

Range Weighted Branch Length Difference (RWiBaLD) Calculations & Figure Generation

Nunzio Knerr

2025-10-24

Table of contents

1	Introduction	2
2	Load packages	2
3	The RWiBaLD metric	2
4	Specify biodiverse results data	3
5	Load data & calculate RWiBaLD	3
6	Elbow point calculation function	3
7	Calculate branches of interest	5
8	Generate figure 2A - RWiBaLD branches of interest	6
9	Calculate RWiBaLD branch categories (neo-endemic, meso-endemic, paleo-endemic)	7
10	Load RWiBaLD results & tree data by cells	10
11	Generate figure 2B - ranked RWiBaLD score by RWiBaLD score	11
12	Generate figure 2C - ranked BaLD by BaLD (non range weighted)	13
13	Generate multi-panel figure 2ABC	14
14	Generate Figure 3A	17
15	Generate Figure 3B	18
16	Generate multi-panel figure 3AB	20
17	Read tree file & generate supplementary figure 1	22
18	Generate tree in 2 halves supplementary figure 1A and 1B	22
19	Load data & generate histograms for all cells	27
20	Load data & generate 12 histograms for figure 4A-L	28
21	Generate multi-panel histogram figure 4A-L	31
22	Load results data & generate data files for biome maps	33
23	Two tailed relative phylogenetic diversity (RPD) & CANAPE functions	34
24	Load CANAPE & randomisation results	35

25 Generate CANAPE map figure 5A	36
26 Generate biome map with neo-endemics figure 5B	38
27 Generate biome map with meso-endemics figure 5C	40
28 Generate biome map with paleo-endemics figure 5D	42
29 Generate 4 up map figure 5ABCD	44
30 Table of RWiBaLD results supplementary table 1	46

1 Introduction

This document contains code for calculating and plotting Range Weighted Branch Length Difference (RWiBaLD). It uses input from [Biodiverse](#) analyses of phylogenetic data. The method is related to the [Categorical Analysis of Neo And Paleo-Endemism \(CANAPE\)](#) published in 2014 and follows up on that paper. Consequently, the [acacia dataset from the CANAPE paper](#) is used to illustrate the method.

2 Load packages

Here we load the packages for use in subsequent code.

```

1 library(sf)
2 library(ggplot2)
3 library(Cairo)
4 library(extrafont)
5 library(dplyr)
6 library(ggrepel)
7 library(stringr)
8 library(patchwork)
9 library(forcats)
10 library(ggtree)
11 library(ape)
12 library(knitr)
13 library(gt)

```

3 The RWiBaLD metric

The steps for calculating RWiBaLD are as follows:

1. For each branch on the phylogeny (terminal or internal), calculate its *range weighted branch length difference* (RWiBaLD) score as the difference between its length on the RWoT and the RWcT. These scores can be used directly as a continuous measure of neo- and paleo-endemism (negative and positive values, respectively). Classification into neo-endemic, paleo-endemic, and meso-endemic requires further processing using steps 2-4:
2. Identify the “highly endemic branches,” i.e., those with range sizes below a threshold. We calculate this threshold by identifying the “elbow” of the distribution, using the *maximum Euclidean distance* method described by [Ramer \(1972\)](#). This is done by ranking all branches according to the inverse of their range size (functionally the same as their lengths on the RWcT), plotting a straight line from the first to last points on that curve, then identifying the point on the curve that produces the longest perpendicular line drawn to it from the straight line (Fig 2A). Branches greater than or equal to the threshold are considered the highly endemic branches of interest.
3. The same elbow statistic is then applied to the RWiBaLD scores from step 1. The positive and negative differences are processed separately (divided at the RWiBaLD = 0 value), resulting in two thresholds (Fig 2B).

4. The highly endemic branches identified in step 2 are then classified using the thresholds from step 3. Highly endemic branches with negative differences less than or equal to the threshold are classified as neo-endemic, those with positive differences greater than or equal to the threshold are classified as paleo-endemic, and those between these two categories are classified as meso-endemic (Fig 2B).

