c("0", "1"),
                           labels = c("Absent", "Present")))

color <- read_csv(
  "~/Dropbox/Costus/costus-genetic-mapping/phenotype/results/processed_data/spectral_shape_descriptors.
```

Make new variables: plant_type

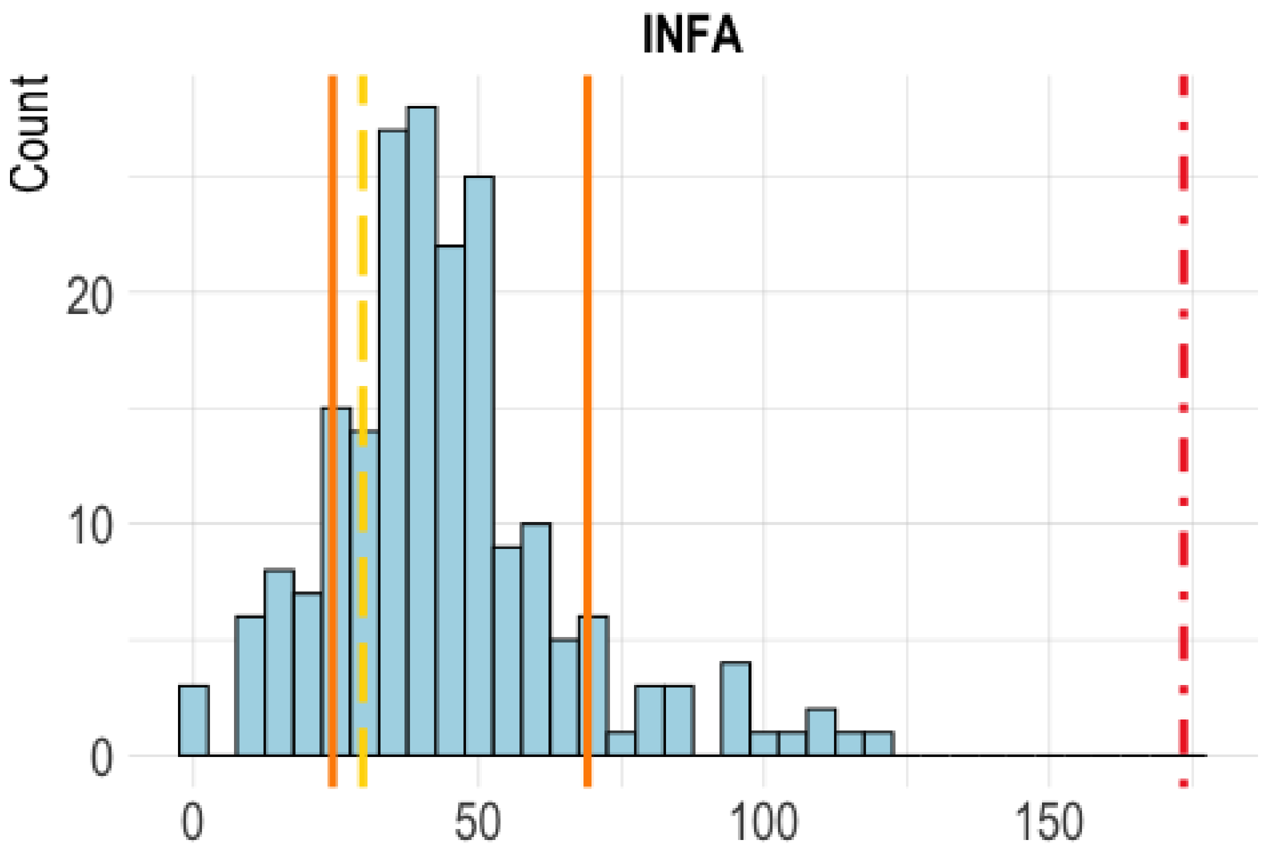
```r
morphometric$plant_type <- substr(morphometric$unique_ID, 1, 2)
color$plant_type <- substr(color$id, 1, 2)
```

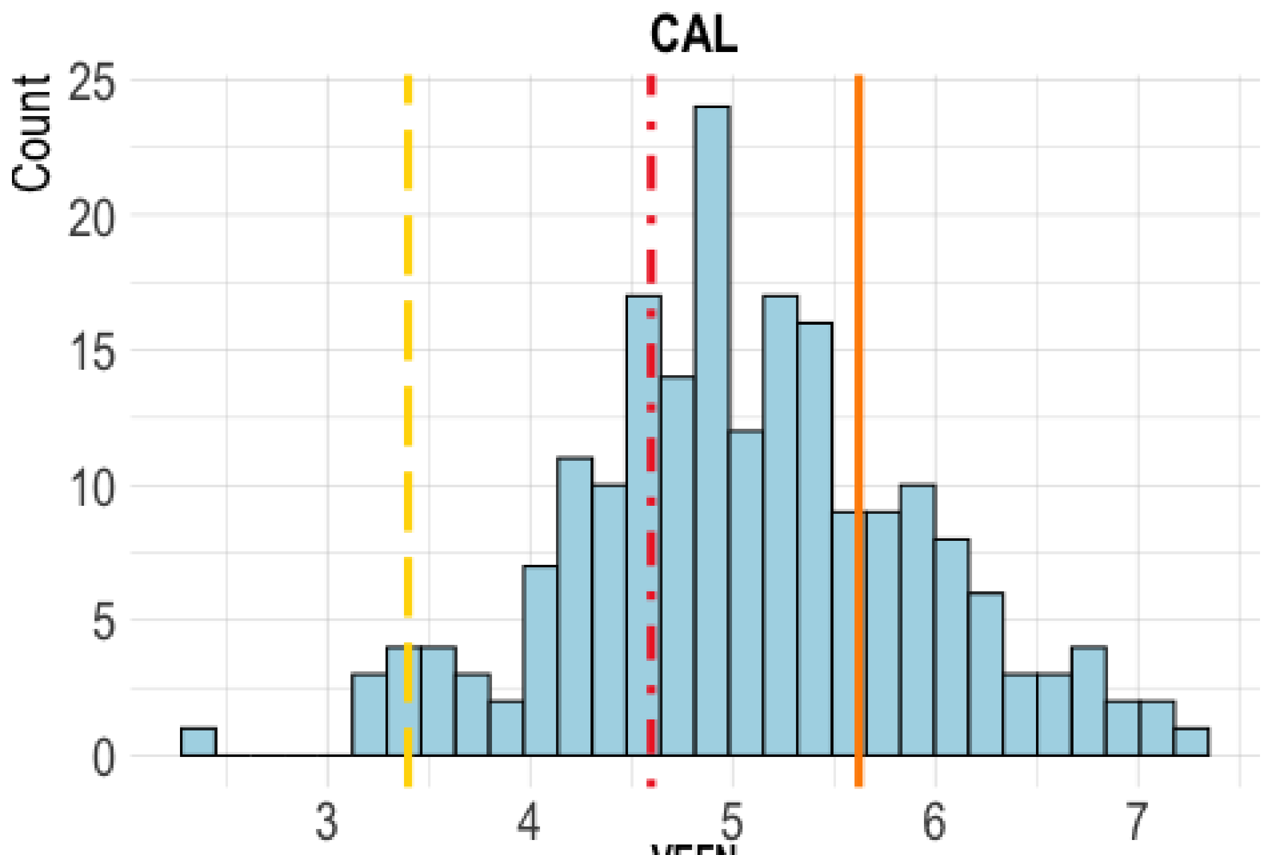
Plots

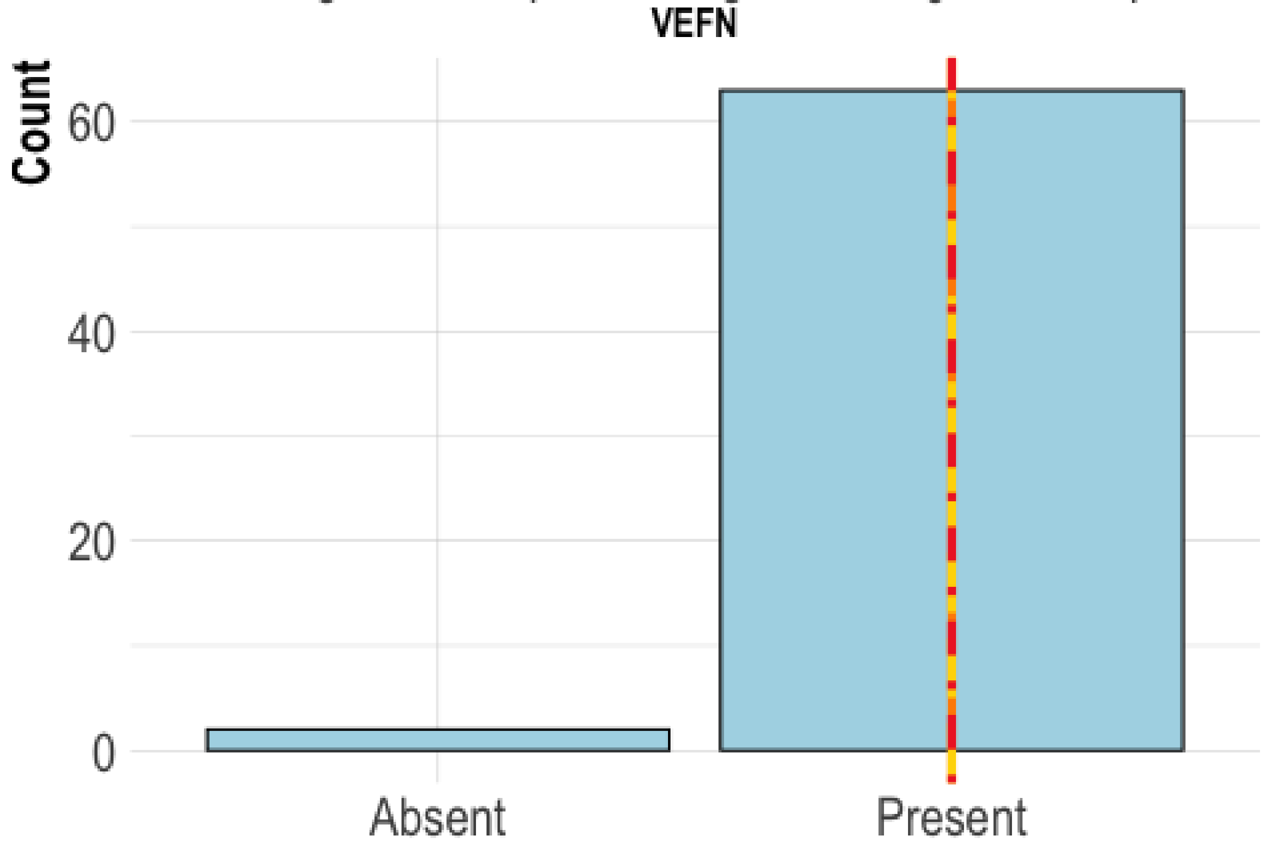
Morphometric histogram plots

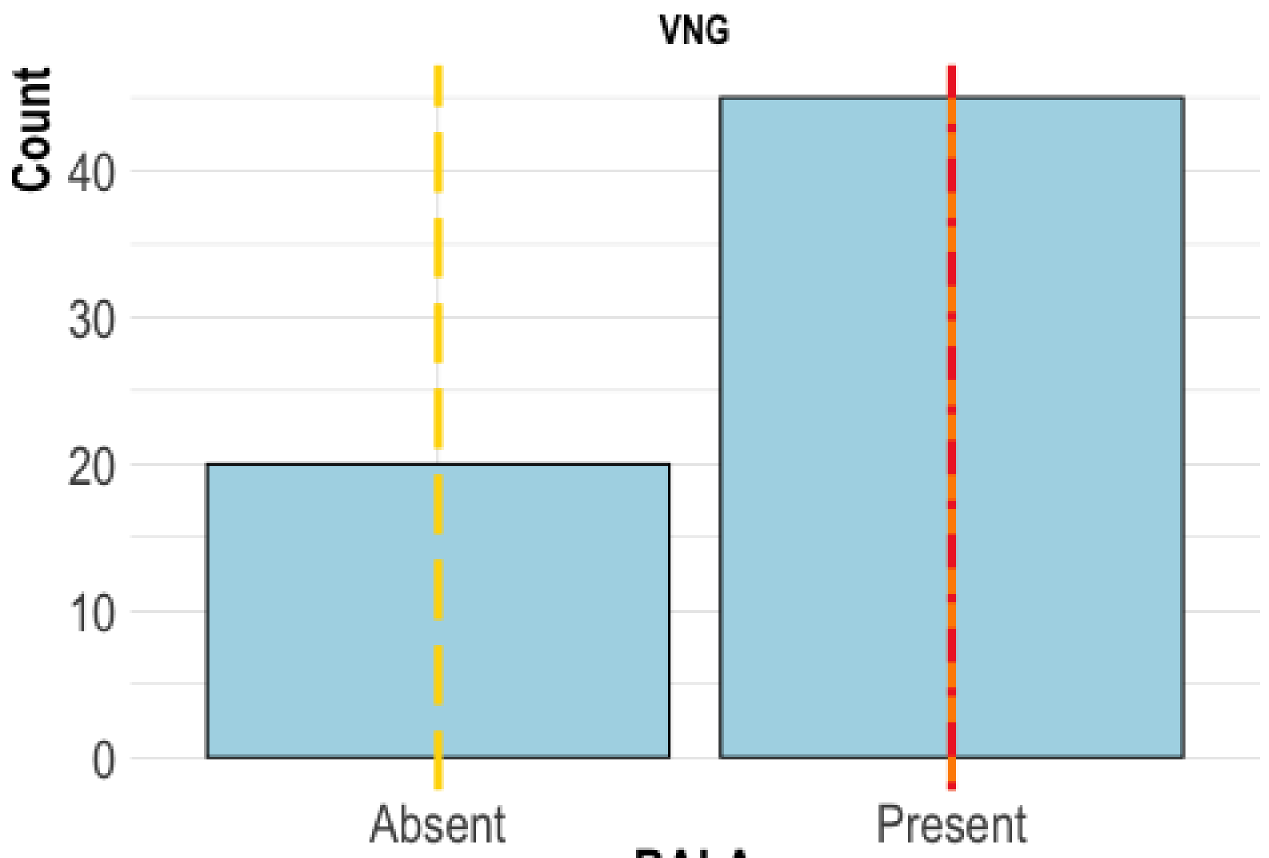
```r
# 1. Define the list of traits you want to plot
# Exclude non-numeric columns if necessary
traits <- c("INFA","CAL","VEFN","VNG","RALA","RAST","COL","COLL","STAE","TUA",
            "STATL","LABL","LABW","CLL","STAL","STAW","ANL","ANW","STIW","STYL",
            "VFN","FNSC","EFNSC40")

# 2. Specify binwidths for each trait
binwidths <- list(
  INFA = 5,
  CAL = NULL,
  VEFN = NULL,
  VNG = NULL,
  RALA = NULL,
  RAST = NULL,
  COL = NULL,
  COLL = NULL,
  STAE = NULL,
  TUA = NULL,
  STATL = NULL,
  LABL = NULL,
  LABW = NULL,
  CLL = NULL,
  STAL = NULL,
  STAW = NULL,
  ANL = NULL,
  ANW = NULL,
  STIW = NULL,
  STYL = NULL,
  VFN = NULL,
  FNSC = NULL,
  EFNSC40 = NULL
)

# Ensure all traits have a specified binwidth
# If some traits are missing, you can set a default binwidth
default_binwidth <- 1
for(trait in traits){
  if(!trait %in% names(binwidths)){
    binwidths[[trait]] <- default_binwidth
    warning(paste("Binwidth for trait", trait, "not specified. Using default binwidth =", default_binwi
  }
}

# 3. Define Unique IDs and Plant Types
unique_ids <- c("F1_39", "F1_62", "P_125", "P_126", "P_950")
plant_types <- c("39", "62") #

# 4. Create a Function to Generate Histogram for a Single Trait
create_histogram <- function(trait, binwidth){

  # Extract averages for the current trait
  F1_39avg <- as.numeric(morphometric[morphometric$unique_ID == "F1_39", trait])
  F1_62avg <- as.numeric(morphometric[morphometric$unique_ID == "F1_62", trait])
  P_125avg <- as.numeric(morphometric[morphometric$unique_ID == "P_125", trait])
  P_126avg <- as.numeric(morphometric[morphometric$unique_ID == "P_126", trait])
  P_950avg <- as.numeric(morphometric[morphometric$unique_ID == "P_950", trait])

  # Calculate mean of P_125avg and P_126avg
  P_mean_avg <- mean(c(P_125avg, P_126avg), na.rm = TRUE)

  # Create the histogram if the trait is numeric:
  if(is.numeric(morphometric[[trait]])){

    hist_plot <- ggplot(morphometric[morphometric$plant_type %in% plant_types, ], aes_string(x = trait))
    geom_histogram(binwidth = binwidth, color = "#000000", fill = "lightblue") +
    ylab("Count") +
    geom_vline(xintercept = F1_39avg, linetype = "solid", color = "darkorange", size = 1.5) +
    geom_vline(xintercept = F1_62avg, linetype = "solid", color = "darkorange", size = 1.5) +
    geom_vline(xintercept = P_mean_avg, linetype = "longdash", color = "gold", size = 1.5) +
    geom_vline(xintercept = P_950avg, linetype = "dotdash", color = "firebrick2", size = 1.5) +
    labs(title = trait) +
    theme_ipsum(base_size = 20) +
    theme(
      plot.title = element_textbox(hjust = 0.5, margin = margin(t = 5, b = 5), size = 20),
      axis.title.x = element_blank(),
      axis.title.y = element_text(
    size = 20,         # Increased y-axis title font size
    color = "black"   # Set y-axis title color to black
  ),
      plot.margin = unit(c(0.1, 0, 0.1, 0), "cm")
    )
  }

  # Create the histogram if the trait is a factor
  if(is.factor(morphometric[[trait]])){

    # Using na.omit() to remove rows with NA in the 'trait' column
    morphometric_filtered <- morphometric %>%
    filter(plant_type %in% plant_types) %>%
    na.omit(select(., all_of(trait)))  # Removes rows where 'trait' is NA

    hist_plot <- ggplot(morphometric_filtered[morphometric_filtered$plant_type %in% plant_types, ], aes
    geom_bar(color = "#000000", fill = "lightblue") +  # Changed from geom_histogram() to geom_bar()
    ylab("Count") +
    geom_vline(xintercept = F1_39avg, linetype = "solid", color = "darkorange", size = 1.5) +
    geom_vline(xintercept = F1_62avg, linetype = "solid", color = "darkorange", size = 1.5) +
    geom_vline(xintercept = P_mean_avg, linetype = "longdash", color = "gold", size = 1.5) +
    geom_vline(xintercept = P_950avg, linetype = "dotdash", color = "firebrick2", size = 1.5) +
    labs(title = trait) +
    theme_ipsum(base_size = 20) +
    theme(
      plot.title = element_textbox(hjust = 0.5, margin = margin(t = 5, b = 5), size = 15),
      axis.title.x = element_blank(),
      axis.title.y = element_text(                            # Added y-axis title customization
        size = 20,                                            # Increased font size
        face = "bold",                                        # Made the text bold
        color = "black",                                      # Set text color to black
        angle = 90,                                           # Ensure the y-axis title is vertical
        vjust = 0.5                                           # Vertically center the y-axis title
      ),
      plot.margin = unit(c(0.1, 0, 0.1, 0), "cm")
    )

  }
  return(hist_plot)
}

# 5. Loop Through Each Trait and Generate Histograms
histograms <- list()
```

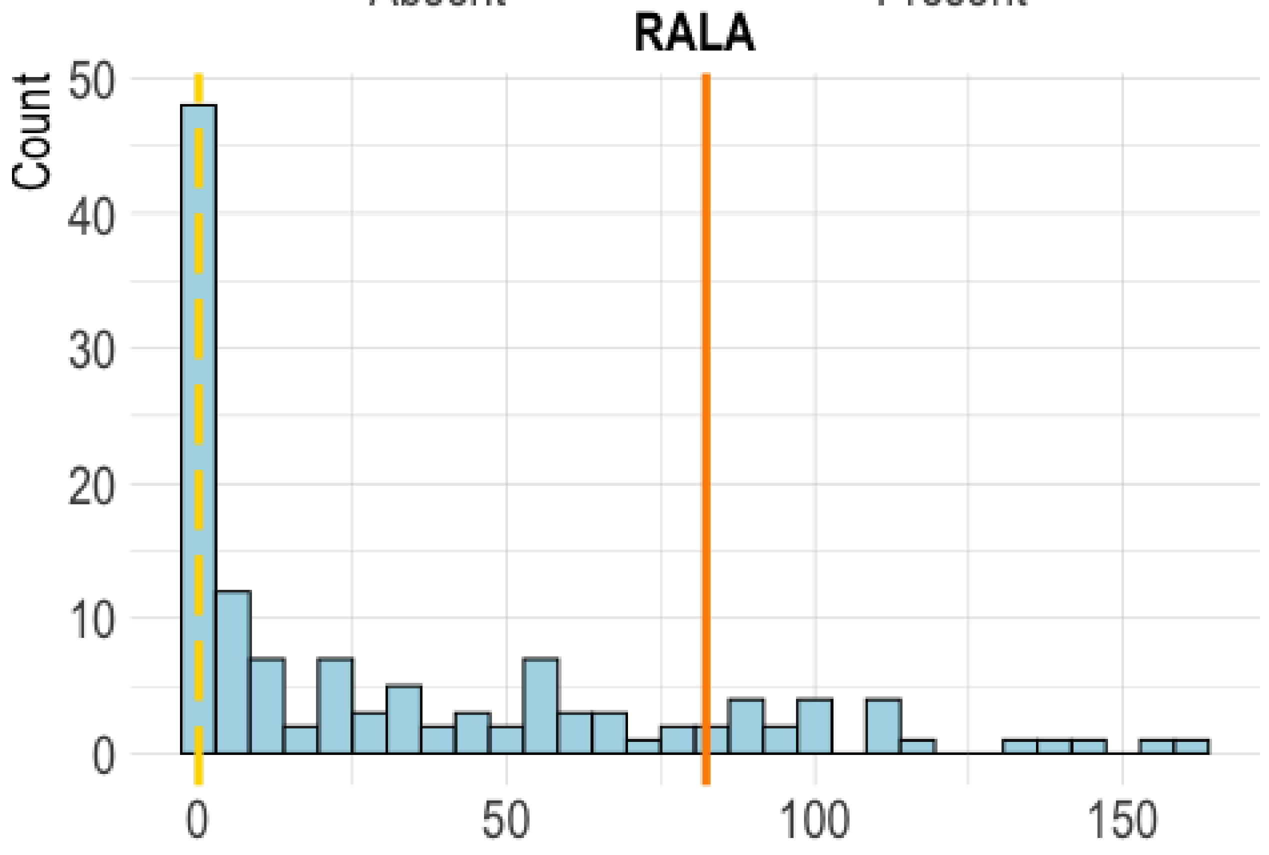
```r
for(trait in traits){