4 Specify biodiverse results data

Specify the tabular data files exported from a biodiverse analysis. Both the observed data & the equal branch length data from a range weighted tree.

```

1 # Set the data locations etc.
2 data_dir <- "Acacia_biodiverse_exports/"
3 observed_data_csv <- paste0(data_dir, "Acacia_RWT_tabular_export.csv")
4 equal_branch_length_data_csv <- paste0(data_dir, "Acacia_RWT_EQBL_tabular_export.csv")

```

5 Load data & calculate RWiBaLD

Load the data in, merge the tables and calculate the RWiBaLD score for each branch, then write out the results to a file.

```

1 # Load data
2 observed_data <- read.table(observed_data_csv, header=T, sep=",")
3 equal_branch_length_data <- read.table(equal_branch_length_data_csv, header=T, sep=",")
4
5 # Calculate RWiBaLD Statistic etc.
6 rwibald_results <- observed_data |>
7   left_join(equal_branch_length_data, by = 'NAME') |>
8   rename(branch_length_observed_tree = LENGTH.x, branch_length_comparison_tree = LENGTH.y) |>
9   mutate(rank_comparison_tree = rank(desc(branch_length_comparison_tree), ties.method = "min")) |>
10  mutate(sorted_asc_bl_comparison_tree = rank(branch_length_comparison_tree, ties.method = "first")) |>
11  mutate(rwibald_score = (branch_length_observed_tree - branch_length_comparison_tree)) |>
12  mutate(rwibald_score_rank = rank(rwibald_score, ties.method = "first"))
13
14 # View(rwibald_results)
15 output_dir <- "quarto_outputs/"
16
17 # Write data to file
18 write.csv(rwibald_results, paste0(output_dir, "Acacia_RWiBaLD_results.csv"), row.names=FALSE)

```

6 Elbow point calculation function

We propose a function to calculate the elbow point of a curve in order to determine branches of interest and identify the different categories of RWiBaLD (e.g., neo, meso, paleo).

It takes a numerical vector data as input and calculates the point where the rate of decrease in a measure of fit (e.g., sum of squared distances) starts to slow down. The function first orders the values of the data and calculates vectors from the first point to all other points, normalizing the vector from the first to the last point. It then computes the Euclidean distance of each point to this line and returns the y-coordinate (value) of the point with the maximum distance from the line, considered the “elbow” point threshold.

The procedure is described step by step below:

1. Function Definition and Input:

- The function `get_elbow` takes one argument, `data`, which is a numerical vector.

2. Initial Setup:

- `n_pts` is assigned the length of `data`.

- `x_coords` is a sequence from 1 to `n_pts`.
- `y_coords` is the values of data, sorted in ascending order.

3. First Point Coordinates:

- `x1` and `y1` are the coordinates of the first point in the sequence.

4. Normalize the Line Vector:

- `x_vec` and `y_vec` are vectors from the first point to all other points.
- `x_vec_max` and `y_vec_max` are the vectors from the first point to the last point.
- `normaliser` is the length of the line vector.
- Normalize `x_vec_max` and `y_vec_max` by dividing by `normaliser`.

5. Vectors from First Point:

- `v_x` and `v_y` are vectors from the first point to all other points.
- `scalar_prod` calculates the scalar projection of `v_x` and `v_y` onto the normalized line vector.

6. Calculate Distance to Line:

- `vec_to_line_x` and `vec_to_line_y` are the components of the vectors from each point to the line.
- `dist_to_line` is the Euclidean distance of each point to the line.
- `i_max` is the index of the maximum distance (the elbow)

7. Return the Elbow Point Threshold:

- The function returns the coordinates (`x,y`) of the point with the maximum distance from the line, considered the “elbow” point threshold.

In summary, the function identifies the point in the data set that is farthest away from a line drawn between the first and last data points, which often corresponds to a significant change. In case of ties we take the point closest to the minimum.

see: Ramer, U. (1972). *An iterative procedure for the polygonal approximation of plane curves*. *Computer Graphics and Image Processing*, 1, 244–256. [https://doi.org/10.1016/S0146-664X\(72\)80017-0](https://doi.org/10.1016/S0146-664X(72)80017-0)

```

1 get_elbow <- function(data){
2
3   n_pts = length(data)
4
5   x_coords = 1:n_pts
6   y_coords = data[order(data)]
7
8   # First points
9   x1 = x_coords[1]
10  y1 = y_coords[1]
11
12  # Normalize the line vector
13  x_vec = x_coords - x1
14  y_vec = y_coords - y1
15  x_vec_max = x_vec[n_pts]
16  y_vec_max = y_vec[n_pts]
17
18  normaliser = sqrt(x_vec_max^2 + y_vec_max^2)
19
20  x_vec_max = x_vec_max / normaliser
21  y_vec_max = y_vec_max / normaliser
22
23  # Vectors from first point
24  v_x = x_coords - x1

```

```

25     v_y = y_coords - y1
26
27     scalar_prod = v_x * x_vec_max + v_y * y_vec_max
28
29     vec_to_line_x = v_x - scalar_prod * x_vec_max
30     vec_to_line_y = v_y - scalar_prod * y_vec_max
31
32     # Distance to line is the norm
33     dist_to_line = sqrt(vec_to_line_x^2 + vec_to_line_y^2)
34
35     # y value at the point of maximum distance from the hypotenuse
36     #return(y_coords[which(dist_to_line==max(dist_to_line))])
37
38     # Find index of the maximum distance (the elbow)
39     i_max <- which.max(dist_to_line)
40
41     # Return both coordinates
42     return(list(x = x_coords[i_max], y = y_coords[i_max]))
43 }
```

7 Calculate branches of interest

Now we take the data and determine the key branches of interest using the elbow point statistic defined in the function above.

```

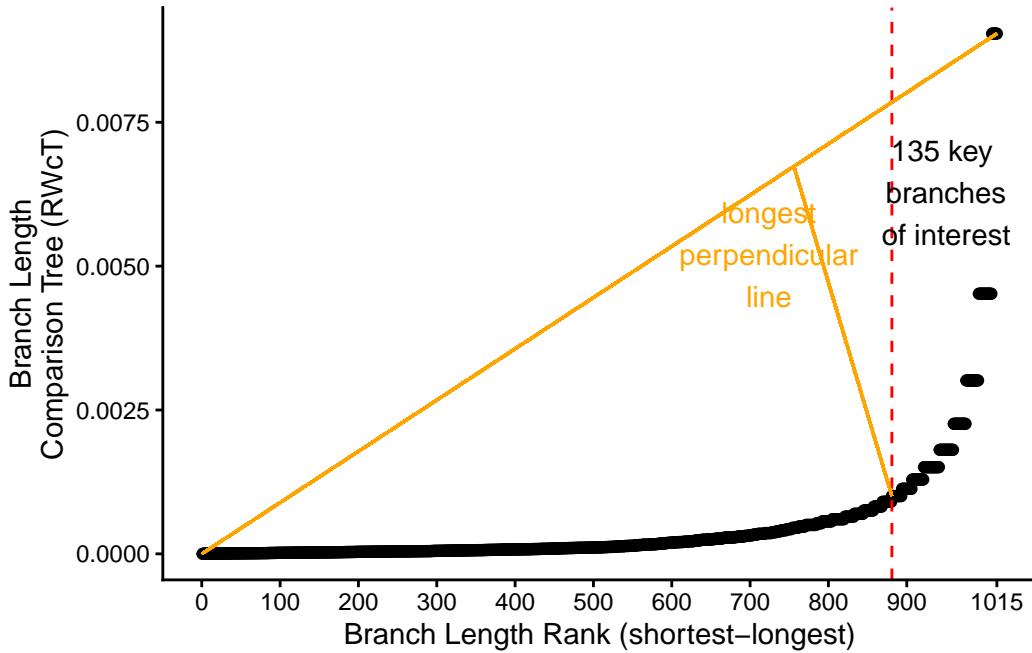
1 # For range weighted tree
2 df <- rwibald_results[,c("NAME", "rwibald_score", "rwibald_score_rank",
3                         "sorted_asc_bl_comparison_tree",
4                         "branch_length_comparison_tree",
5                         "branch_length_observed_tree")] %>%
6   arrange(., branch_length_comparison_tree)
7
8 # Call the get_elbow function to find the threshold
9 elbow_thresh <- get_elbow(df$branch_length_comparison_tree)
10
11 # Define key branches
12 key_branches <- df$branch_length_comparison_tree >= elbow_thresh$y
13
14 # Count them
15 number_of_key_branches <- sum(key_branches)
16
17 # Add a column to the results marking the branches of interest as 'sig'
18 rwibald_results_key <- rwibald_results %>%
19   arrange(., branch_length_comparison_tree) %>%
20   mutate(bl_comparison_tree_rwibald_elbow_key = ifelse(key_branches, 'key', 'not_key'))
21
22 # Merge data
23 rwibald_results_all <- merge(rwibald_results, rwibald_results_key[, 
24   c("NAME", "bl_comparison_tree_rwibald_elbow_key")], by=c('NAME','NAME'), all.x=T)
25
26 #View(rwibald_results_all)
27
28 # Print number of key branches
print(paste0("Number of key branches: ", number_of_key_branches))
```

[1] "Number of key branches: 135"