  # Retrieve the binwidth for the current trait
  binwidth <- binwidths[[trait]]

  # Create the histogram
  plot <- create_histogram(trait, binwidth)

  # Store the plot in the list
  histograms[[trait]] <- plot

  # Display the plot
  print(plot)
}
```

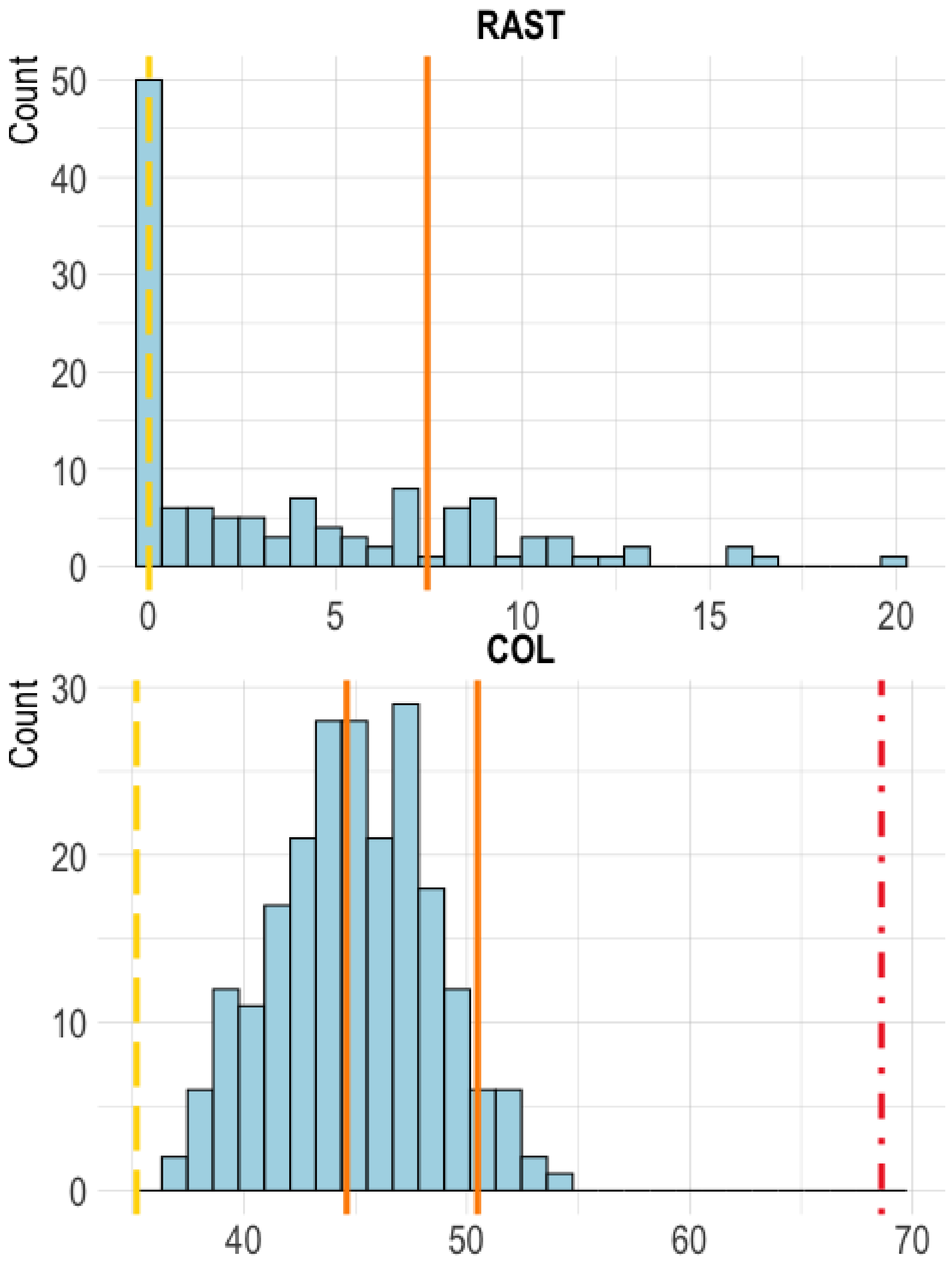


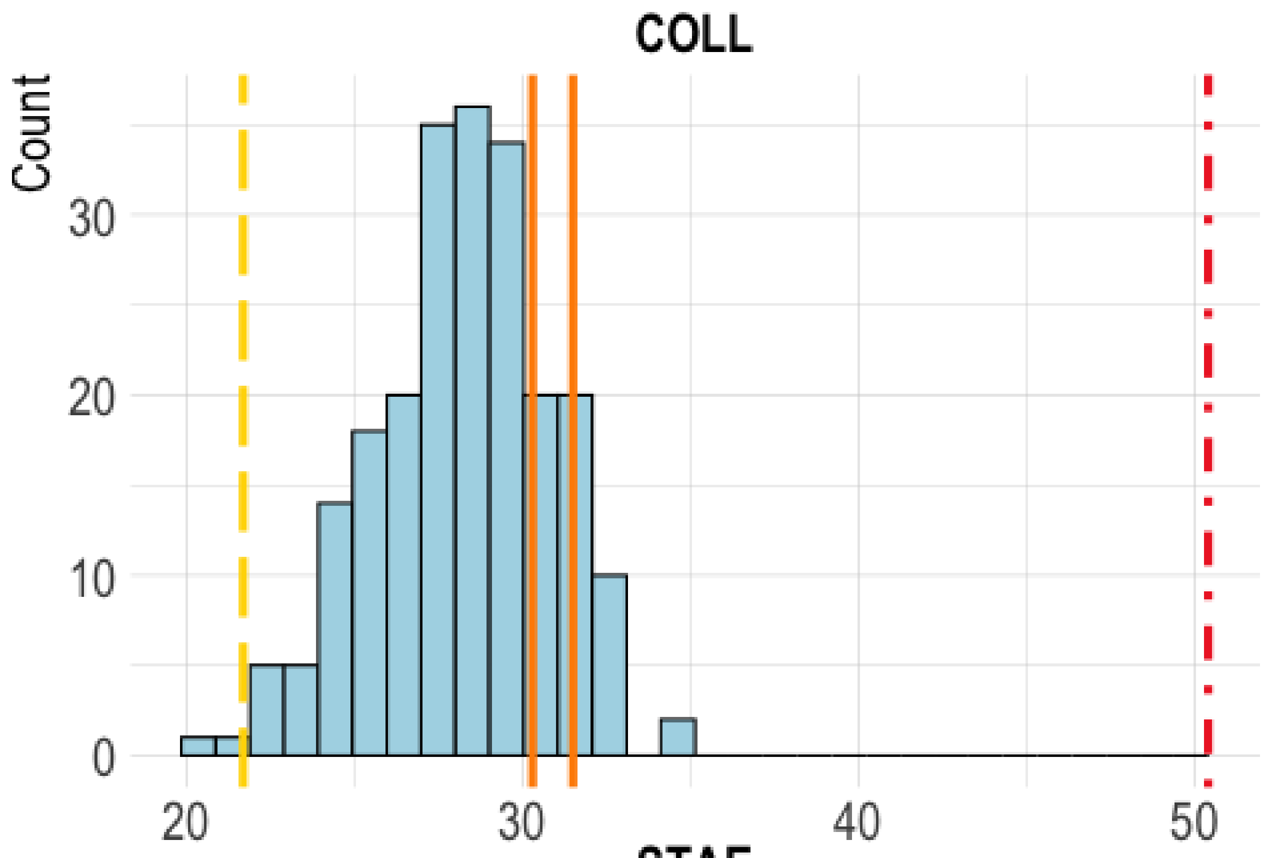

CAL
Count
25
20
15
10
5
0
3
4
5
6
7

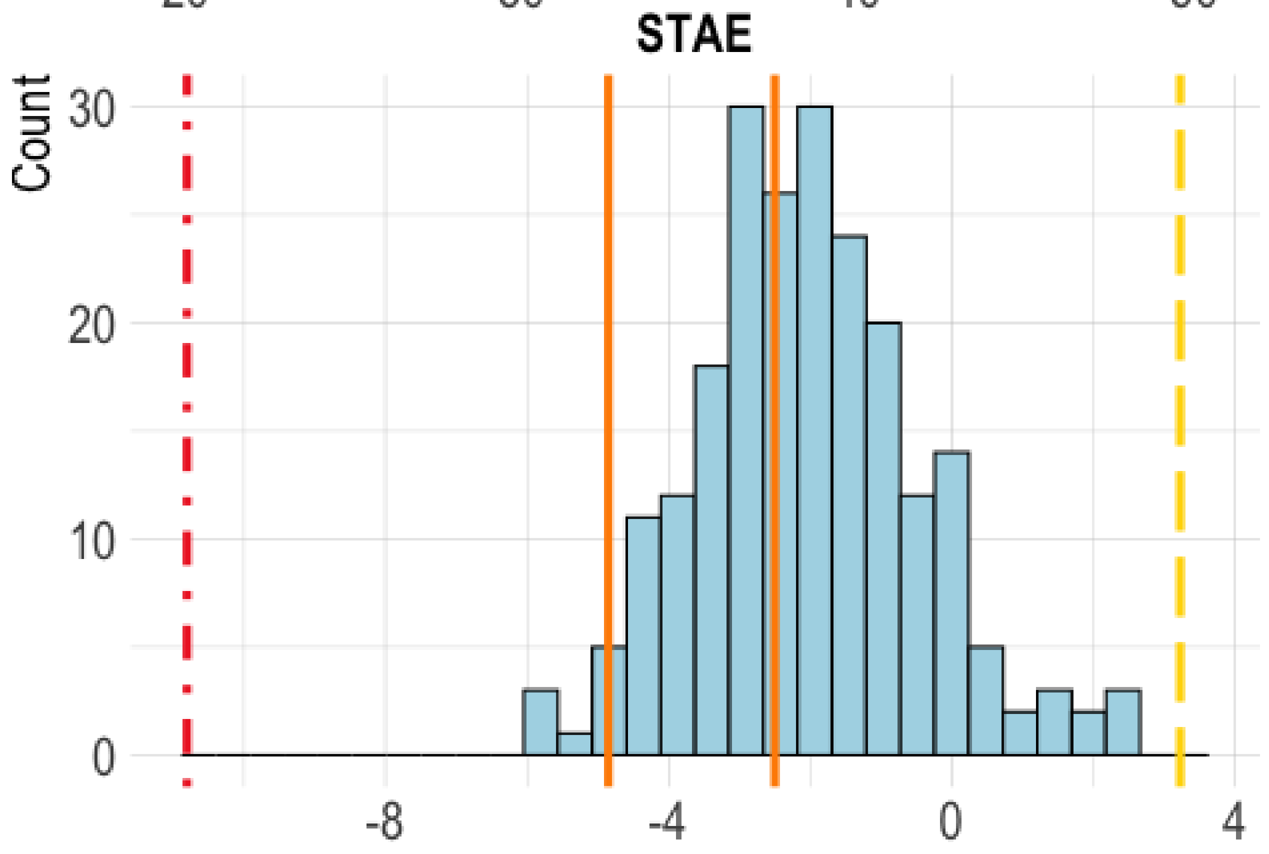

VEFN
Count
60
40
20
0
Absent
Present

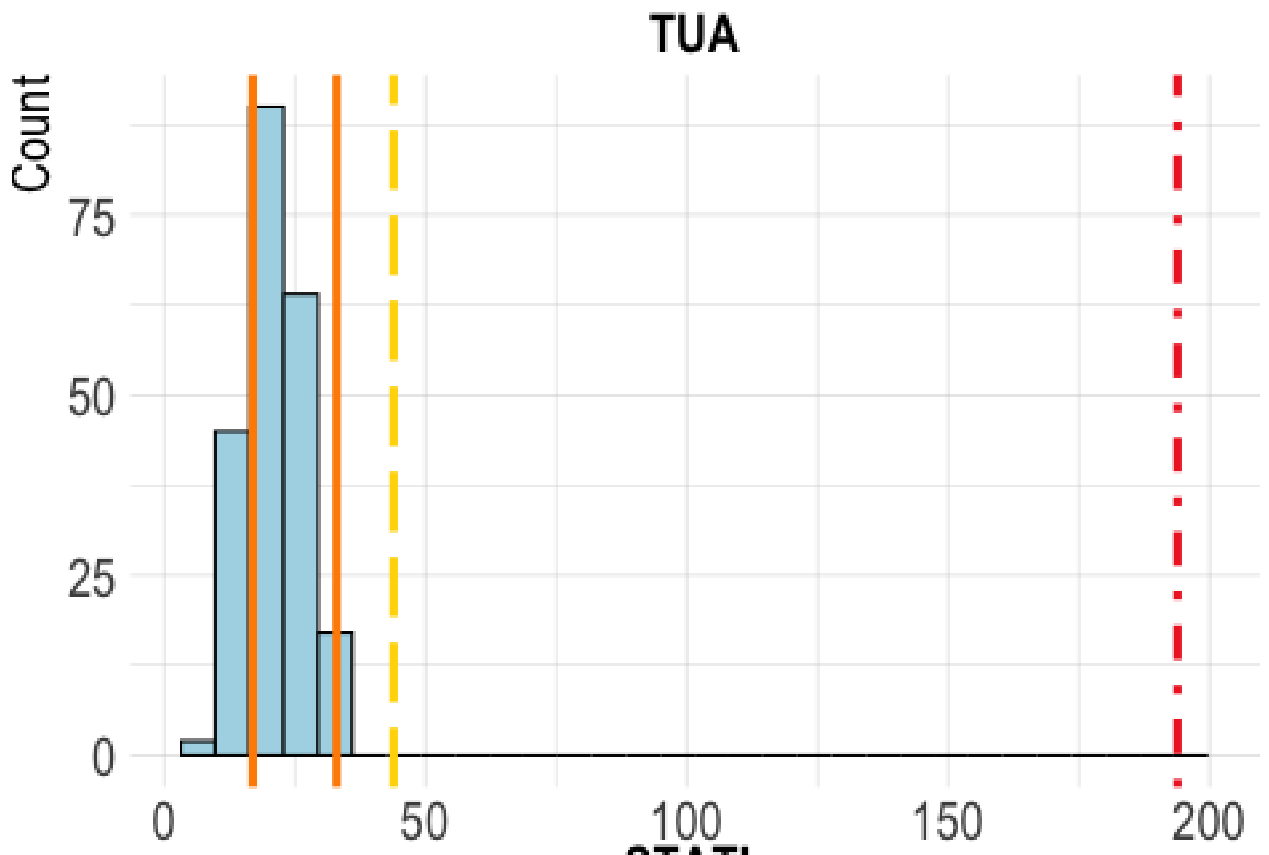

VNG
Count
0
10
20
30
40
Absent
Present

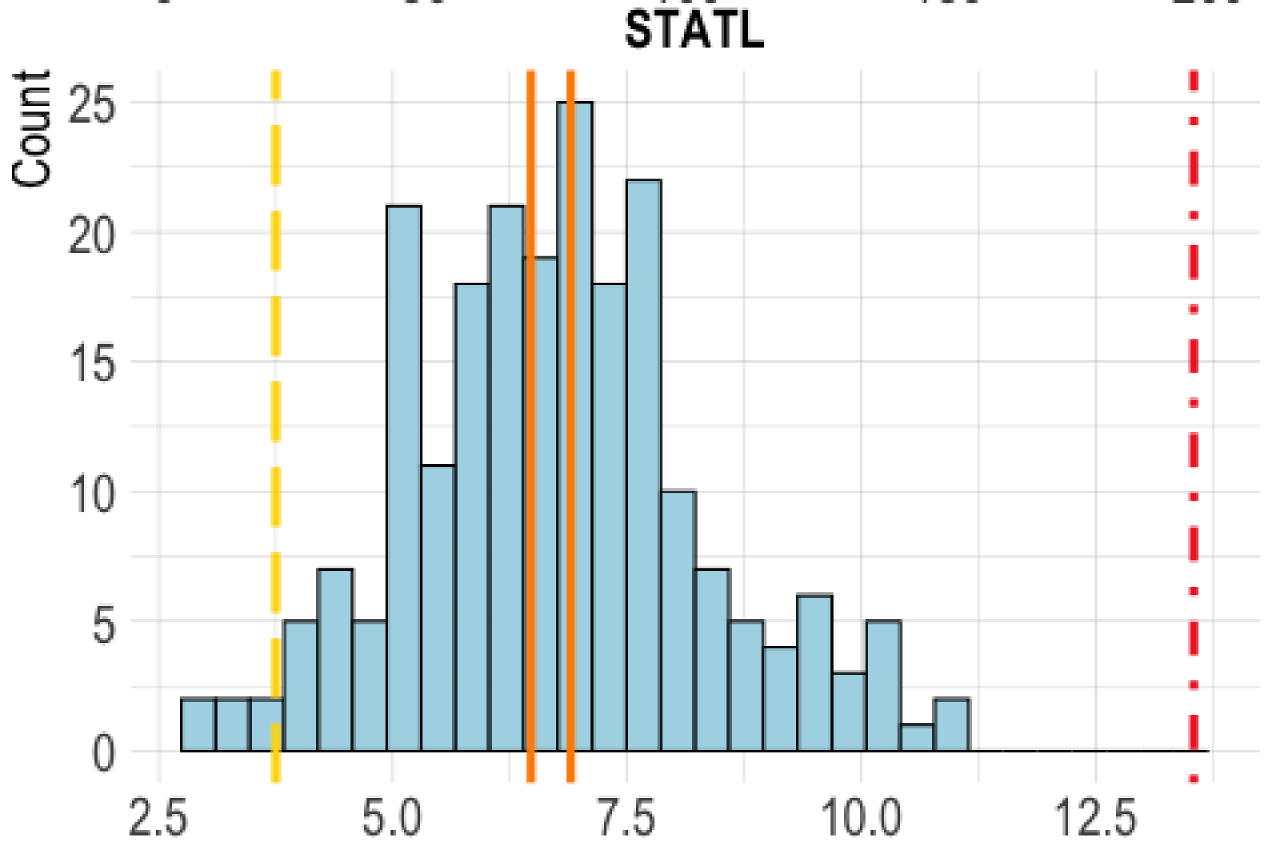

RALA
Count
0
10
20
30
40
50
0
50
100
150

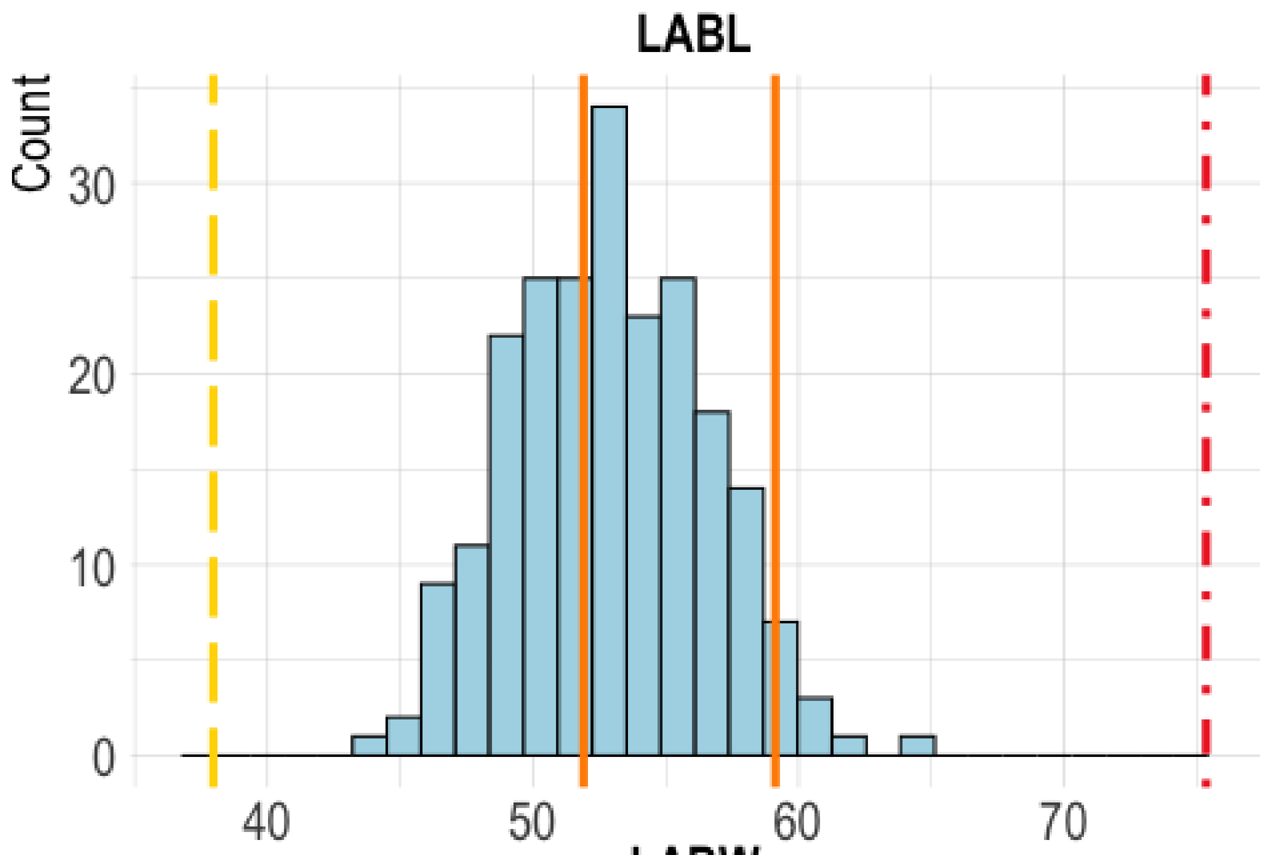

RAST
Count
50
40
30
20
10
0
0
5
10
15
20
COL
Count
30
20
10
0
40
50
60
70

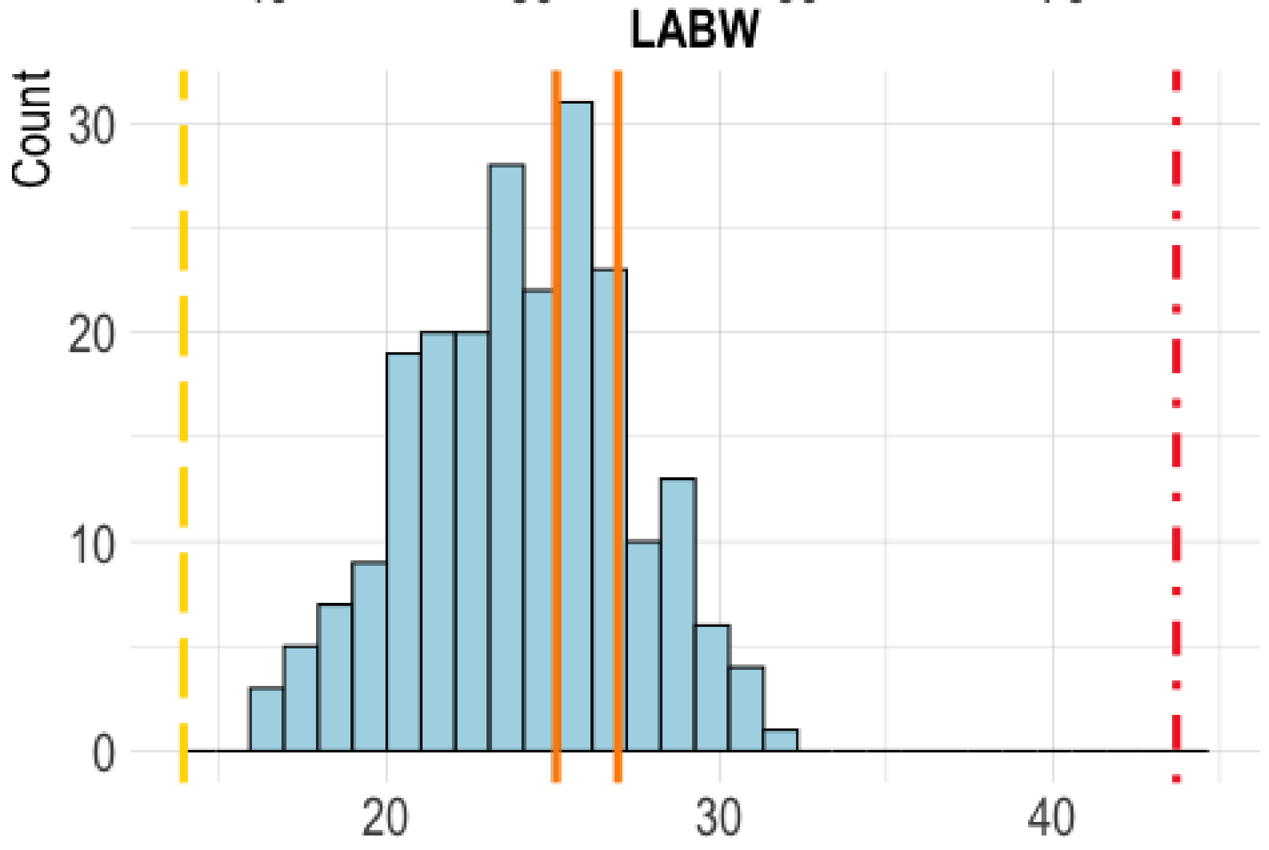

COLL
Count
0
10
20
30
20
30
40
50

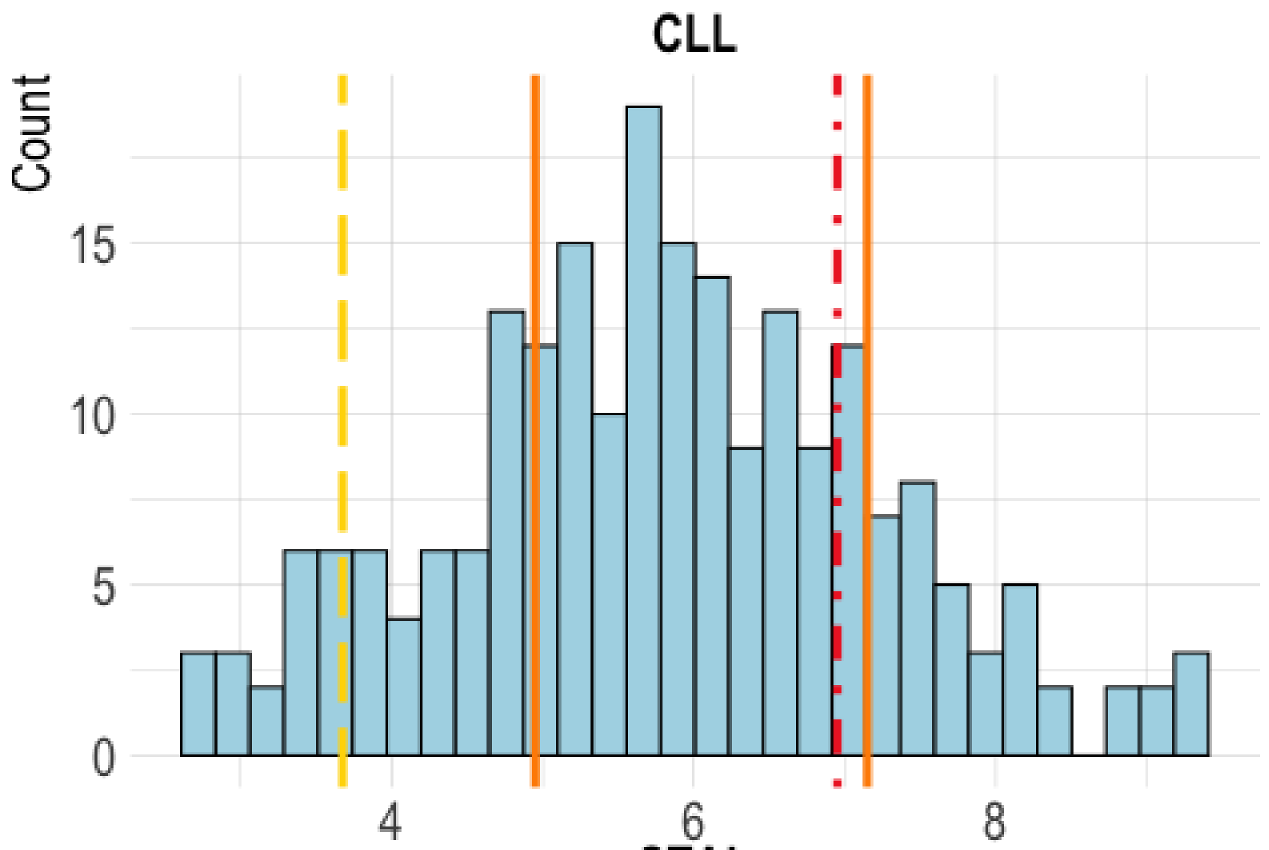

STAE
Count
0
10
20
30
-8
-4
0
4

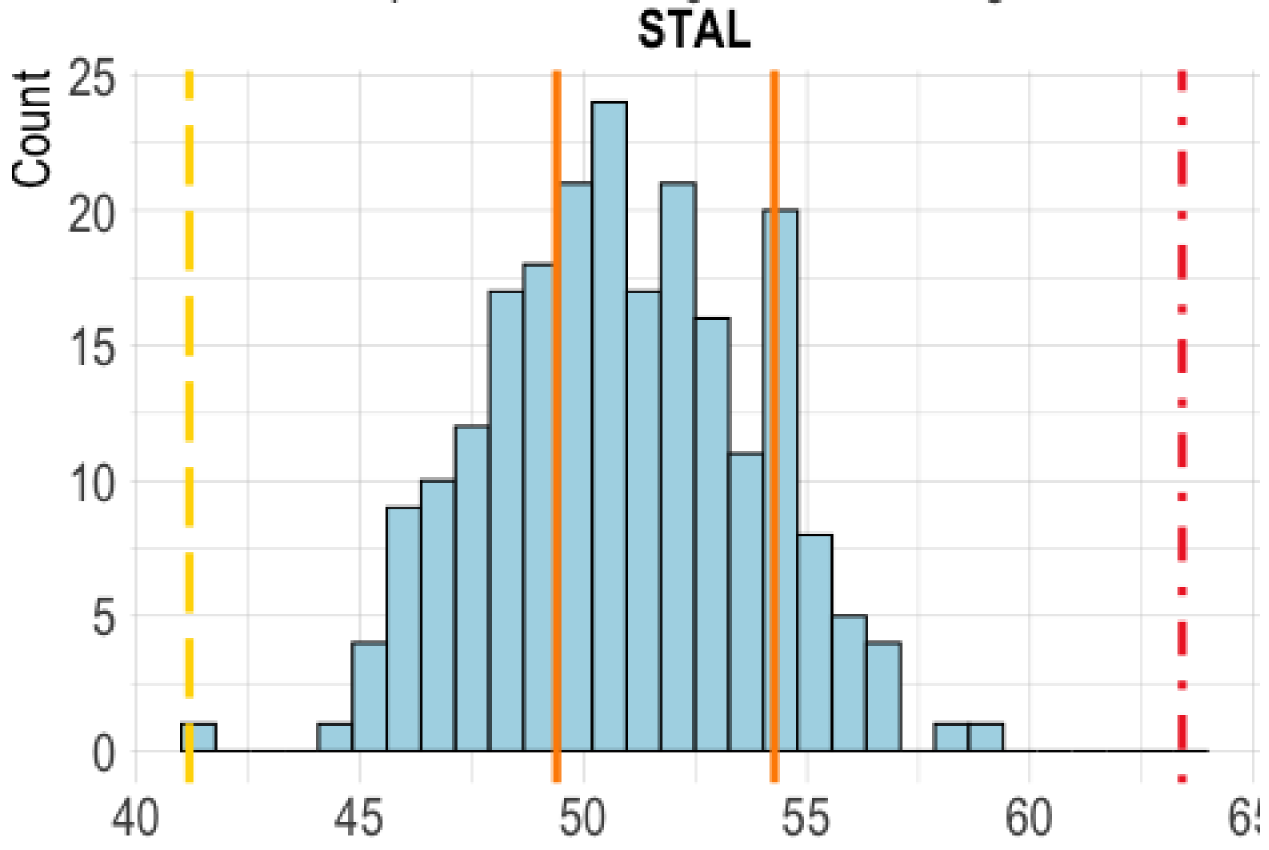

TUA
Count
0
25
50
75
0
50
100
150
200

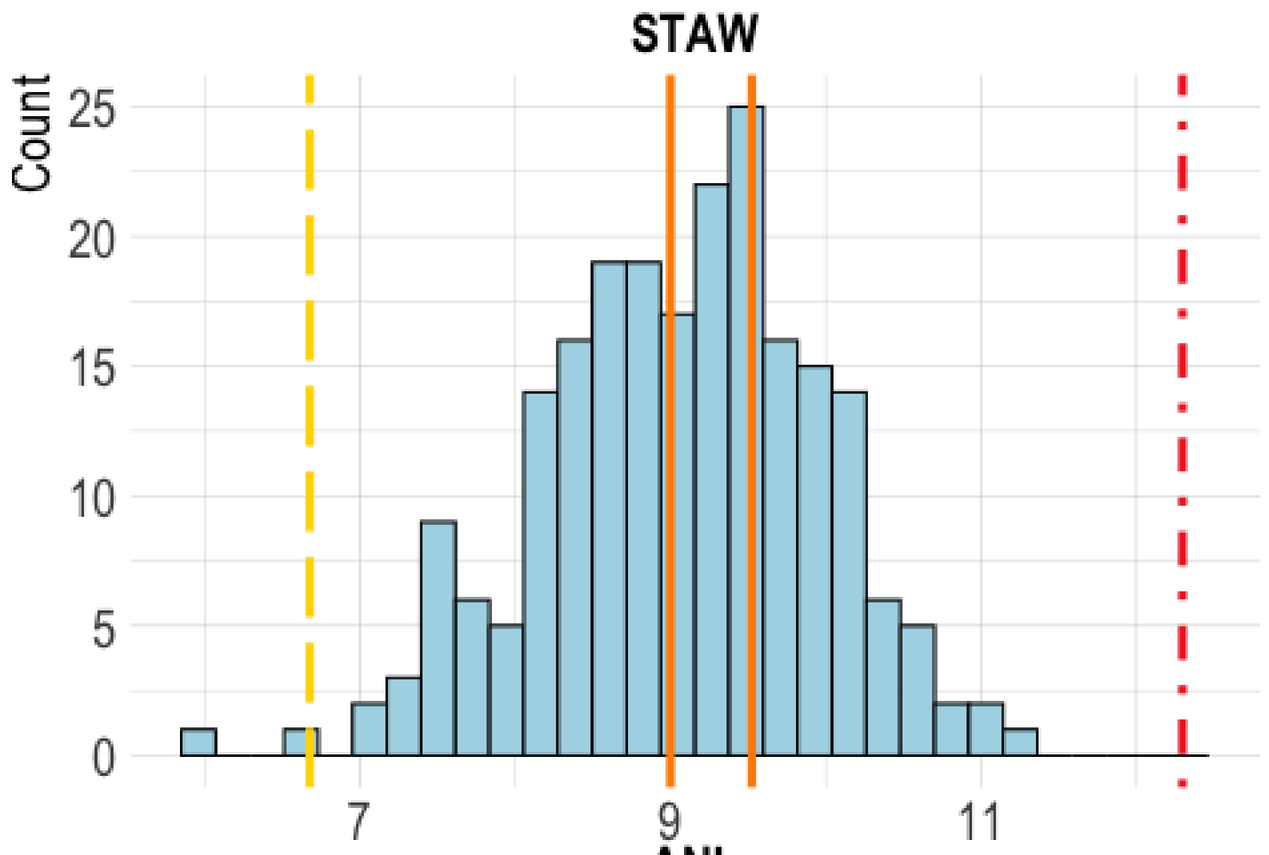
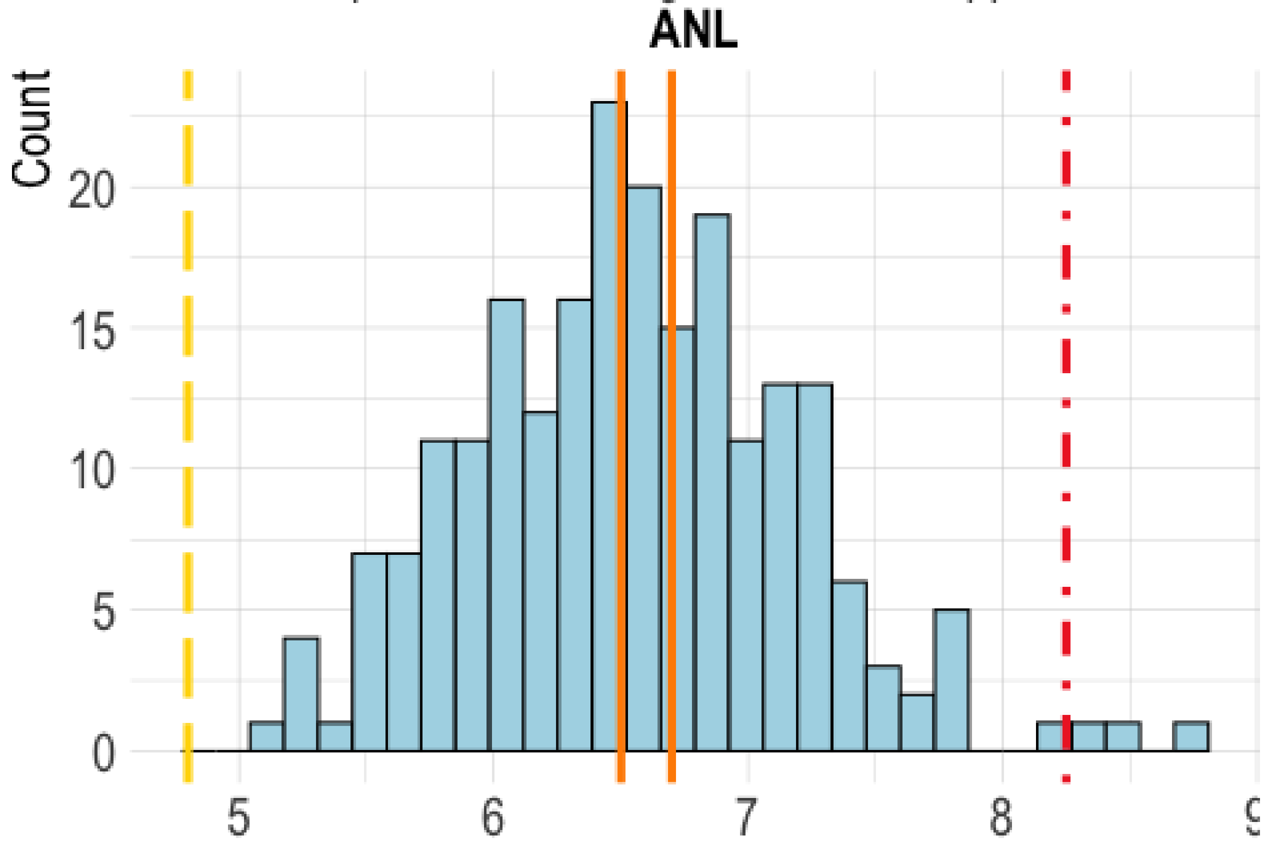

STATL
Count
0
5
10
15
20
25
2.5
5.0
7.5
10.0
12.5

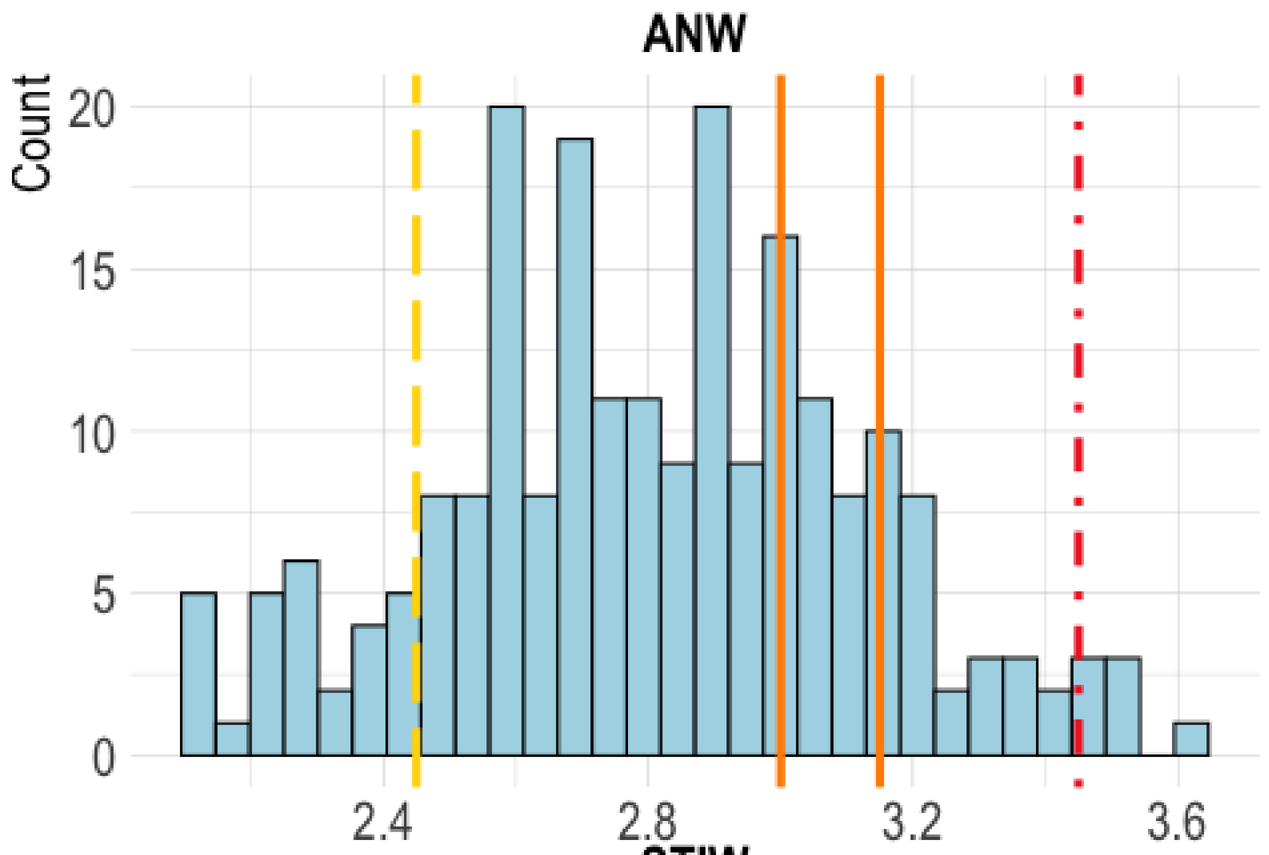

LABL
Count
0
10
20
30
40
50
60
70

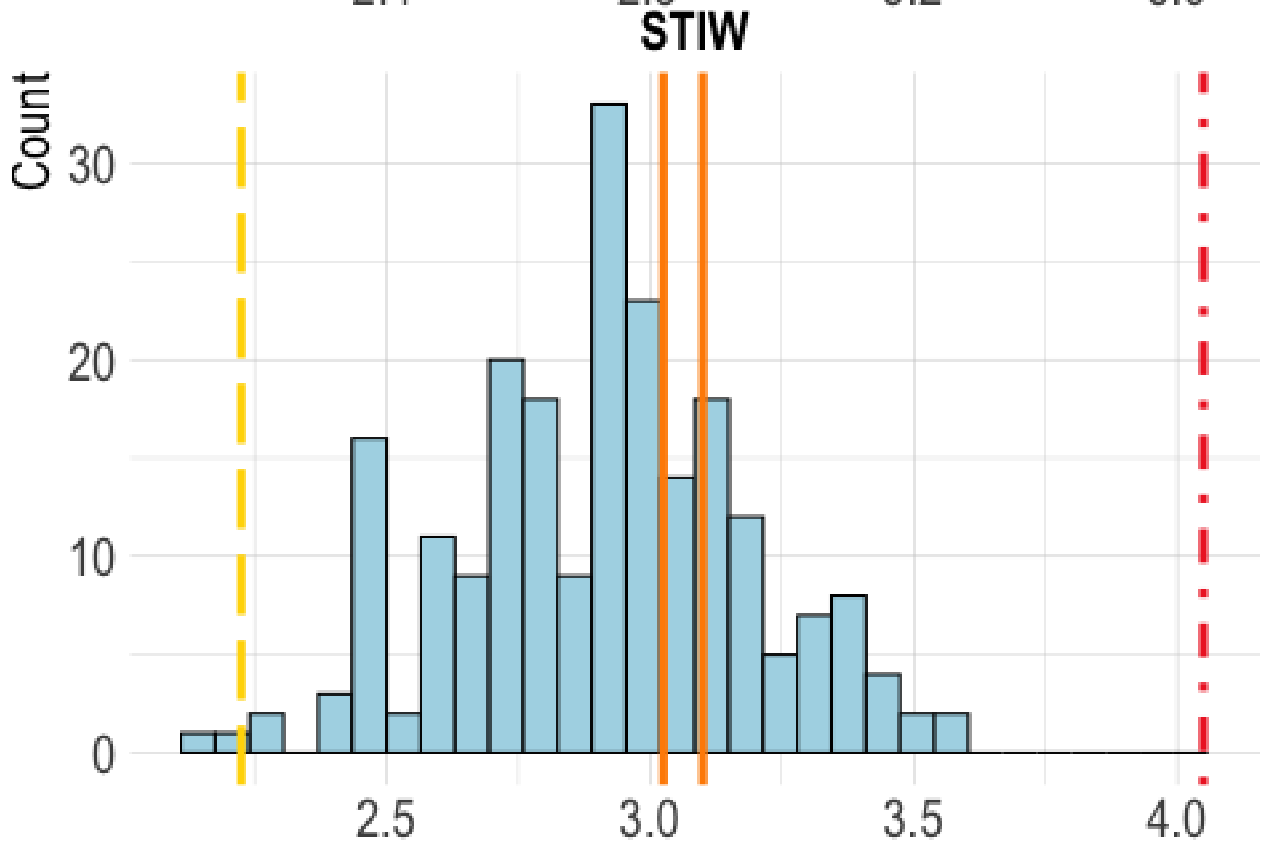

LABW
Count
0
10
20
30
20
30
40

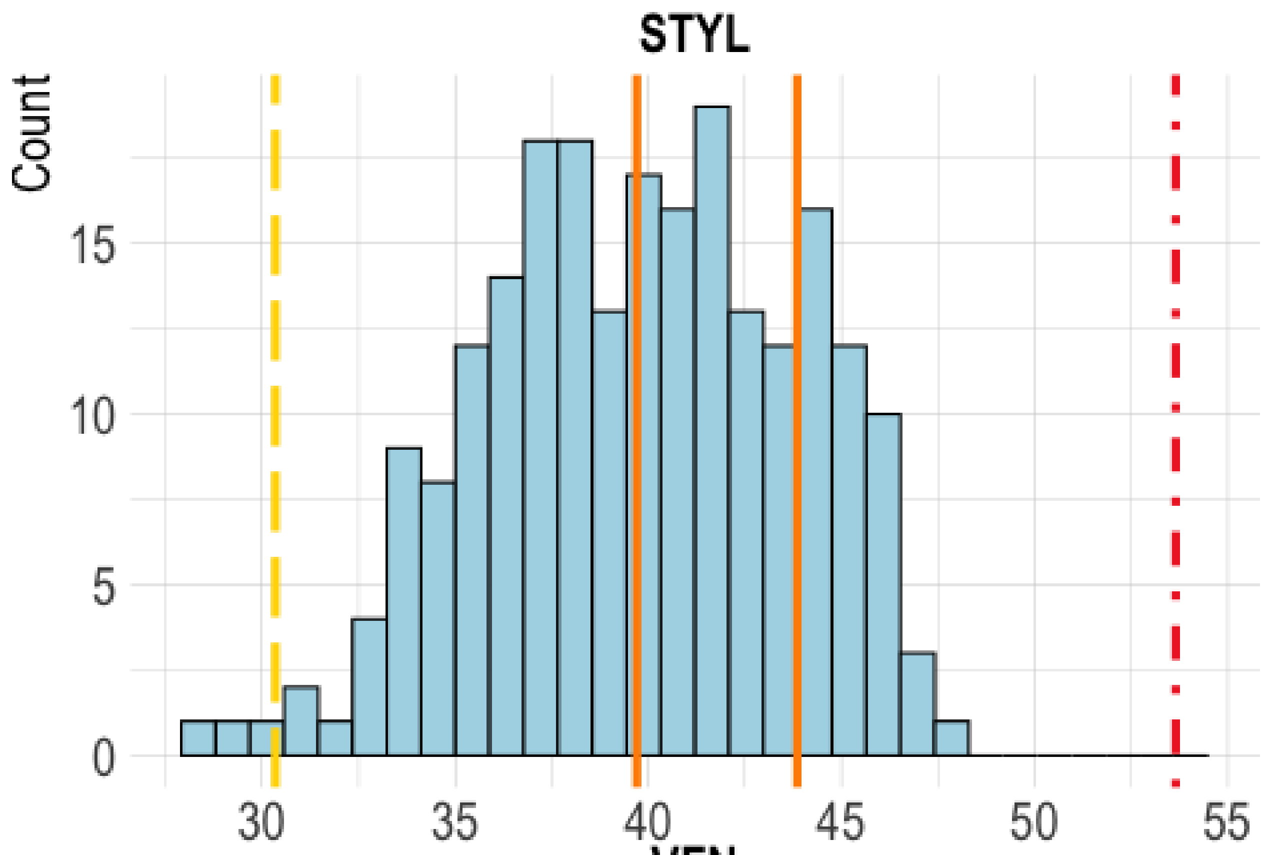

CLL
Count
0
5
10
15
4
6
8

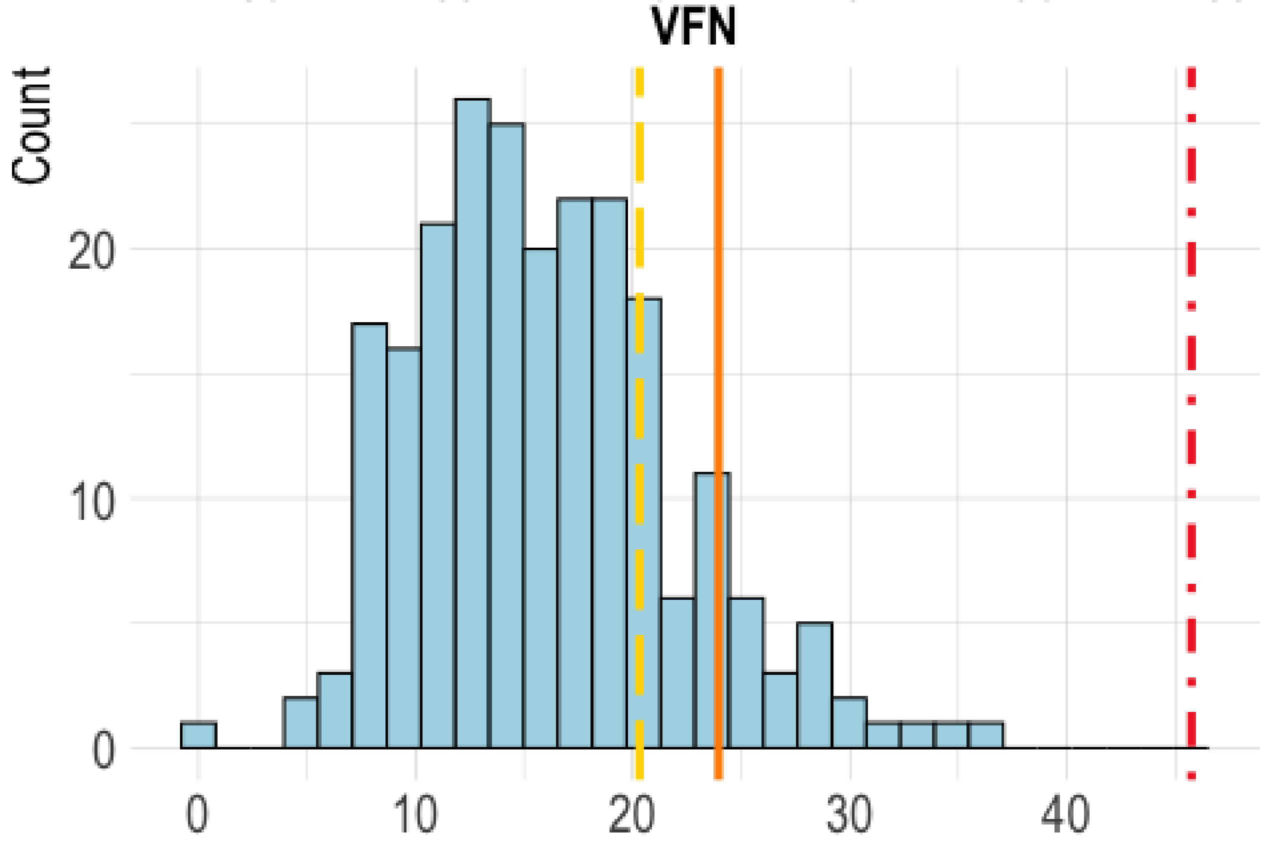

STAL
Count
0
5
10
15
20
25
40
45
50
55
60

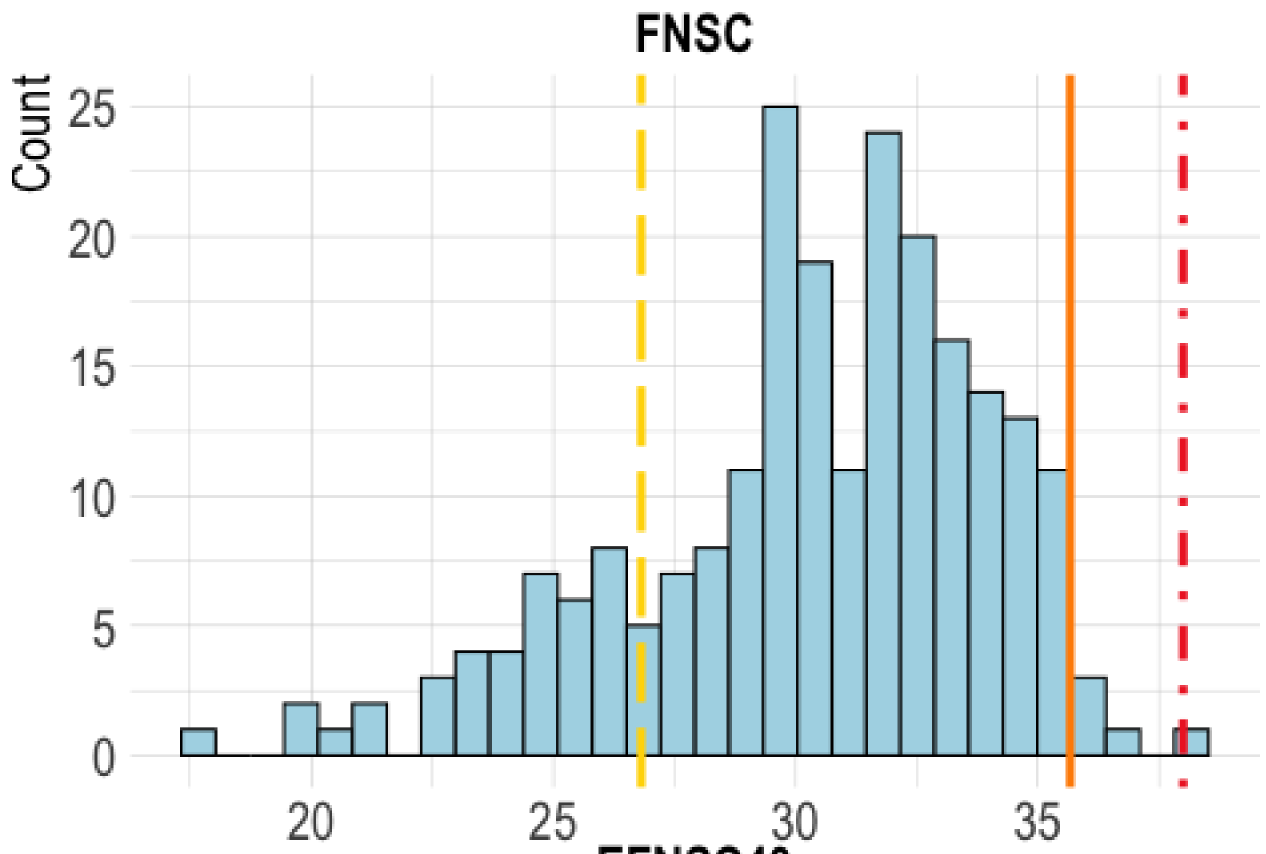

STAW
Count
25
20
15
10
5
0
7
9
11

ANL
Count
20
15
10
5
0
5
6
7
8

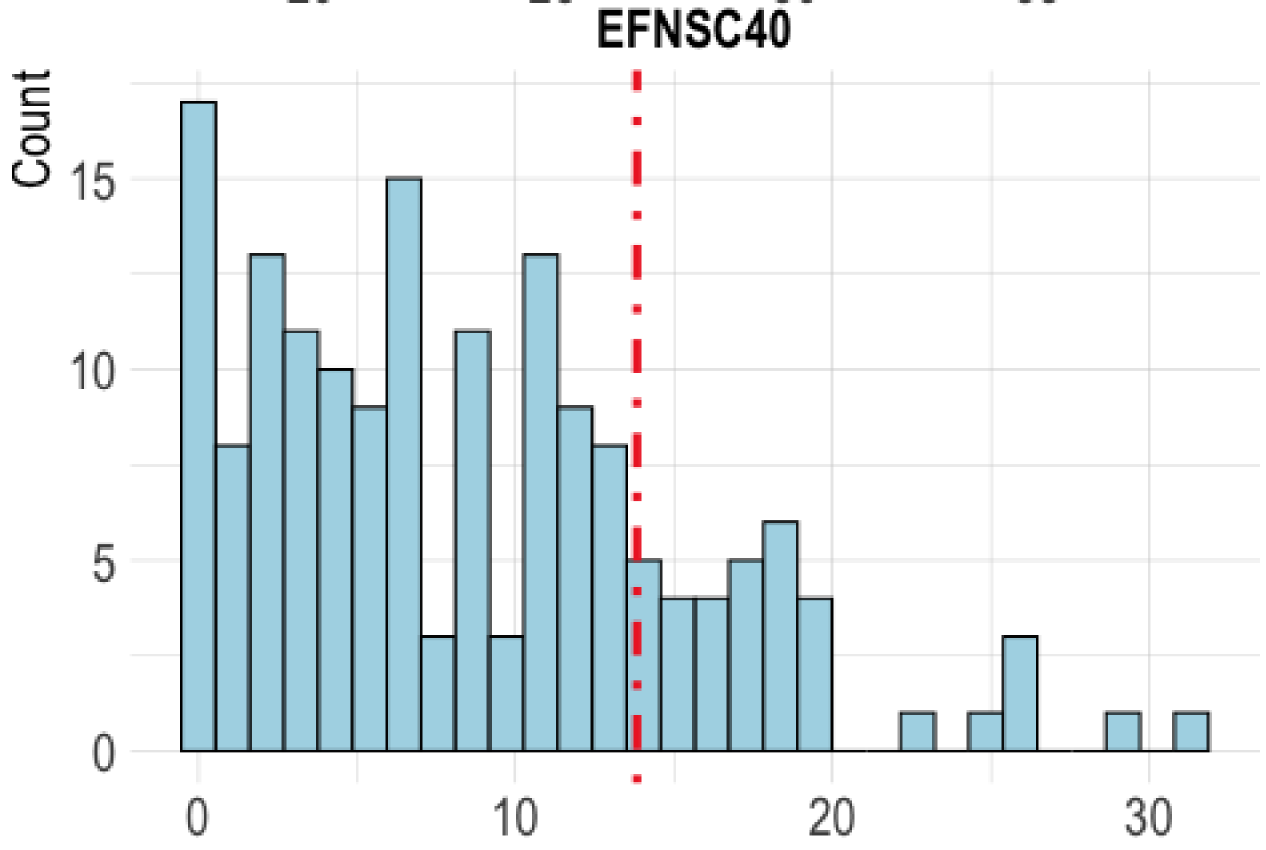

ANW
Count
20
15
10
5
0
2.4
2.8
3.2
3.6


STIW
Count
30
20
10
0
2.5
3.0
3.5
4.0


STYL
Count
0
5
10
15
30
35
40
45
50
55


VFN
Count
0
10
20
0
10
20
30
40


FNSC
Count
0
5
10
15
20
25
20
25
30
35


EFNSC40
Count
0
5
10
15
0
10
20
30

Combines plots