8 Generate figure 2A - RWiBaLD branches of interest

Now we plot the ranked branch lengths on the comparison tree showing the cut-off line for the key RWiBaLD branches of interest.

```
1 # Check the total number of branches
2 #length(df$branch_length_comparison_tree)
3
4 # Add x_marks every 100
5 x_marks <- c(0, 100, 200, 300, 400, 500, 600, 700, 800, 900, length(df$branch_length_comparison_tree))
6
7 branch_length_by_rank <- ggplot(
8   df,
9   aes(x = as.numeric(rownames(df)), y = sort(branch_length_comparison_tree)))
10 ) +
11   geom_point() +
12   geom_vline(
13     xintercept = (length(df$branch_length_comparison_tree) - number_of_key_branches + 1),
14     color = "red", linetype = "dashed"
15   ) +
16   geom_segment(x = 0, y = 0, xend = 1015, yend = 0.0090413023,
17     color = "orange", linewidth = 0.5, inherit.aes = FALSE) +
18   geom_segment(x = 881, y = 0.001004589, xend = 755.55, yend = 0.00675, color = "orange", linewidth =
19     0.5, inherit.aes = FALSE) +
20   annotate("text", x = 724, y = 0.0052, label = "longest\nperpendicular\nline",
21     color = "orange") +
22   xlab("Branch Length Rank (shortest-longest)") +
23   ylab("Branch Length\nComparison Tree (RWcT)") +
24   theme_classic() +
25   theme(plot.title = element_text(hjust = 0.5)) +
26   scale_x_continuous(breaks = x_marks, labels = x_marks) +
27   annotate("text", x = 945, y = 0.0072,
28     label = paste0(number_of_key_branches, " key\nbranches\nof interest"),
29     vjust = 1, hjust = 0.5)
30
31 print(branch_length_by_rank)
```



```

1 output_dir <- "quarto_outputs/"
2
3 ggsave(paste0(output_dir, "figures/Figure2_A.png"), plot = branch_length_by_rank, scale = 1,
4     width = 3000,
5     height = 1354,
6     units = "px",
7     dpi = 300)

```

9 Calculate RWiBaLD branch categories (neo-endemic, meso-endemic, paleo-endemic)

Next we take the key branches of interest and define which of them are neo-endemic, meso-endemic or paleo-endemic. We use the same elbow statistic defined above but only consider the branches of key interest calculated above as valid. We split the data at 0 with those equal or below the elbow point threshold in the negative data being neo-endemics. Those in the positive dataset equal or above the elbow point threshold being classified as paleo-endemic, and rest are classified as meso-endemic. We have described three categories, but of course one could break the continuous distribution into further categories if required.

```

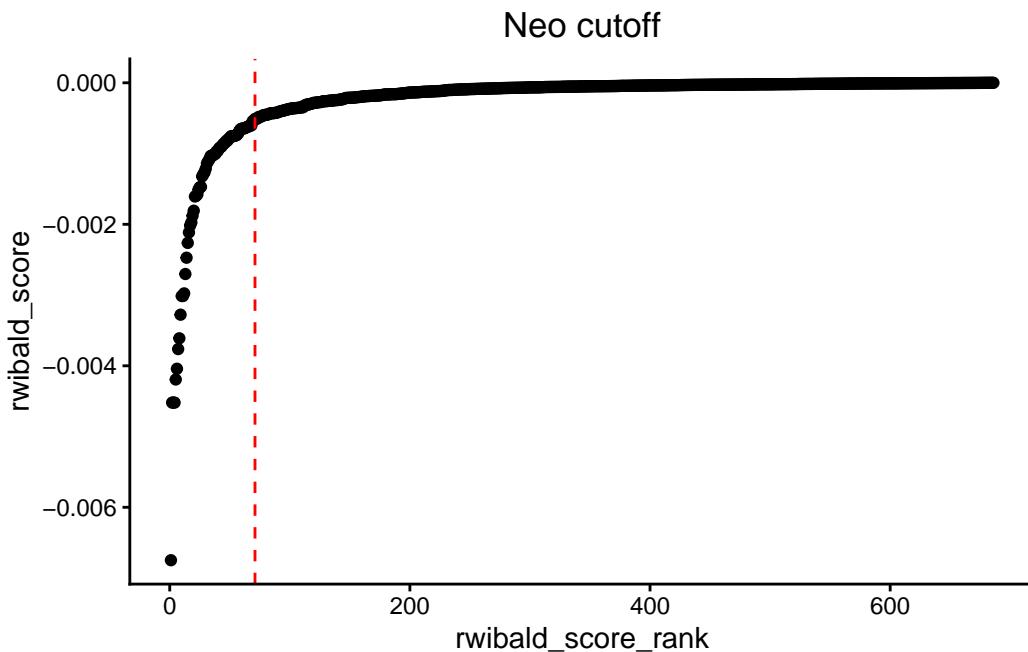
1 # Load the data
2 data <- rwibald_results_all[,c("NAME", "bl_comparison_tree_rwibald_elbow_key", "rwibald_score",
3   ↪ "rwibald_score_rank")] %>%
4   arrange(., rwibald_score_rank)
5
6 # Subset the left and right dataframes
7 subset_left_df <- data[data$rwibald_score <= 0, ]
8 subset_right_df <- data[data$rwibald_score >= 0, ]
9
10 # Create a list to store the results for the table
11 results <- list()
12
13 # Records in negative data
14 negative_records <- nrow(subset_left_df)
15 results$Negative_Records <- negative_records
16

```

```

17 left_point_x <- get_elbow(subset_left_df$rwibald_score)
18 rwibald_score_rank_at_left_point <- subset_left_df[subset_left_df$rwibald_score == left_point_x$y,
19   ↪ "rwibald_score_rank"]
20 results$Neo_Cutoff <- rwibald_score_rank_at_left_point
21
22 # Plot for left half of curve
23 l <- ggplot(subset_left_df, aes(x = rwibald_score_rank, y = rwibald_score)) +
24   geom_point() +
25   geom_vline(xintercept = rwibald_score_rank_at_left_point, color = "red", linetype = "dashed") +
26   ggtitle(label= "Neo cutoff") +
27   theme_classic() +
28   theme(plot.title = element_text(hjust = 0.5))
29 print(l)

```



```

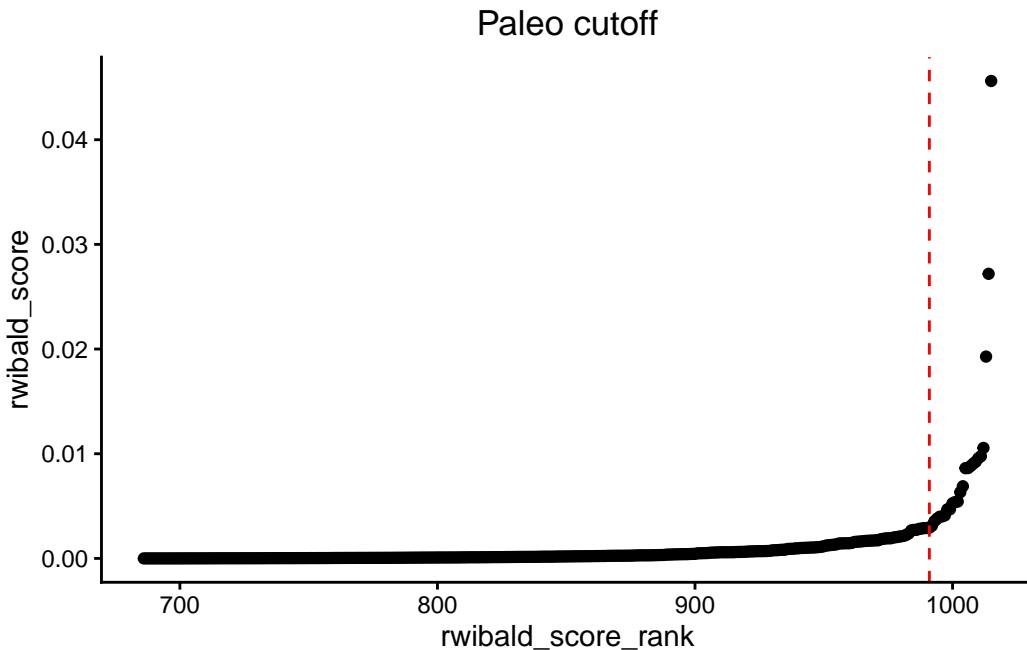
1 ggsave(paste0(output_dir, "figures/Figure2B_L.png"), plot = l, scale = 1,
2   width = 2048,
3   height = 1024,
4   units = "px",
5   dpi = 300)
6
7 # Records in positive data
8 positive_records <- nrow(subset_right_df)
9 results$Positive_Records <- positive_records
10
11 # Paleo cutoff
12 right_point_x <- get_elbow(subset_right_df$rwibald_score)
13 rwibald_score_rank_at_right_point <- subset_right_df[subset_right_df$rwibald_score == right_point_x$y,
14   ↪ "rwibald_score_rank"]
15 results$Paleo_Cutoff <- rwibald_score_rank_at_right_point
16
17 # Plot for right half of curve
18 r <- ggplot(subset_right_df, aes(x = rwibald_score_rank, y = rwibald_score)) +
19   geom_point() +
20   geom_vline(xintercept = rwibald_score_rank_at_right_point, color = "red", linetype = "dashed") +
21   ggtitle(label= "Paleo cutoff") +
22   theme_classic() +
23   theme(plot.title = element_text(hjust = 0.5))
24 print(r)