```r
# 6. Create Legend Plot
legend_data <- data.frame(
  x = c(1, 2, 3),
  Category = c("C. lasius",
               "C. bracteatus",
               "F1 hybrid")
)

legend_plot <- ggplot(legend_data, aes(x = x)) +
  geom_vline(aes(xintercept = x, color = Category, linetype = Category), size = 1.2) +
  scale_color_manual(
    name = NULL,
    values = c(
      "C. lasius" = "gold",
      "C. bracteatus" = "firebrick2",
      "F1 hybrid" = "darkorange"
    )
  ) +
  scale_linetype_manual(
    name = NULL,
    values = c(
      "C. lasius Parental Species" = "longdash",
      "C. bracteatus Parental Species" = "dotdash",
      "F1 Hybrids" = "solid"
    )
  ) +
  theme_void() +
  theme(
    legend.position = "bottom",
    legend.text = element_text(size = 12)
  )

# 7. Combine Histograms into a Grid
nrow_grid <- 5 # Adjust based on number of histograms and desired layout

histogram_grid <- plot_grid(
  plotlist = histograms,
  nrow = nrow_grid,
  align = "v"
)

# 8. Combine Histogram Grid with Legend Plot
combined_plot <- plot_grid(
  histogram_grid,
  legend_plot,
  ncol = 1,
  rel_heights = c(1, 0.05) # Adjust relative heights as needed
)

# 9. Save the Combined Plot to a PDF
CairoPDF(file = "~/Dropbox/Costus/costus-genetic-mapping/phenotype/results/figures/morphometric_combine
print(combined_plot)
```