```

```

20   ggtile(label= "Paleo cutoff") +
21   theme_classic() +
22   theme(plot.title = element_text(hjust = 0.5))
23 print(r)

```



```

1 ggsave(paste0(output_dir, "figures/Figure2B_R.png"), plot = r, scale = 1,
2       width = 2048,
3       height = 1024,
4       units = "px",
5       dpi = 300)
6
7 # Create a column with the results
8 rwibald_results_key_type <- rwibald_results_all %>%
9   filter(bl_comparison_tree_rwibald_elbow_key == "key") %>%
10  mutate(rwibald_type = case_when(rwibald_score_rank <= rwibald_score_rank_at_left_point ~ "neo-endemic",
11    ~ rwibald_score_rank >= rwibald_score_rank_at_right_point ~
12      "paleo-endemic",
13      TRUE ~ "meso-endemic")) %>%
14  select(NAME, rwibald_type)
15
16 # Convert the results list to a dataframe for gt
17 summary_results_df <- as.data.frame(t(sapply(results, c)))
18 results_gt <- gt(data = summary_results_df)
19
20 # Display the table
21 results_gt %>%
22   tab_header(title = "Summary of Records and Cutoffs") %>%
23   fmt_number(columns = everything(), decimals = 0) %>%
24   tab_style(style = cell_text(alignment = "center"),
25             locations = cells_body())
26
27 # View(rwibald_results_key_type)

```

Summary of Records and Cutoffs

Negative_Records	Neo_Cutoff	Positive_Records	Paleo_Cutoff
686	71	330	991

```
28 # Merge the data
29 rwibald_results_all <- merge(rwibald_results_all, rwibald_results_key_type[, c("NAME", "rwibald_type")],
30   by=c('NAME', 'NAME'), all.x=T)
31
32 # View(rwibald_results_all)
33 output_dir <- "quarto_outputs/"
34
35 # Write data to file
36 write.csv(rwibald_results_all, paste0(output_dir, "Acacia_RWiBaLD_results_all.csv"), row.names=FALSE)
```

10 Load RWiBaLD results & tree data by cells

Now we load the data for the RWiBaLD results and the range weighted tree data in tabular format and then replace the counts in the matrix with the branch lengths for that group so we can generate the histograms for each grid cell location. We also add the CANAPE scores to the data table for comparison and calculate the ranges of each branch.

```
1 data_dir <- "Acacia_biodiverse_exports/"
2 output_dir <- "quarto_outputs/"
3
4 # Data with internal branches (spatial analysis from BD) x grid cells
5 all_tree_data_csv <- paste0(data_dir, "Acacia_PD_Included_Node_List.csv")
6 RWiBaLD_results_csv <- paste0(output_dir, "Acacia_RWiBaLD_results_all.csv")
7
8 all_tree_data <- read.table(all_tree_data_csv, header=T, sep=",", check.names = FALSE )
9 RWiBaLD_results <- read.table(RWiBaLD_results_csv, header=T, sep=",")
10
11 #View(all_tree_data)
12 #View(RWiBaLD_results)
13 valueToUse <- "rwibald_score"
14
15 # Iterate over the columns in Table 1
16 for (col in colnames(all_tree_data)[-c(1:3)]) {
17   # get the corresponding "valueToUse" column specified above from Table 2
18   branch_length <- RWiBaLD_results[RWiBaLD_results$NAME == col, paste0(valueToUse)]
19   # replace the numbers in Table 1 with the "valueToUse" value
20   all_tree_data[, col] <- ifelse(is.na(all_tree_data[, col]), NA, as.numeric(branch_length))
21 }
22
23 # View(all_tree_data)
24
25 # Write data to file
26 write.csv(all_tree_data, paste0(output_dir, "Acacia_PD_Included_Node_List_", valueToUse, ".csv"),
27   row.names=FALSE)
28
29 # Get CANAPE data from biodiverse as well
30 canape_csv <- paste0(data_dir, "Acacia_Rand1_CANAPE_Export.csv")
31 canape_results <- read.table(canape_csv, header=T, sep=",")
32
33 # View(canape_results)
```

```

33
34 canape_group_data <- canape_results %>%
35   left_join(all_tree_data %>%
36     select(-c(Axis_0, Axis_1)), by = "ELEMENT"
37   )
38
39 # Transpose the dataframe
40 data <- t(canape_group_data)
41
42 # View(data)
43
44 # Set the first row as column names
45 transposed_df <- setNames(data.frame(data[-1,]), data[1,])
46
47 # View(transposed_df)
48
49 # Merge CANAPE and RWiBaLD data
50 canape_group_data_rwibald <- transposed_df %>%
51   mutate(range_cell_count = rowSums(!is.na(.))) %>%
52   tibble::rownames_to_column(var = "NAME") %>%
53   left_join(select(rwibald_results_all, NAME, rwibald_type), by = 'NAME') %>%
54   mutate(rwibald_type = ifelse(is.na(rwibald_type), "other", rwibald_type))
55
56 # Set row names
57 rownames(canape_group_data_rwibald) <- canape_group_data_rwibald$NAME
58
59 # View(canape_group_data_rwibald)
60
61 # Add range data to results and create three new columns
62 rwibald_results_all_with_range <- rwibald_results_all %>%
63   left_join(select(canape_group_data_rwibald, NAME, range_cell_count), by = "NAME") %>%
64   mutate(rwibald_type = replace(rwibald_type, is.na(rwibald_type), "other"),
65         non_range_weighted_branch_length_observed_tree = branch_length_observed_tree * range_cell_count,
66         non_range_weighted_branch_length_comparison_tree = branch_length_comparison_tree *
67           ↪ range_cell_count,
68         diff_non_range_weighted_observed_to_comparison_bl =
69           ↪ non_range_weighted_branch_length_observed_tree -
70           ↪ non_range_weighted_branch_length_comparison_tree
71       ) %>%
72   mutate(rank_BaLD = rank(diff_non_range_weighted_observed_to_comparison_bl, ties.method = "first"))
73
74
75 # Write data to file
76 write.csv(rwibald_results_all_with_range, paste0(output_dir,
77   ↪ "Acacia_RWiBaLD_results_all_with_range.csv"), row.names=FALSE)
78
79 # View(rwibald_results_all_with_range)
80 # View(rwibald_results_all)

```

11 Generate figure 2B - ranked RWiBaLD score by RWiBaLD score

Here we plot the ranked RWiBaLD score (x) by the RWiBaLD score (y) (Figure 2B), colouring the different RWiBaLD Categories: neo-endemic (red), meso-endemic (darkolivegreen4) and paleo-endemic (royalblue1).

```

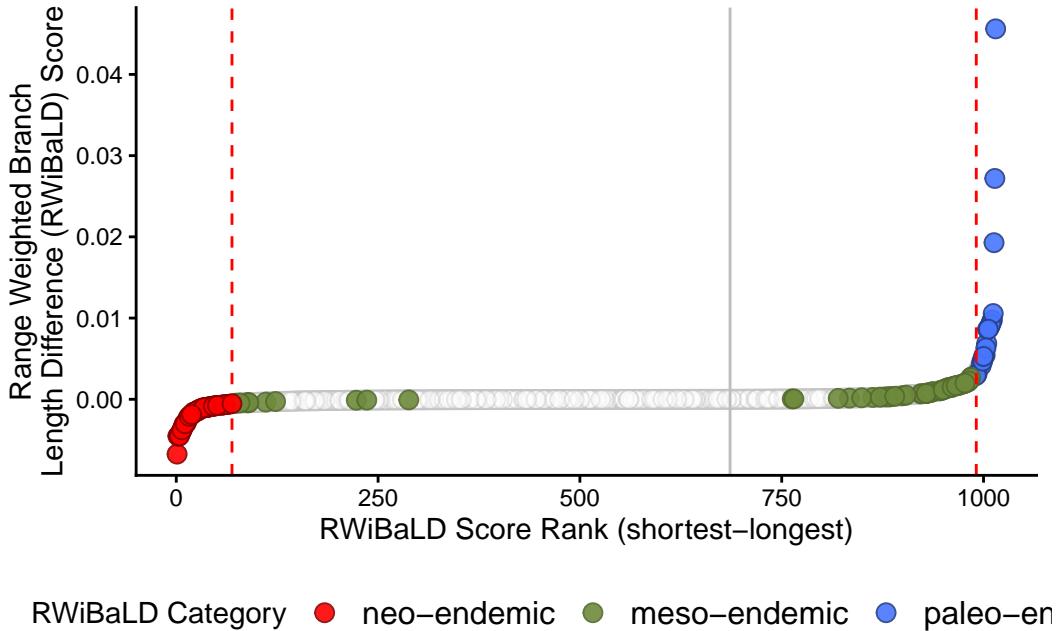
1 # Find the neo_cutoff
2 neo_cutoff <- rwibald_results_all_with_range %>%
3   filter(rwibald_type == "neo-endemic") %>%