```r
dev.off()
```

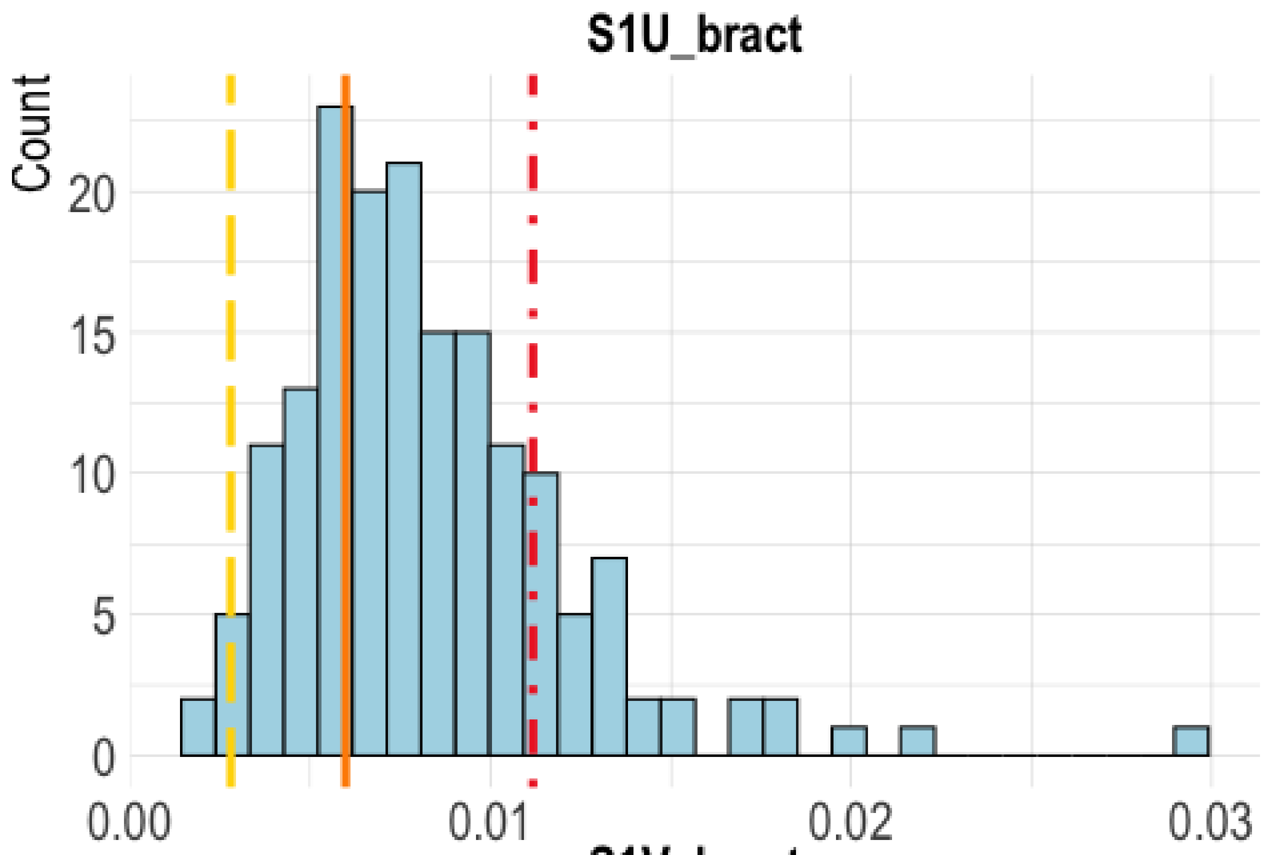
```
pdf
  2
```

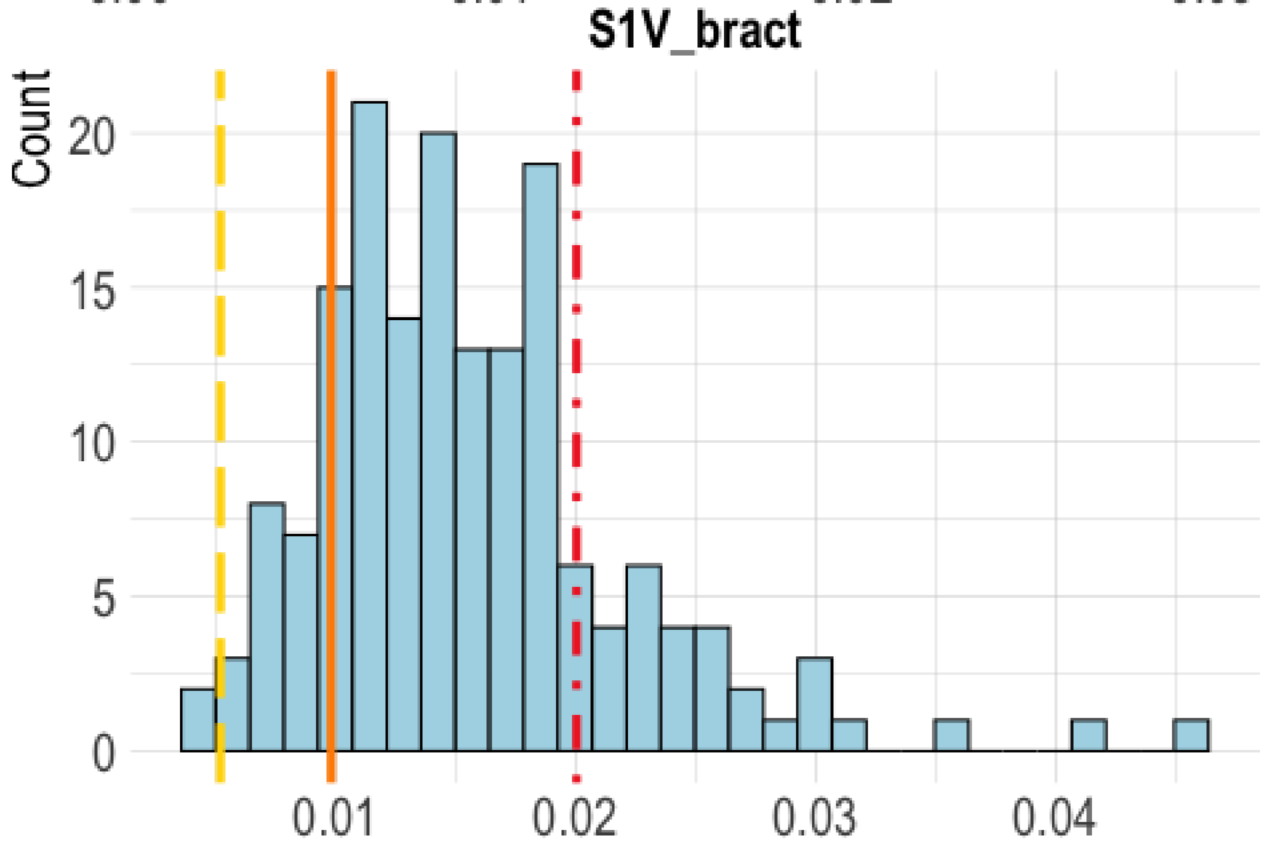
```r
# Optional: Display the Combined Plot in R Session
# print(combined_plot)
```

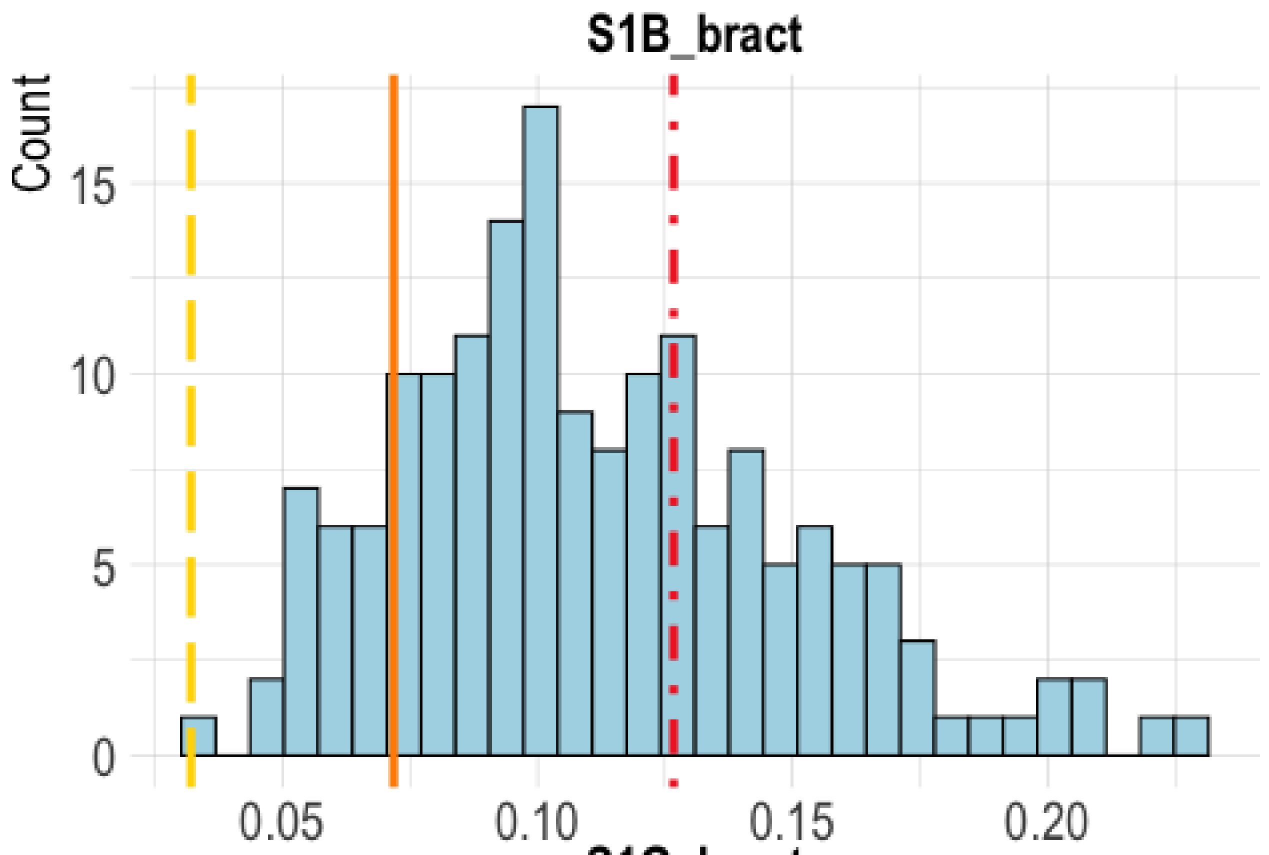
Colormetric histogram plots

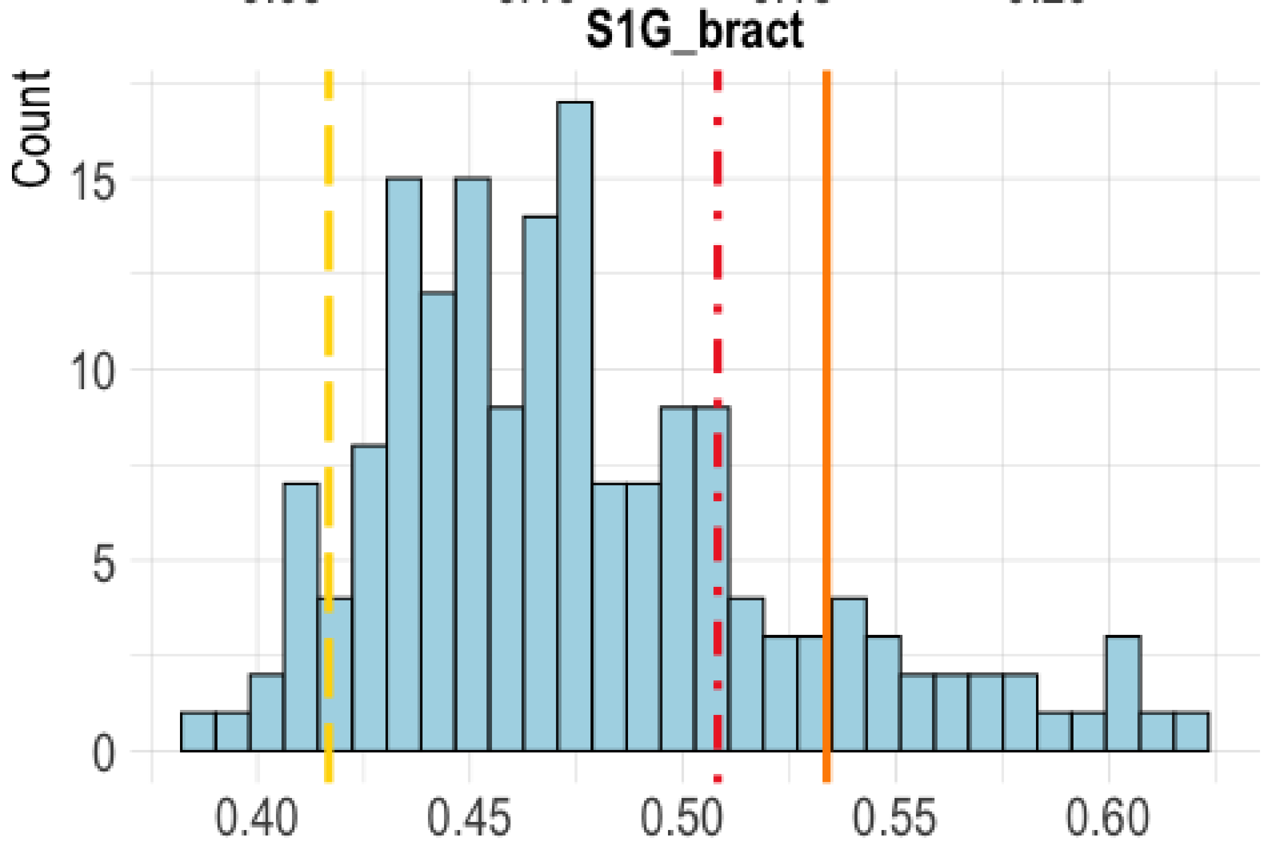
```r
# 1. Define the list of traits you want to plot
# Exclude non-numeric columns if necessary
traits <- c("S1U_bract","S1V_bract","S1B_bract","S1G_bract","S1R_bract",
            "S5_bract","S9_bract","H4_bract","B3_petal","S1U_petal",
            "S1V_petal","S1B_petal","S1Y_petal","S1R_petal","S5_petal",
            "S9_petal","H4_petal","B3_labellum","S1B_labellum","S1Y_labellum",
            "S1R_labellum","S5_labellum","S6_labellum","S9_labellum","H3_labellum",
            "H4_labellum")

# 2. Specify binwidths for each trait
binwidths <- list(
  INFA = 5,
  CAL = NULL,
  VEFN = NULL,
  VNG = NULL,
  RALA = NULL,
  RAST = NULL,
  COL = NULL,
  COLL = NULL,
  STAE = NULL,
  TUA = NULL,
  STATL = NULL,
  LABL = NULL,
  LABW = NULL,
  CLL = NULL,
  STAL = NULL,
  STAW = NULL,
  ANL = NULL,
  ANW = NULL,
  STIW = NULL,
  STYL = NULL,
  VFN = NULL,
  FNSC = NULL,
  EFNSC40 = NULL
)

# Ensure all traits have a specified binwidth
# If some traits are missing, you can set a default binwidth
default_binwidth <- NULL
for(trait in traits){
  if(!trait %in% names(binwidths)){
    binwidths[[trait]] <- default_binwidth
    warning(paste("Binwidth for trait", trait, "not specified. Using default binwidth =", default_binwi
  }
```

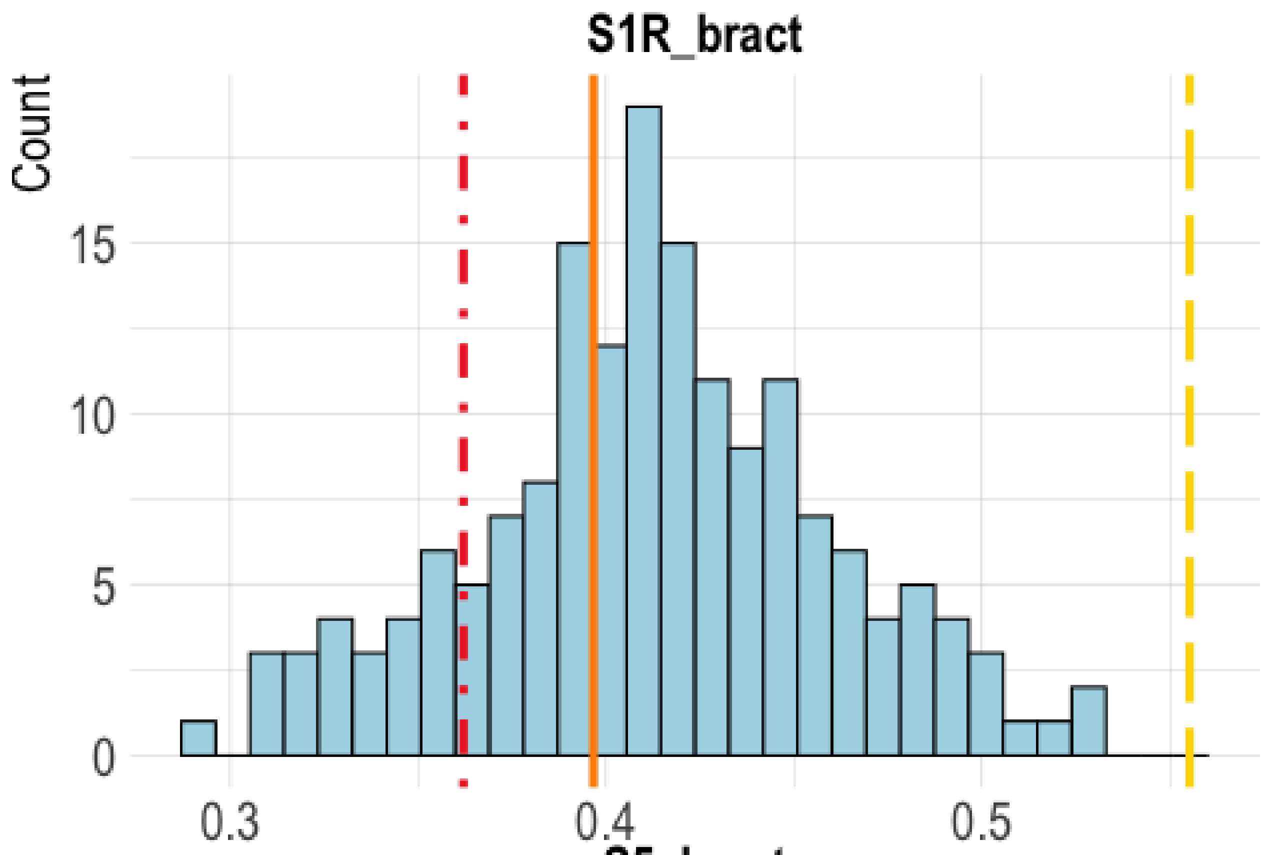
```
}

# 3. Define Unique IDs and Plant Types
unique_ids <- c("F1_39", "F1_62", "P_125", "P_126", "P_950")
plant_types <- c("39", "62") #

# 4. Create a Function to Generate Histogram for a Single Trait
create_histogram <- function(trait, binwidth){

  # Extract averages for the current trait
  F1_39avg <- as.numeric(color[color$id == "39", trait])
  F1_62avg <- as.numeric(color[color$id == "62", trait])
  P_125avg <- as.numeric(color[color$id == "125", trait])
  P_126avg <- as.numeric(color[color$id == "126", trait])
  P_950avg <- as.numeric(color[color$id == "BRAC", trait])

  # Calculate mean of P_125avg and P_126avg
  P_mean_avg <- mean(c(P_125avg, P_126avg), na.rm = TRUE)

  # Removing rows where id is "39" or "62" (character or factor)
    filtered_color <- color %>%
      filter(!id %in% c("39", "62"))

  # Create the histogram if the trait is numeric:
  if(is.numeric(filtered_color[[trait]])){

    hist_plot <- ggplot(filtered_color[filtered_color$plant_type %in% plant_types, ], aes_string(x = tra
    geom_histogram(binwidth = binwidth, color = "#000000", fill = "lightblue") +
    ylab("Count") +
    geom_vline(xintercept = F1_39avg, linetype = "solid", color = "darkorange", size = 1.5) +
    geom_vline(xintercept = F1_62avg, linetype = "solid", color = "darkorange", size = 1.5) +
    geom_vline(xintercept = P_mean_avg, linetype = "longdash", color = "gold", size = 1.5) +
    geom_vline(xintercept = P_950avg, linetype = "dotdash", color = "firebrick2", size = 1.5) +
    labs(title = trait) +
    theme_ipsum(base_size = 20) +
    theme(
      plot.title = element_textbox(hjust = 0.5, margin = margin(t = 5, b = 5), size = 20),
      axis.title.x = element_blank(),
      axis.title.y = element_text(
    size = 20,         # Increased y-axis title font size
    color = "black"    # Set y-axis title color to black
  ),
      plot.margin = unit(c(0.1, 0, 0.1, 0), "cm")
    )
  }

  # Create the histogram if the trait is a factor
  if(is.factor(filtered_color[[trait]])){

    # Using na.omit() to remove rows with NA in the 'trait' column
    color_filtered <- filtered_color %>%
    filter(plant_type %in% plant_types) %>%
    na.omit(select(., all_of(trait)))  # Removes rows where 'trait' is NA
    hist_plot <- ggplot(filtered_color[filtered_color$plant_type %in% plant_types, ], aes_string(x = tra
    geom_bar(color = "#000000", fill = "lightblue") +  # Changed from geom_histogram() to geom_bar()
    ylab("Count") +
    geom_vline(xintercept = F1_39avg, linetype = "solid", color = "darkorange", size = 1.5) +
    geom_vline(xintercept = F1_62avg, linetype = "solid", color = "darkorange", size = 1.5) +
    geom_vline(xintercept = P_mean_avg, linetype = "longdash", color = "gold", size = 1.5) +
    geom_vline(xintercept = P_950avg, linetype = "dotdash", color = "firebrick2", size = 1.5) +
    labs(title = trait) +
    theme_ipsum(base_size = 20) +
    theme(
      plot.title = element_textbox(hjust = 0.5, margin = margin(t = 5, b = 5), size = 15),
      axis.title.x = element_blank(),
      axis.title.y = element_text(                             # Added y-axis title customization
        size = 20,                                             # Increased font size
        face = "bold",                                         # Made the text bold
        color = "black",                                       # Set text color to black
        angle = 90,                                            # Ensure the y-axis title is vertical
        vjust = 0.5                                            # Vertically center the y-axis title
      ),
      plot.margin = unit(c(0.1, 0, 0.1, 0), "cm")
    )

  }
  return(hist_plot)
}

# 5. Loop Through Each Trait and Generate Histograms
histograms <- list()

for(trait in traits){

  # Retrieve the binwidth for the current trait
  binwidth <- binwidths[[trait]]

  # Create the histogram
  plot <- create_histogram(trait, binwidth)

  # Store the plot in the list
  histograms[[trait]] <- plot

  # Display the plot
  print(plot)
}
```

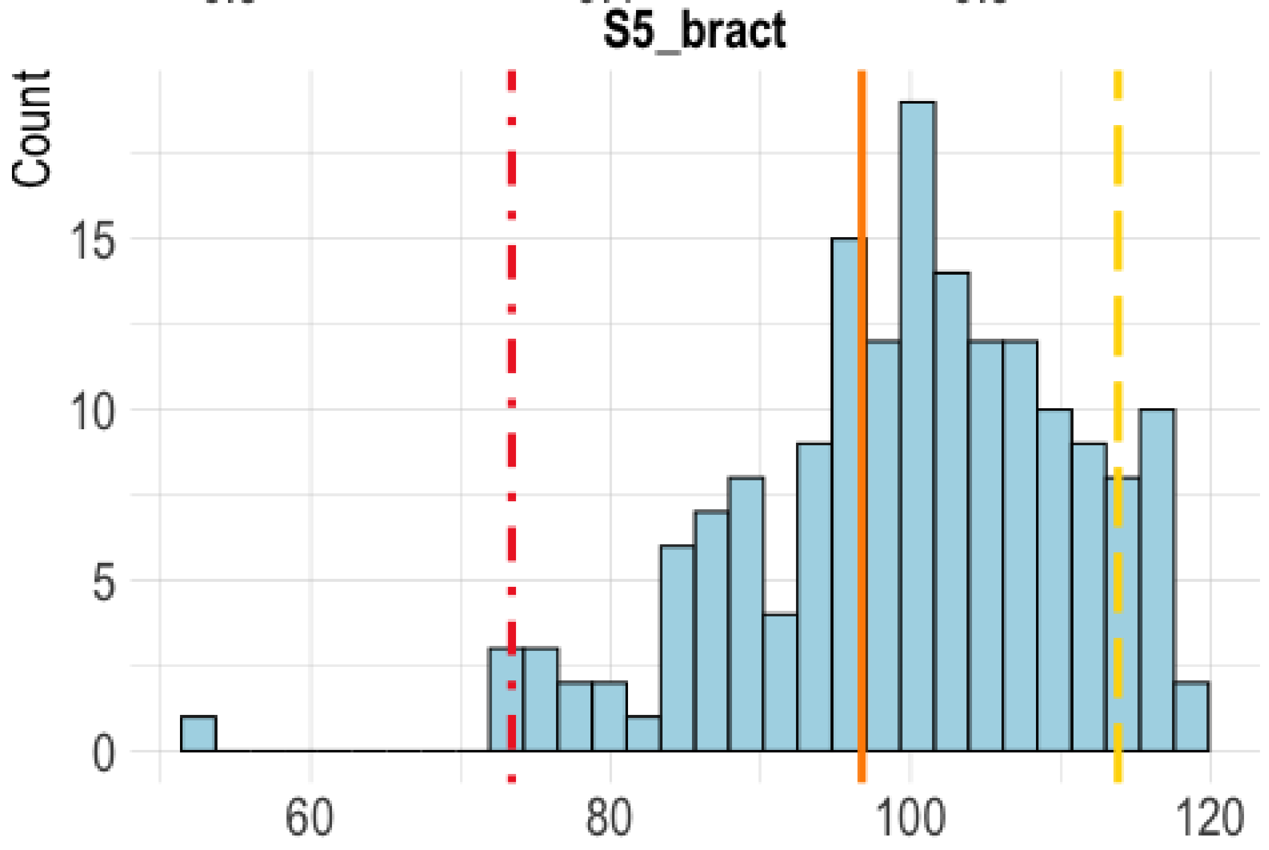

S1U_bract
Count
0
5
10
15
20
0.00
0.01
0.02
0.03

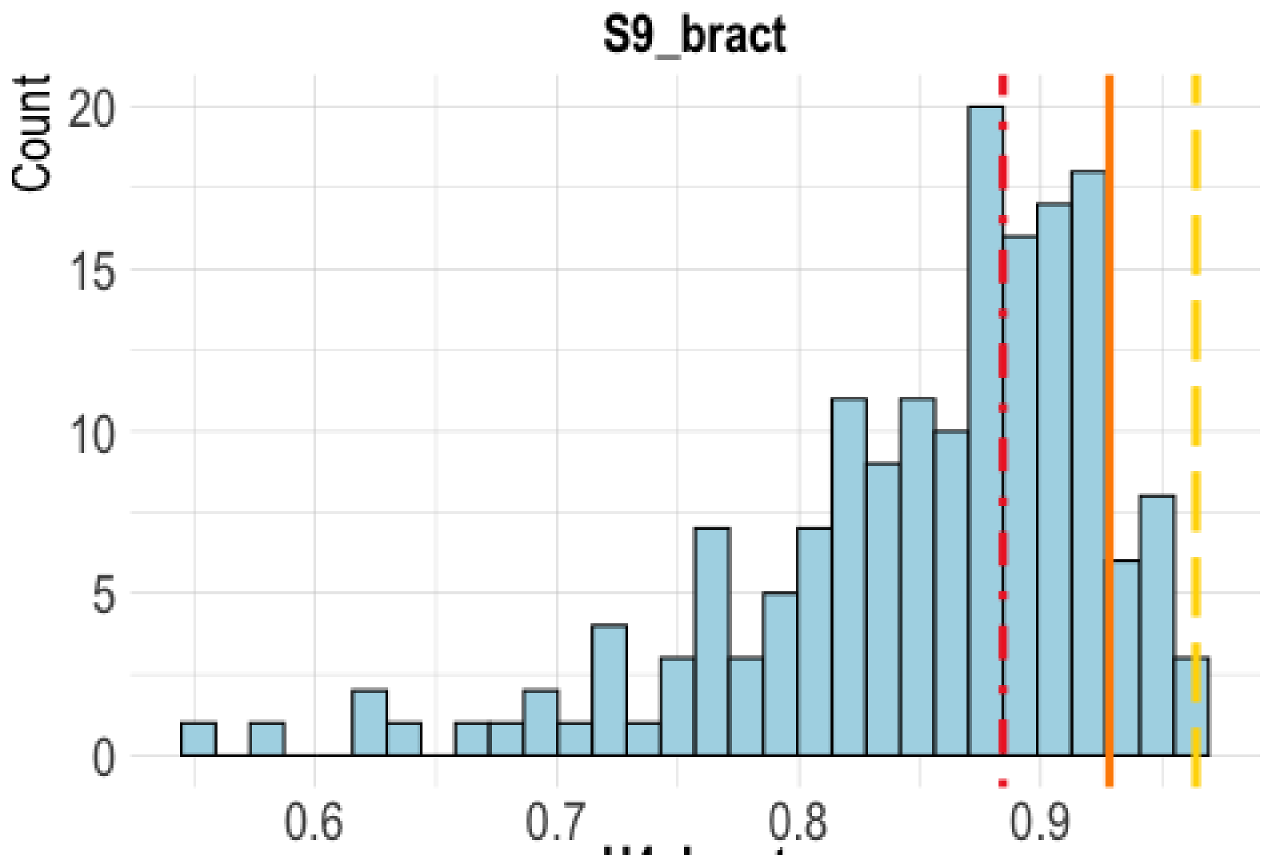

S1V_bract
Count
0
5
10
15
20
0.01
0.02
0.03
0.04

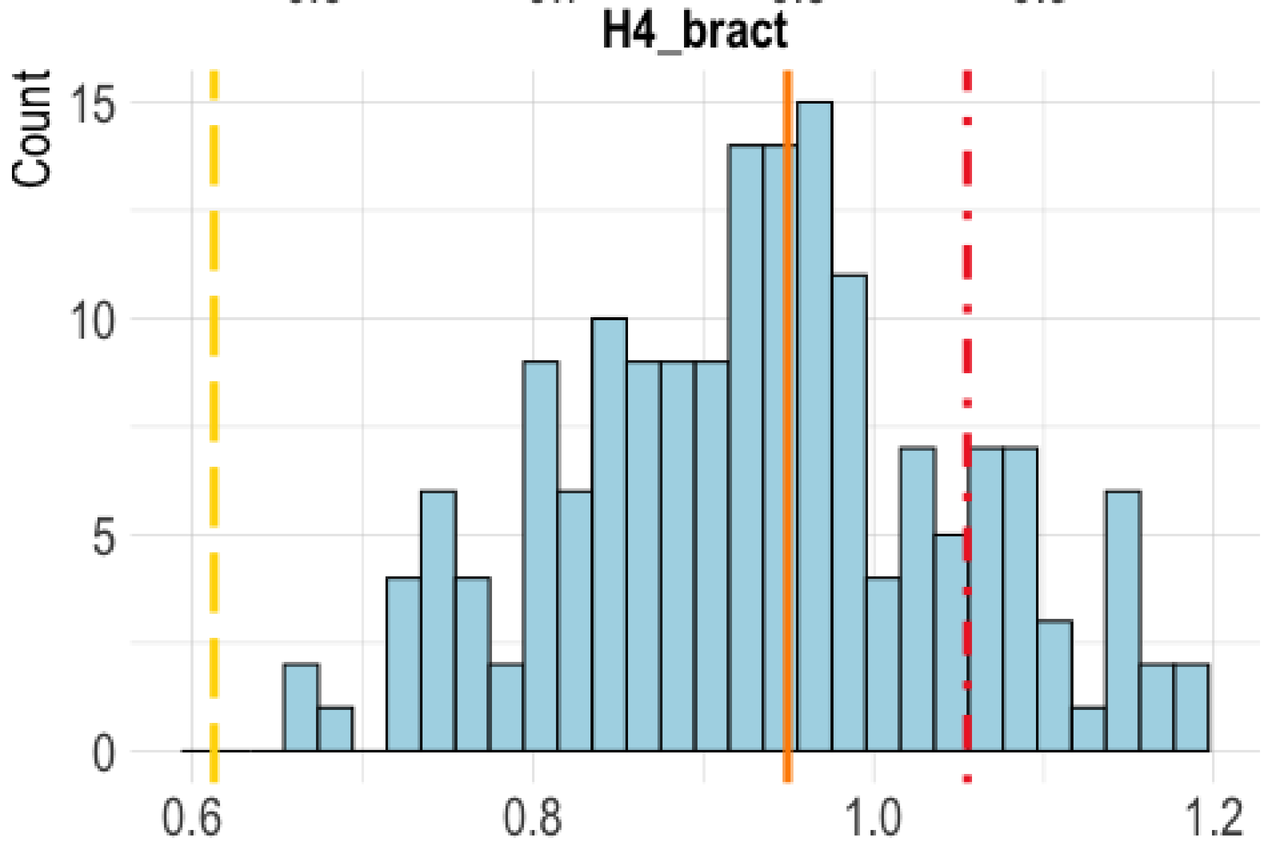

S1B_bract
Count
0
5
10
15
0.05
0.10
0.15
0.20

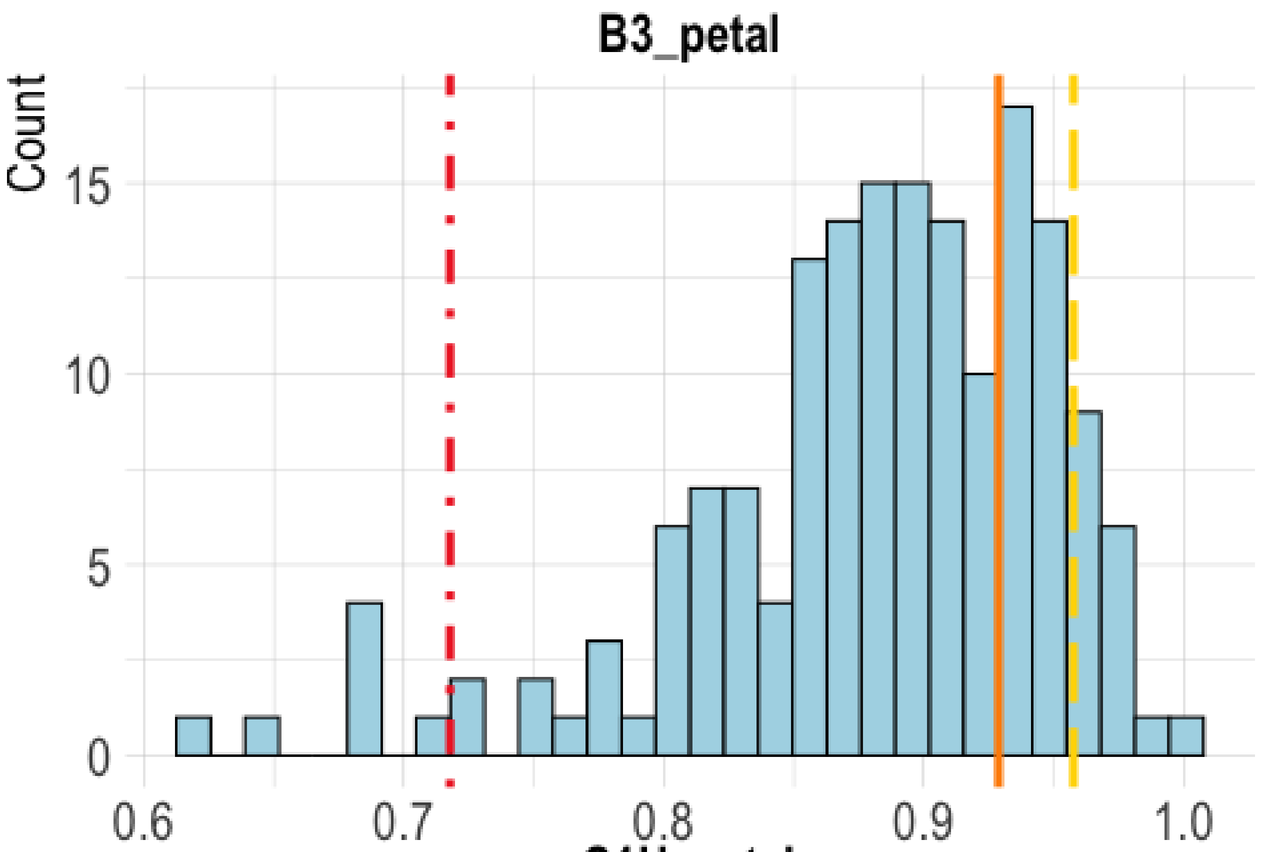

S1G_bract
Count
0
5
10
15
0.40
0.45
0.50
0.55
0.60

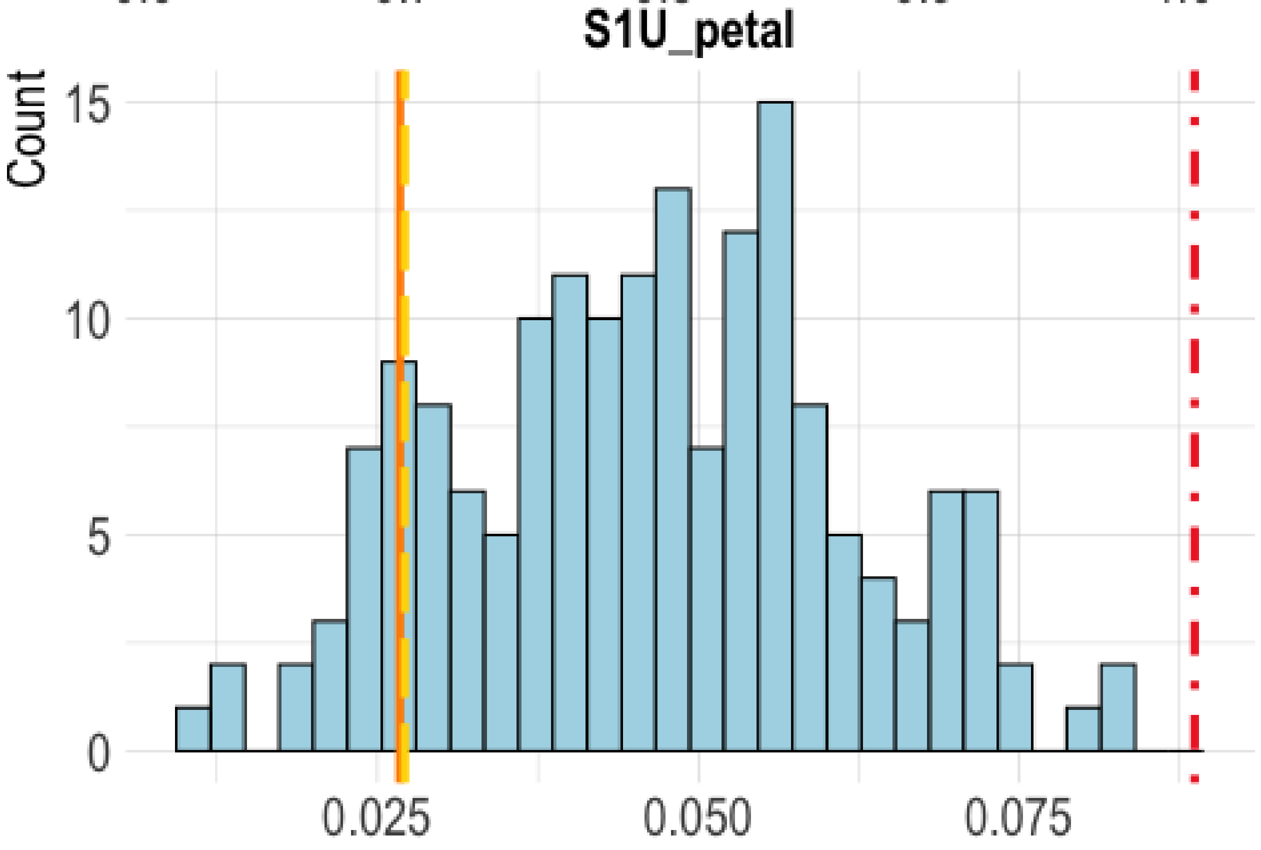

S1R_bract
Count
15
10
5
0
0.3
0.4
0.5

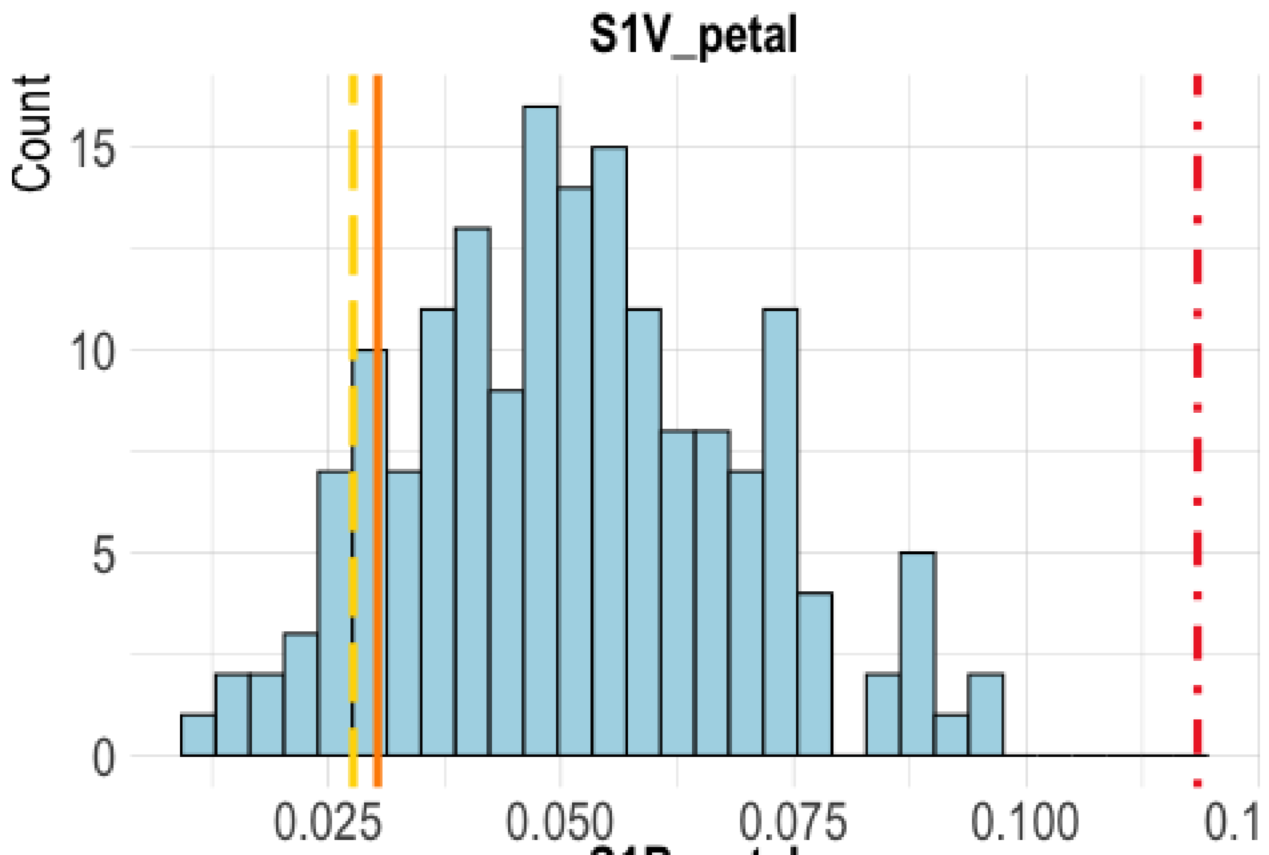

S5_bract
Count
15
10
5
0
60
80
100
120


S9_bract
Count
20
15
10
5
0
0.6
0.7
0.8
0.9

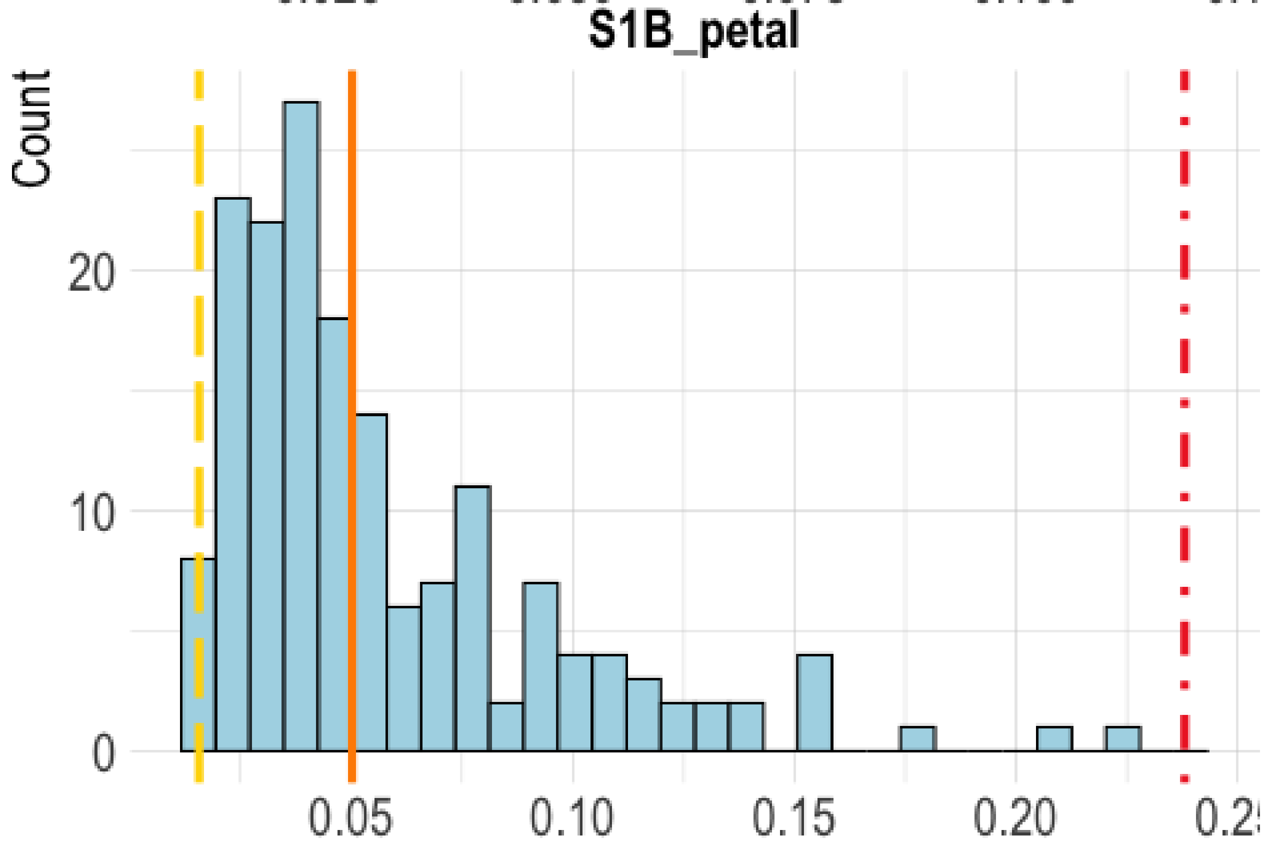

H4_bract
Count
15
10
5
0
0.6
0.8
1.0
1.2


B3_petal
Count
0
5
10
15
0.6
0.7
0.8
0.9
1.0

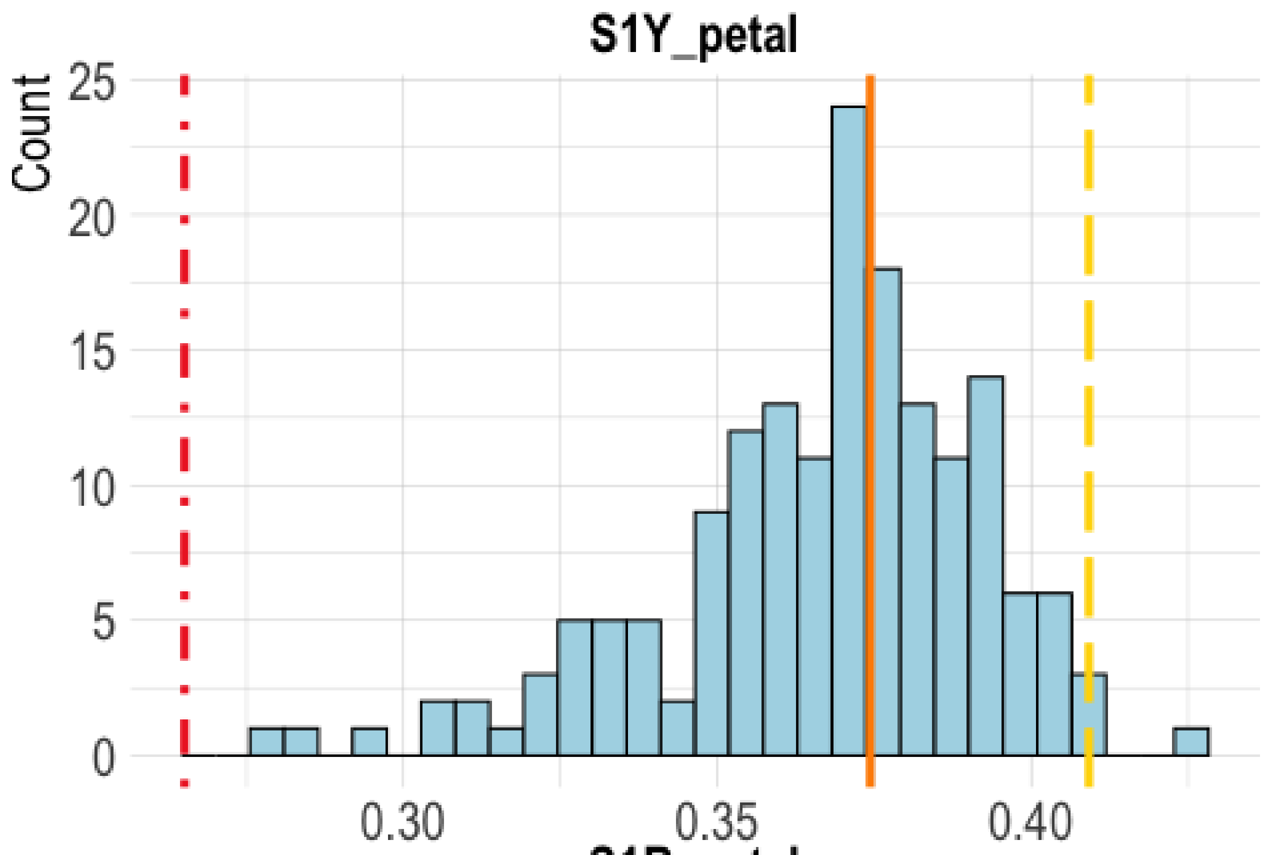

S1U_petal
Count
0
5
10
15
0.025
0.050
0.075

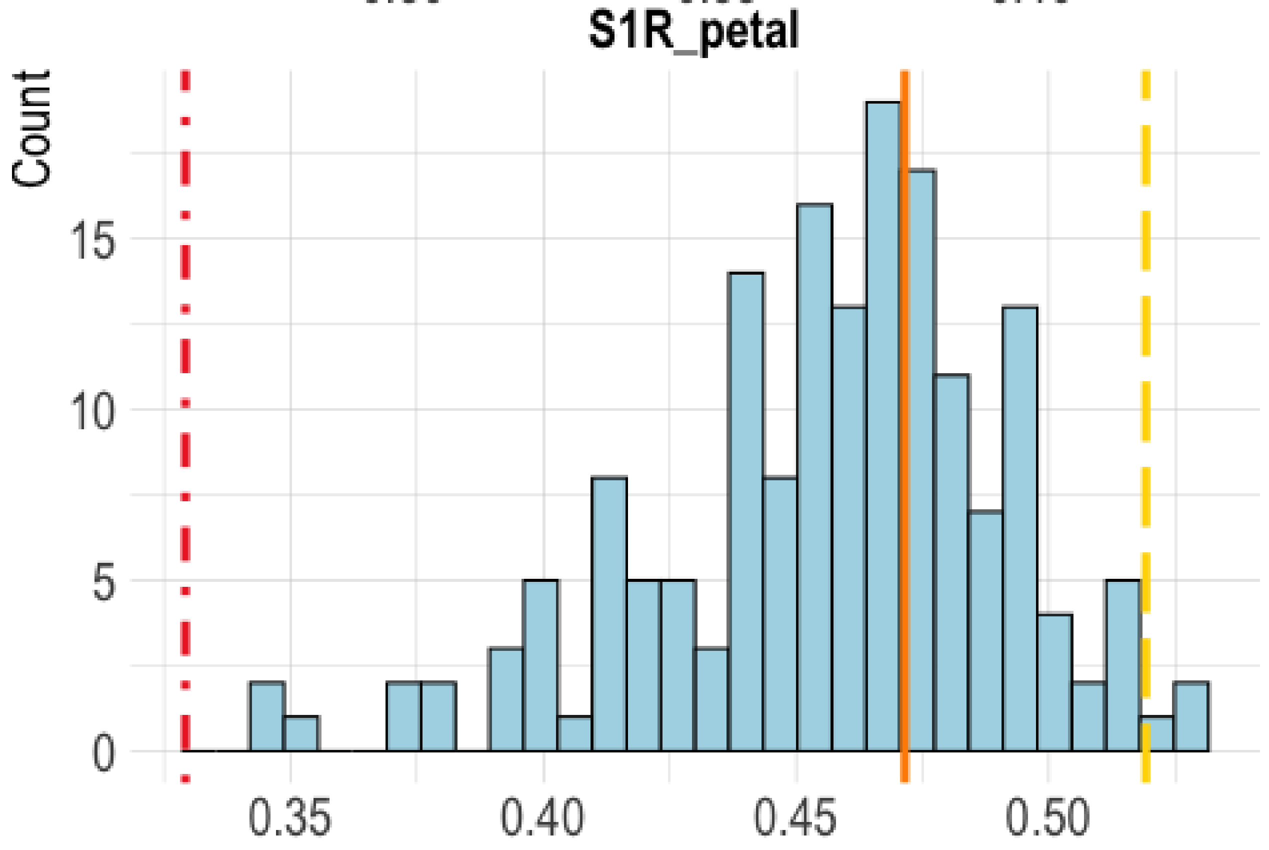


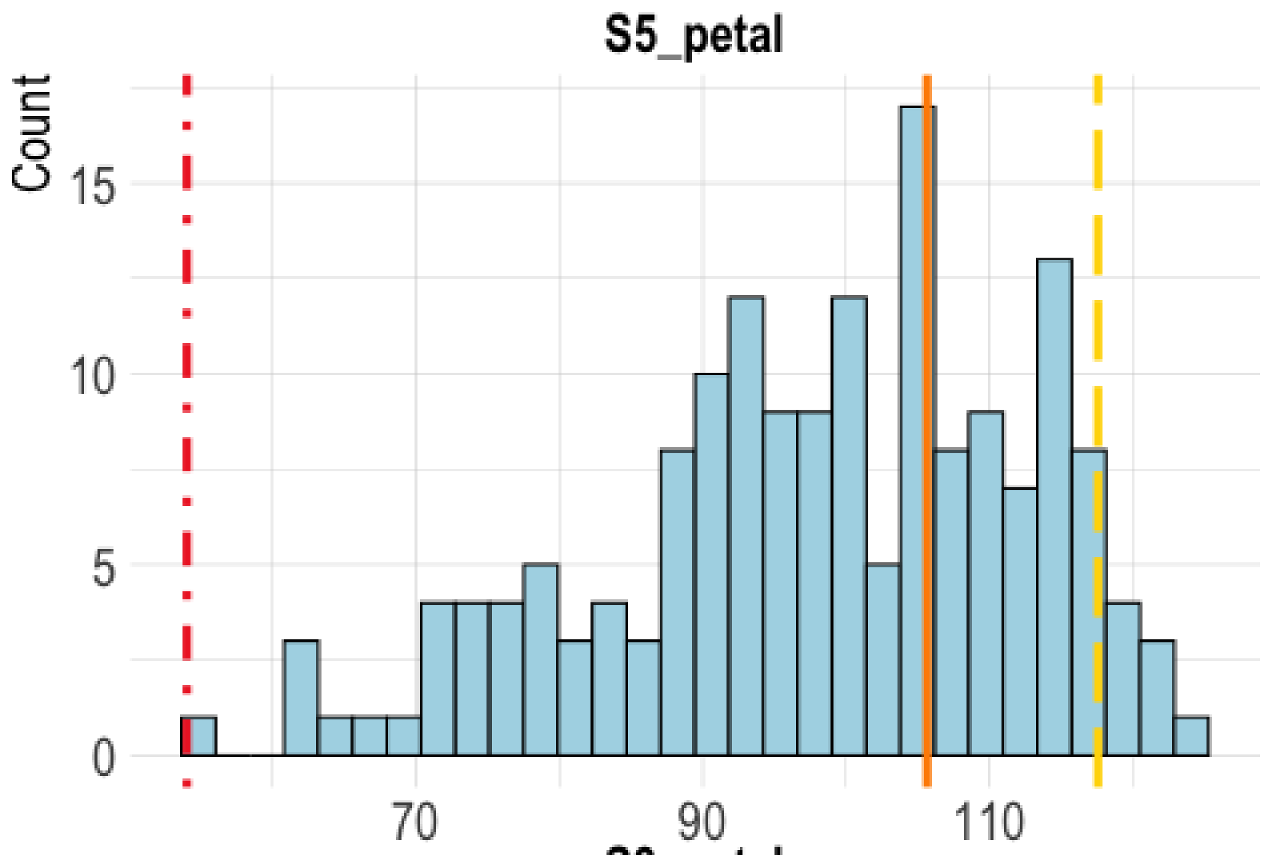
S1V_petal
Count
0
5
10
15
0.025
0.050
0.075
0.100
0.1

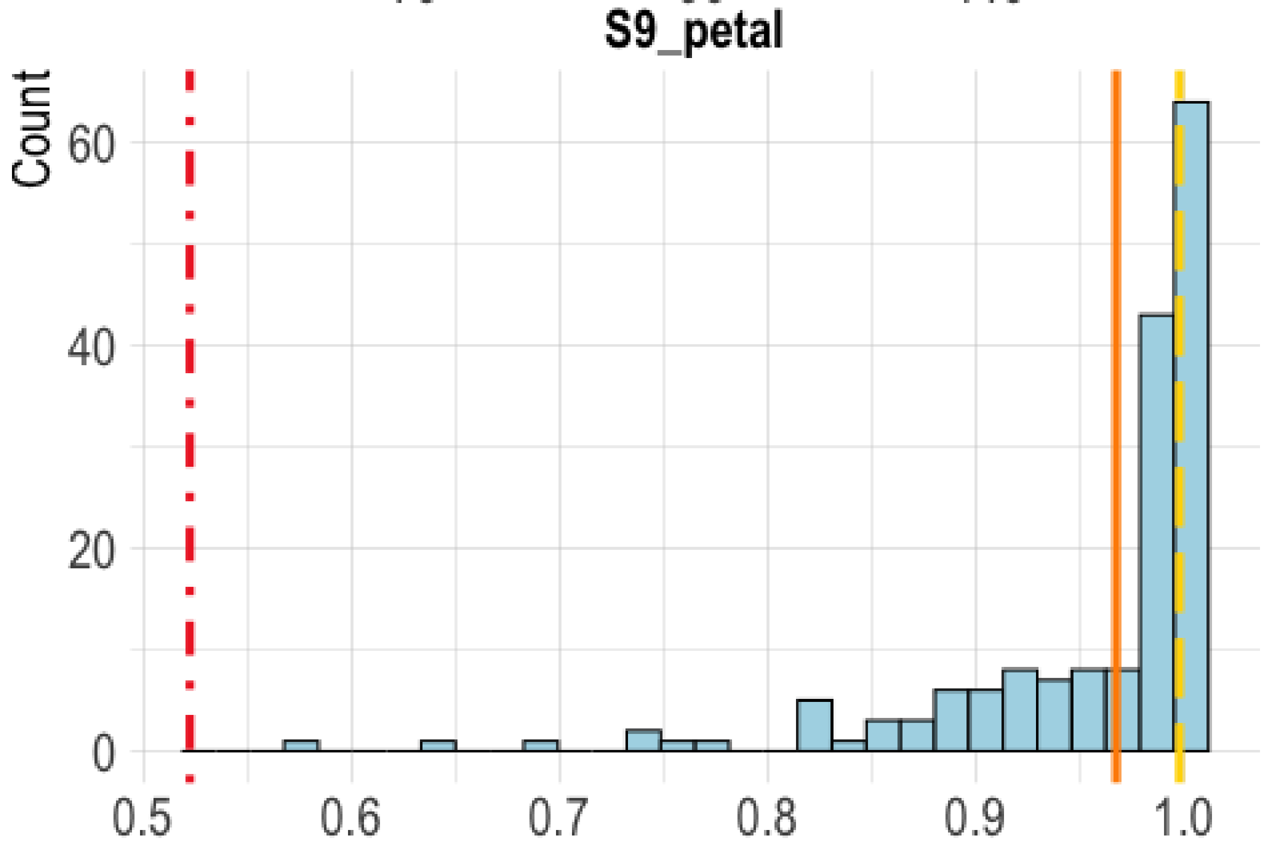


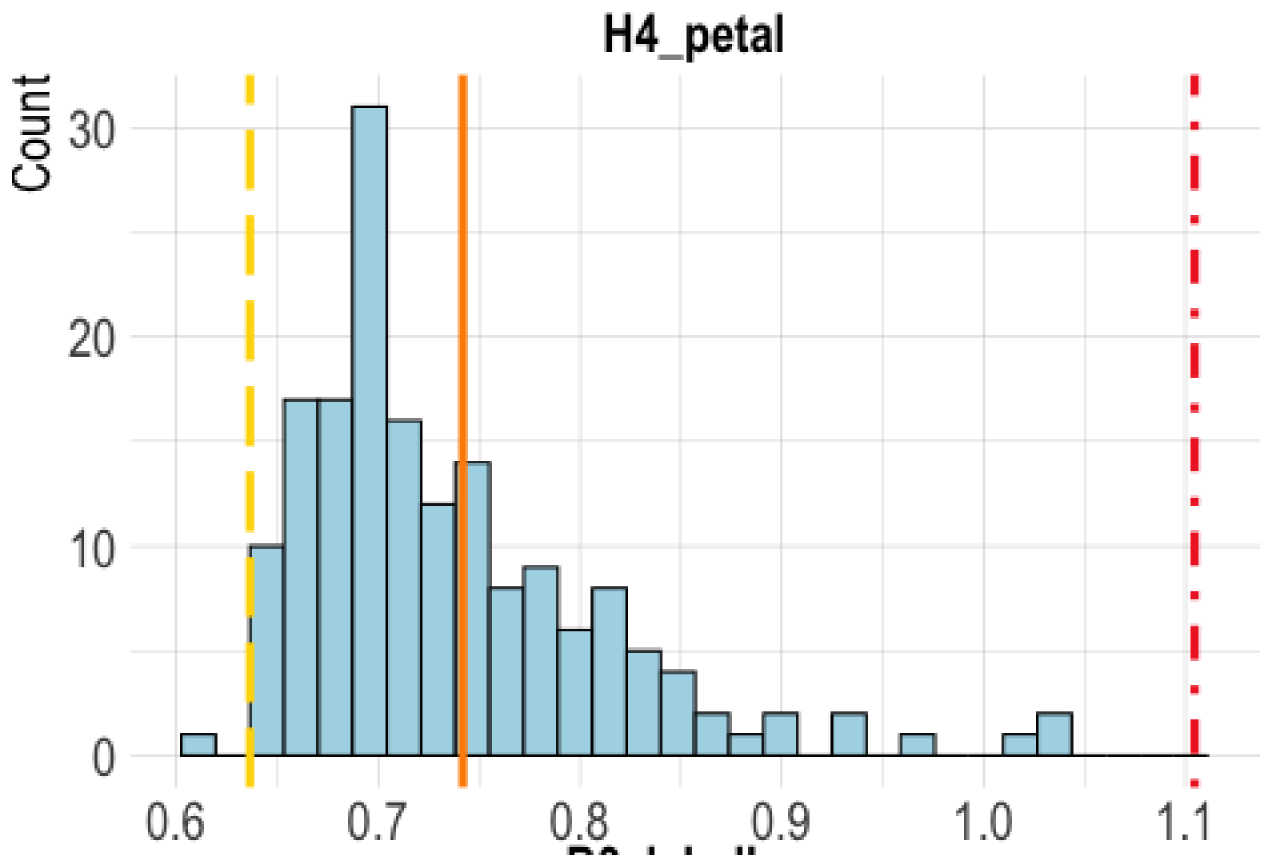
S1B_petal
Count
0
10
20
0.05
0.10
0.15
0.20
0.2

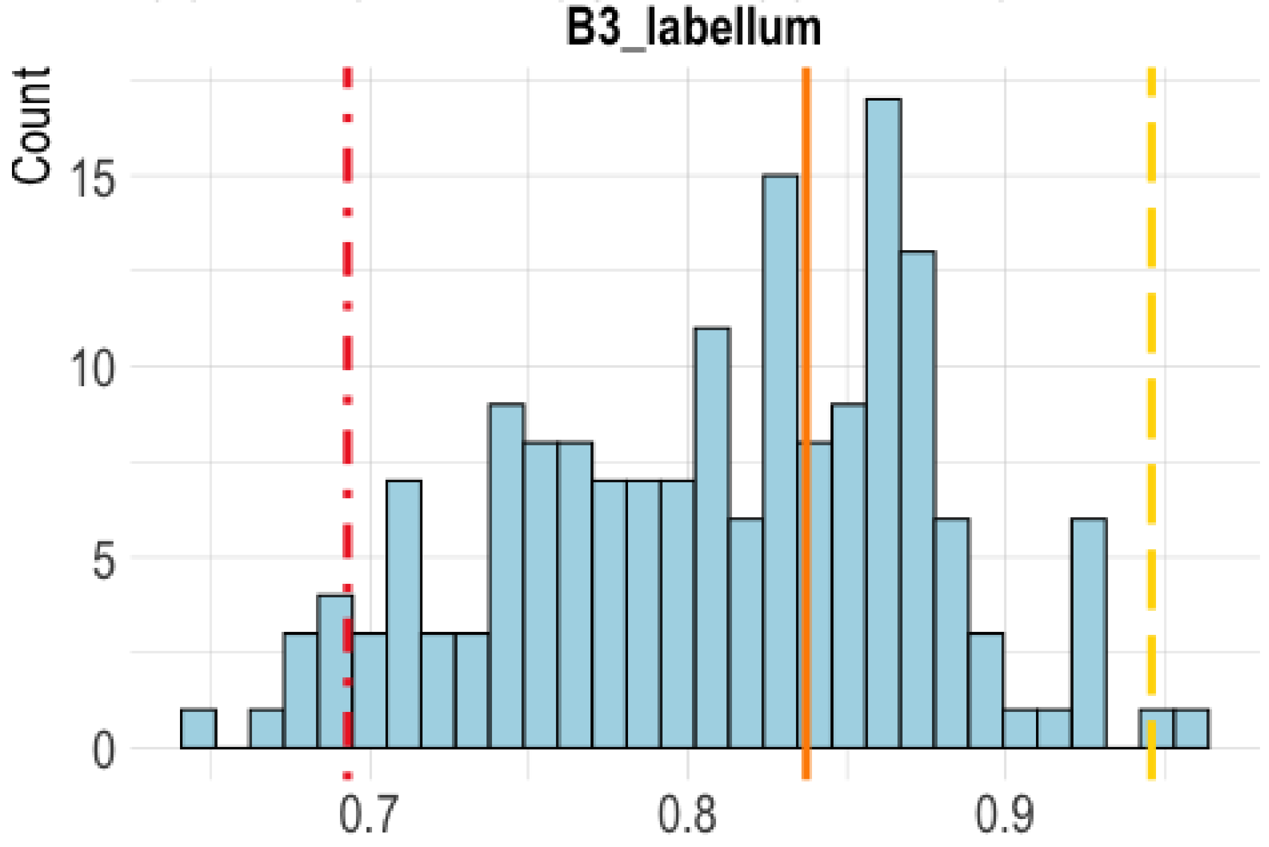

S1Y_petal
Count
25
20
15
10
5
0
0.30
0.35
0.40

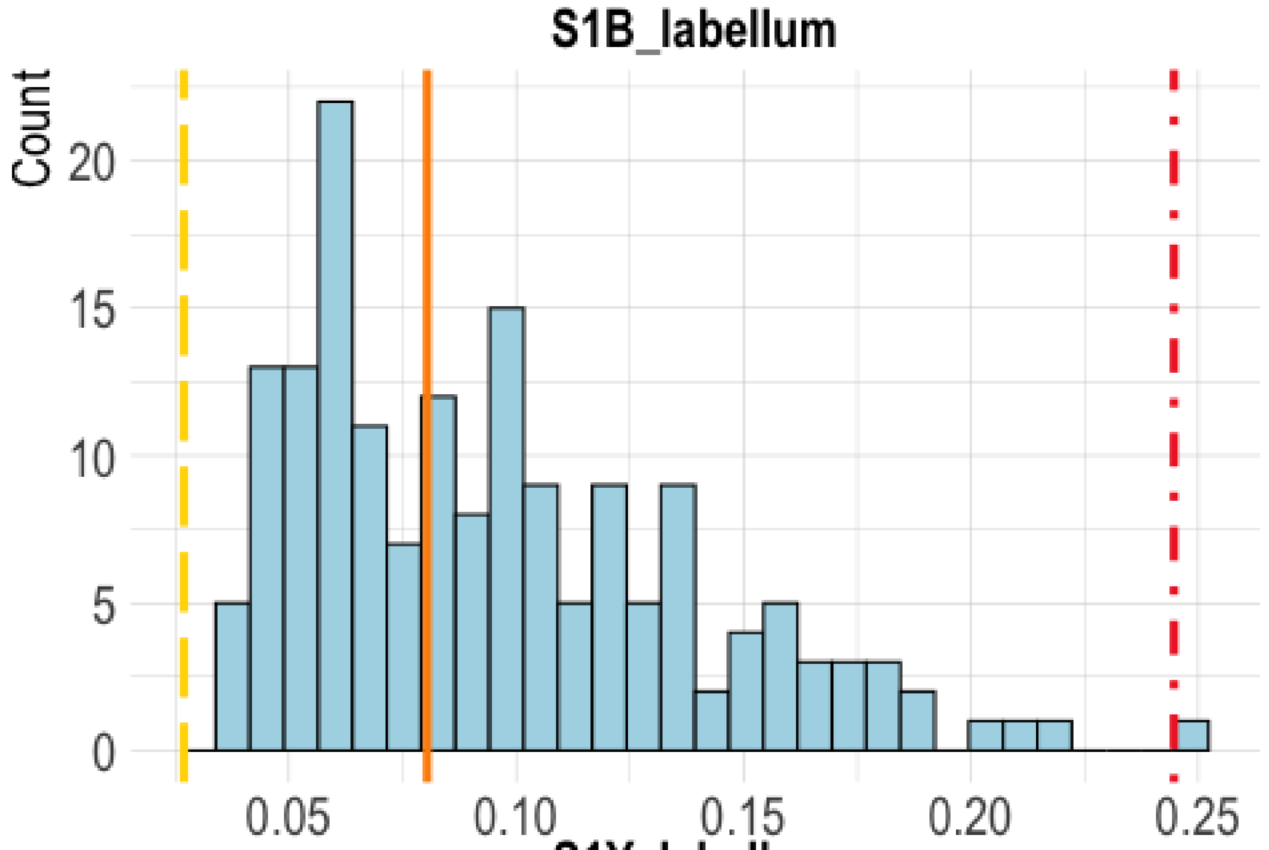

S1R_petal
Count
15
10
5
0
0.35
0.40
0.45
0.50

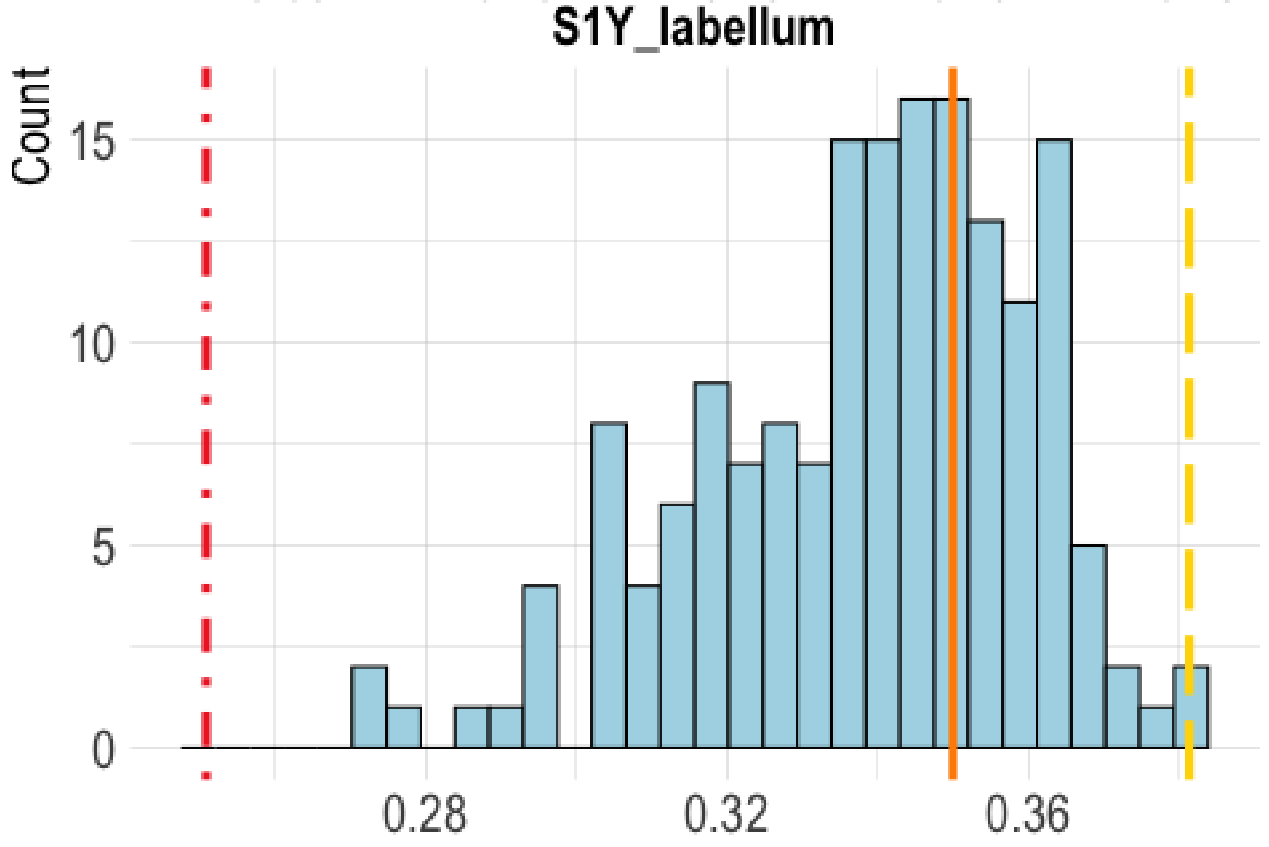

S5_petal
Count
15
10
5
0
70
90
110

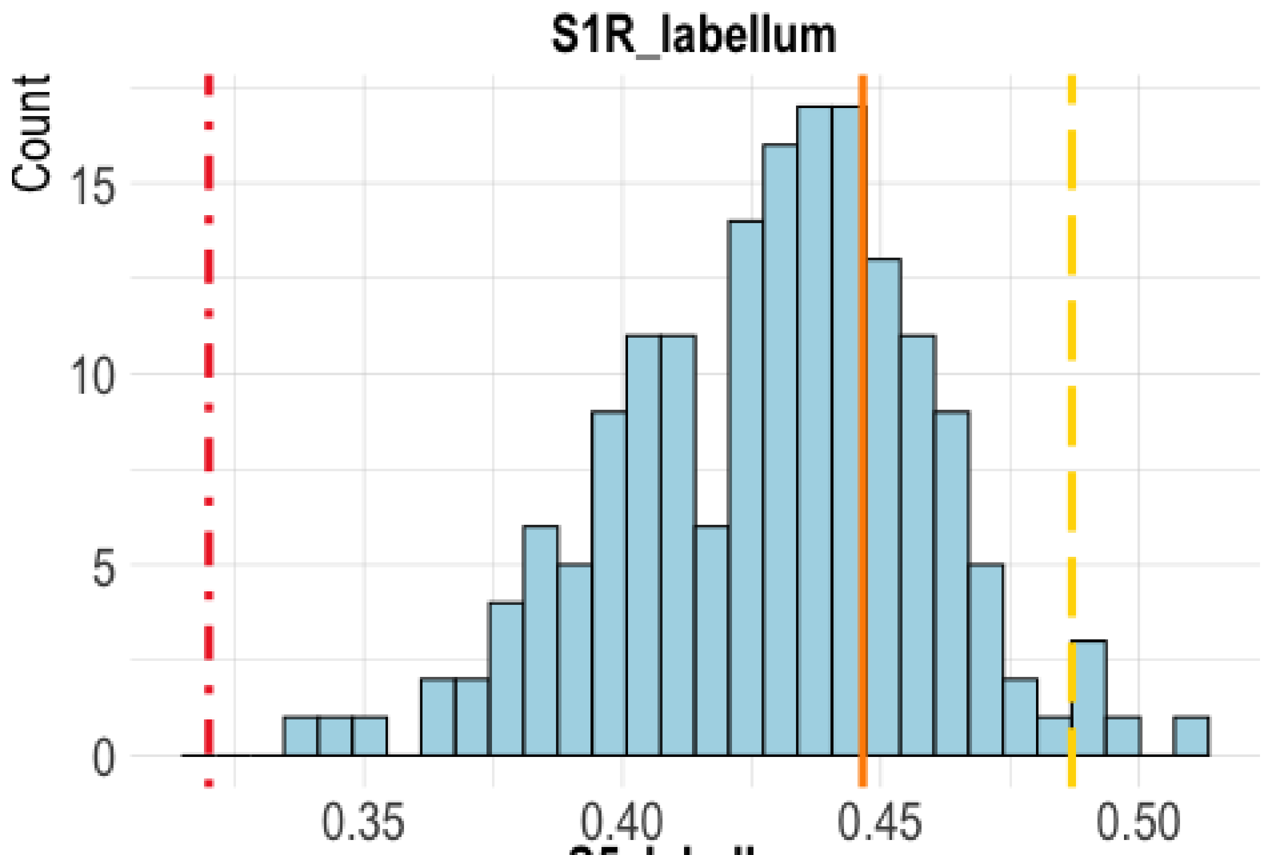

S9_petal
Count
60
40
20
0
0.5
0.6
0.7
0.8
0.9
1.0

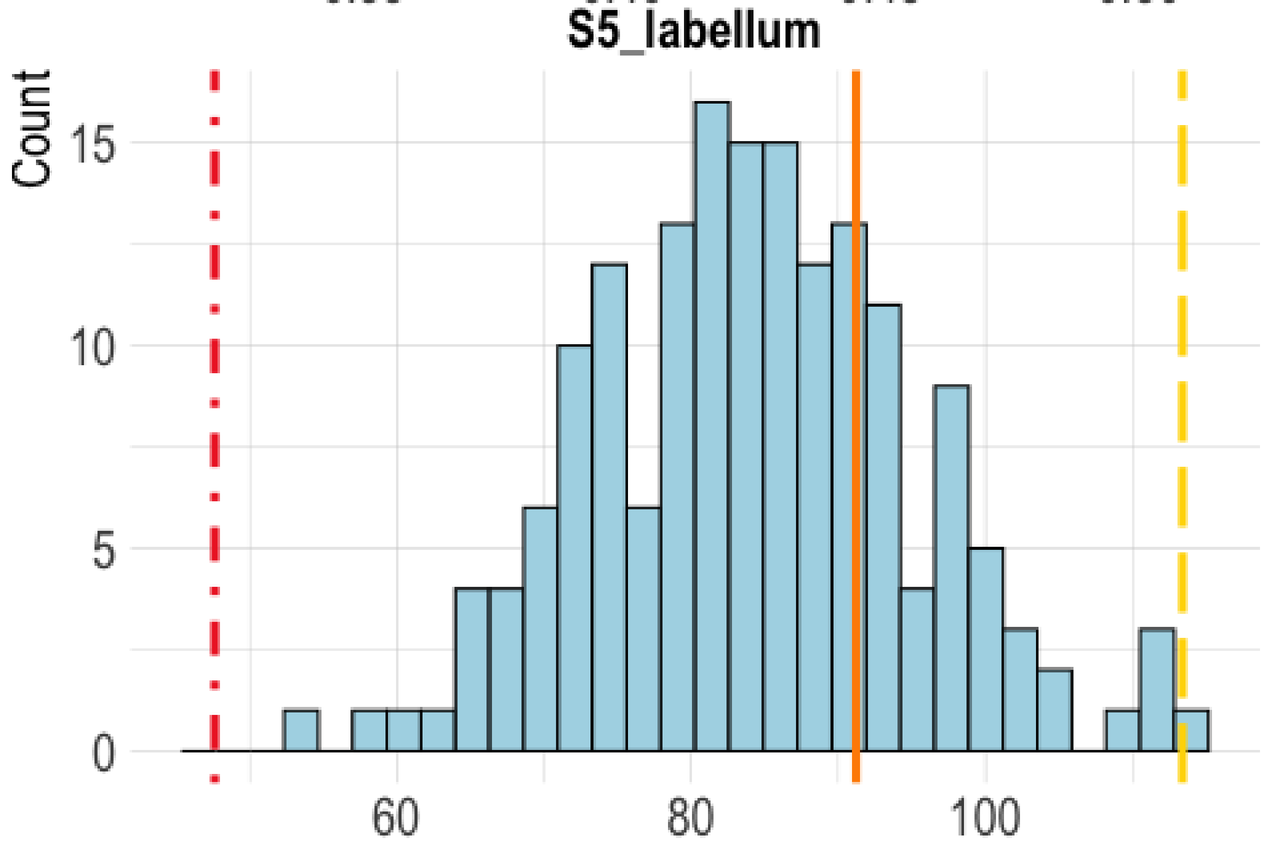

H4_petal
Count
30
20
10
0
0.6
0.7
0.8
0.9
1.0
1.1

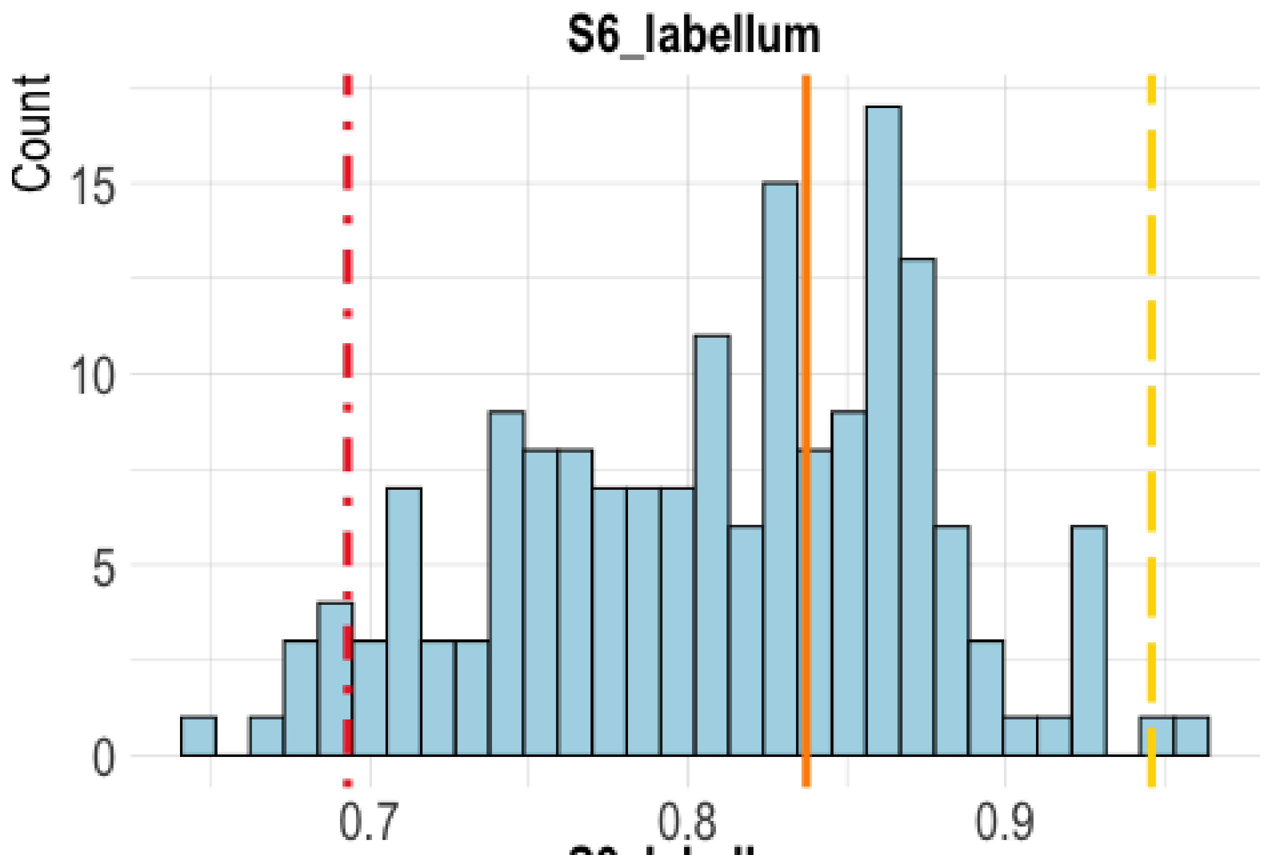

B3_labellum
Count
15
10
5
0
0.7
0.8
0.9

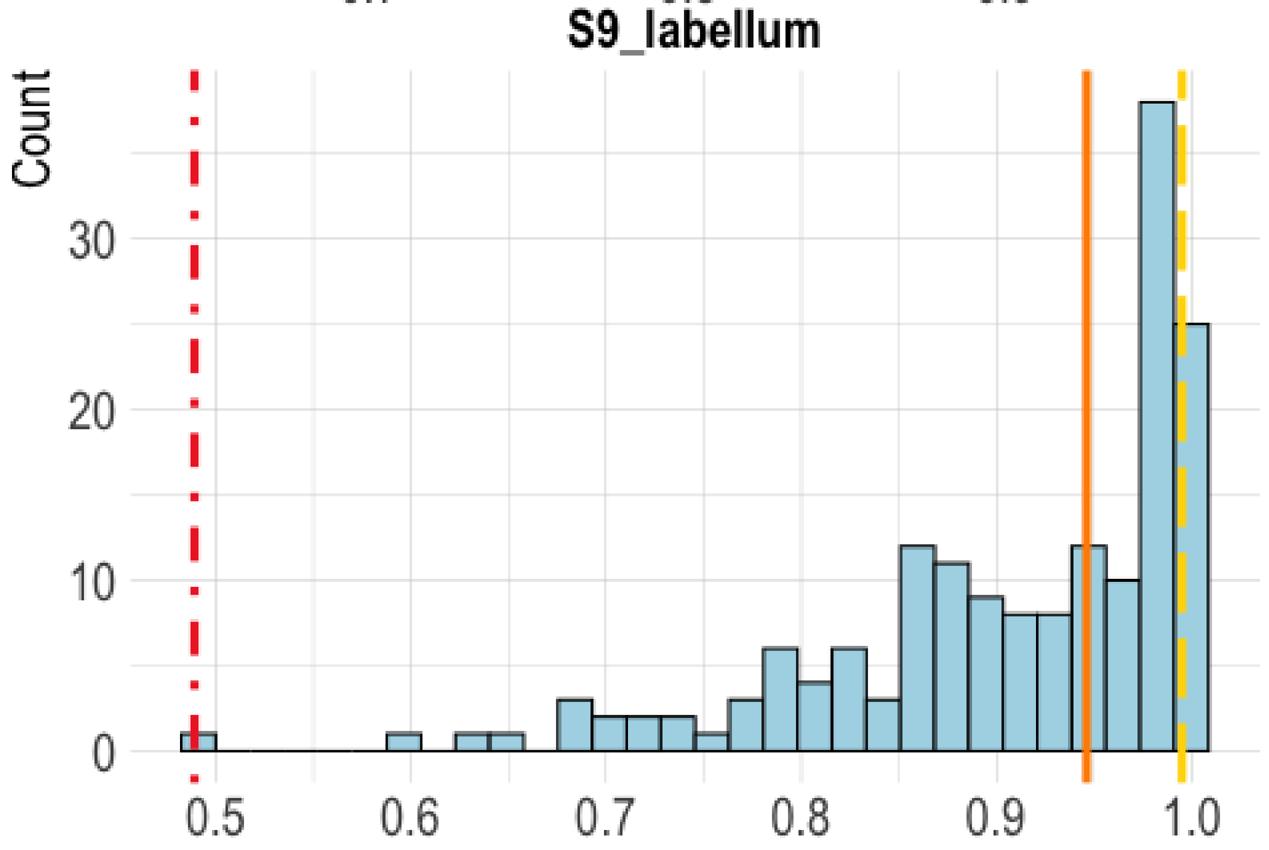

S1B_labellum
Count
0
5
10
15
20
0.05
0.10
0.15
0.20
0.25

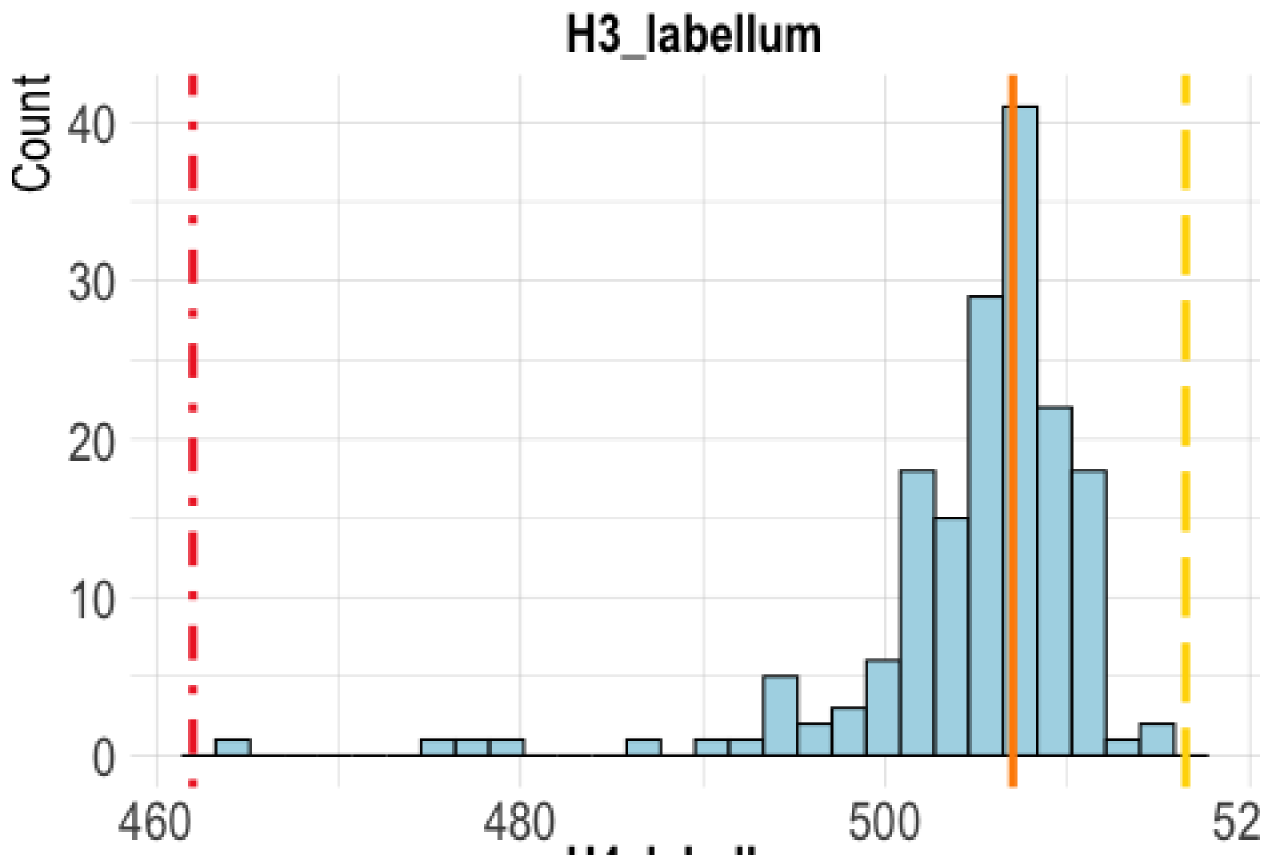

S1Y_labellum
Count
0
5
10
15
0.28
0.32
0.36

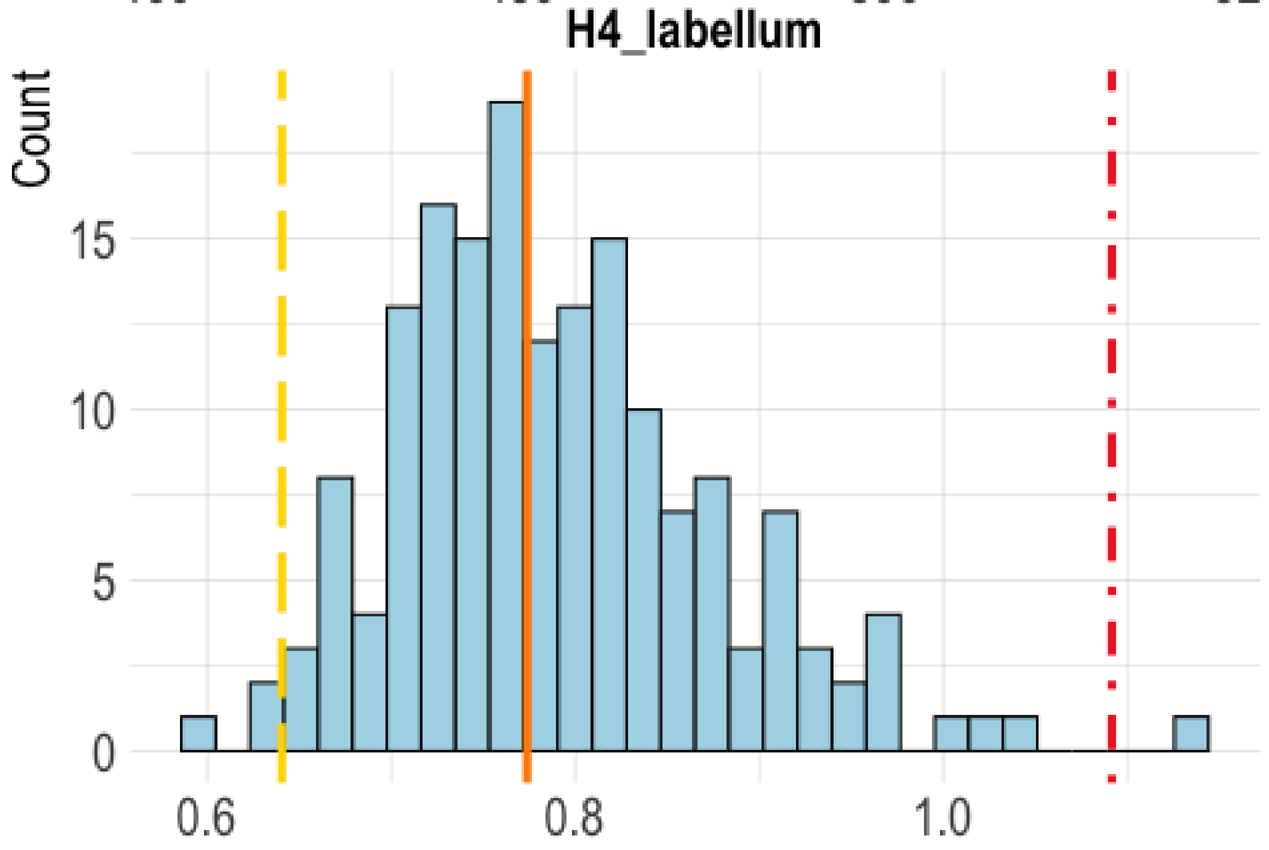

S1R_labellum
Count
0
5
10
15
0.35
0.40
0.45
0.50


S5_labellum
Count
0
5
10
15
60
80
100


S6_labellum
Count
15
10
5
0
0.7
0.8
0.9


S9_labellum
Count
30
20
10
0
0.5
0.6
0.7
0.8
0.9
1.0


H3_labellum
Count
0
10
20
30
40
460
480
500
52


H4_labellum
Count
0
5
10
15
0.6
0.8
1.0

Combines plots

```
# 6. Create Legend Plot
legend_data <- data.frame(
  x = c(1, 2, 3),
  Category = c("C. lasius",
               "C. bracteatus",
               "F1 hybrid")
)

legend_plot <- ggplot(legend_data, aes(x = x)) +
  geom_vline(aes(xintercept = x, color = Category, linetype = Category), size = 1.2) +
  scale_color_manual(
    name = NULL,
    values = c(
      "C. lasius" = "gold",
      "C. bracteatus" = "firebrick2",
      "F1 hybrid" = "darkorange"
    )
  ) +
  scale_linetype_manual(
    name = NULL,
    values = c(
      "C. lasius Parental Species" = "longdash",
      "C. bracteatus Parental Species" = "dotdash",
      "F1 Hybrids" = "solid"
    )
  ) +
  theme_void() +
  theme(
    legend.position = "bottom",
    legend.text = element_text(size = 12)
  )

# 7. Combine Histograms into a Grid
nrow_grid <- 5 # Adjust based on number of histograms and desired layout

histogram_grid <- plot_grid(
  plotlist = histograms,
  nrow = nrow_grid,
  align = "v"
)

# 8. Combine Histogram Grid with Legend Plot
combined_plot <- plot_grid(
  histogram_grid,
  legend_plot,
  ncol = 1,
  rel_heights = c(1, 0.05) # Adjust relative heights as needed
)

# 9. Save the Combined Plot to a PDF
CairoPDF(file = "~/Dropbox/Costus/costus-genetic-mapping/phenotype/results/figures/colormetric_combined
print(combined_plot)
```

```r
dev.off()
```

```
pdf 
  2 
```

```r
# Optional: Display the Combined Plot in R Session
# print(combined_plot)
```