```

```

4   summarize(highest_rank = max(rwibald_score_rank, na.rm = TRUE)) %>%
5   pull(highest_rank)
6
7 # Find the paleo_cutoff
8 paleo_cutoff <- rwibald_results_all_with_range %>%
9   filter(rwibald_type == "paleo-endemic") %>%
10  summarize(lowest_rank = min(rwibald_score_rank, na.rm = TRUE)) %>%
11  pull(lowest_rank)
12
13 # Get only the rwibald branches of key interest
14 rwibald_results_all_key_only <- rwibald_results_all_with_range %>%
15   filter(bl_comparison_tree_rwibald_elbow_key == "key")
16
17 # View(rwibald_results_all_key_only)
18 # View(rwibald_results_all_with_range)
19
20 # Plot rwibald rank X rwibald score
21 rwibald_key_plot <- ggplot() +
22   geom_point(data=rwibald_results_all_with_range, aes(x = rwibald_score_rank, y = rwibald_score),
23   ↪ fill="transparent", size = 3, colour= "grey77", alpha = 0.4, pch = 21) +
24   geom_point(data = rwibald_results_all_key_only, aes(x = rwibald_score_rank, y = rwibald_score, fill =
25   ↪ rwibald_type, color = rwibald_type), size = 3, alpha = 0.9, pch = 21) +
26   scale_fill_manual(name = "RWiBaLD Category", values = c("paleo-endemic" = "royalblue1", "neo-endemic" =
27   ↪ "red", "meso-endemic" = "darkolivegreen4"), labels = c("neo-endemic","meso-endemic",
28   ↪ "paleo-endemic"), limits = c("neo-endemic","meso-endemic", "paleo-endemic"), na.value =
29   ↪ "transparent") +
30   scale_color_manual(name = "RWiBaLD Category", values = c("paleo-endemic" = "royalblue4", "neo-endemic"
31   ↪ = "red4", "meso-endemic" = "darkolivegreen"), labels = c("neo-endemic", "meso-endemic",
32   ↪ "paleo-endemic"), limits = c("neo-endemic", "meso-endemic", "paleo-endemic"), na.value =
33   ↪ "transparent") +
34   geom_vline(xintercept = neo_cutoff, color = "red", linetype = "dashed") +
35   geom_vline(xintercept = paleo_cutoff, color = "red", linetype = "dashed") +
36   xlab("RWiBaLD Score Rank (shortest-longest)") +
37   ylab("Range Weighted Branch\nLength Difference (RWiBaLD) Score") +
#annotate("segment", x = results$Negative_Records, xend = results$Negative_Records,
38   ↪ y = max(rwibald_results_all_with_range$rwibald_score) * 1.1, yend = 0,
39   ↪ arrow = arrow(length = unit(0.2, "cm")), color = "grey74") +
40   geom_vline(xintercept = results$Negative_Records, color = "grey74", linetype = "solid") +
41   theme_classic() +
42   theme(plot.title = element_text(hjust = 0.5), legend.position="bottom", legend.text = element_text(size
43   ↪ = 12))
44
45 print(rwibald_key_plot)

```



```

1 ggsave("quarto_outputs/figures/Figure2_B.png", plot = rwibald_key_plot, scale = 1,
2   width = 3000,
3   height = 1354,
4   units = "px",
5   dpi = 300)

```

12 Generate figure 2C - ranked BaLD by BaLD (non range weighted)

Illustrating the distribution of branch lengths on the observed tree with no range-weighting; the Y-axis is the difference between length on the observed tree and length on the comparison tree for each branch; the X-axis is ranked branch length difference (shortest - longest). The branches of interest are colored by RWiBaLD category as in 2B

```

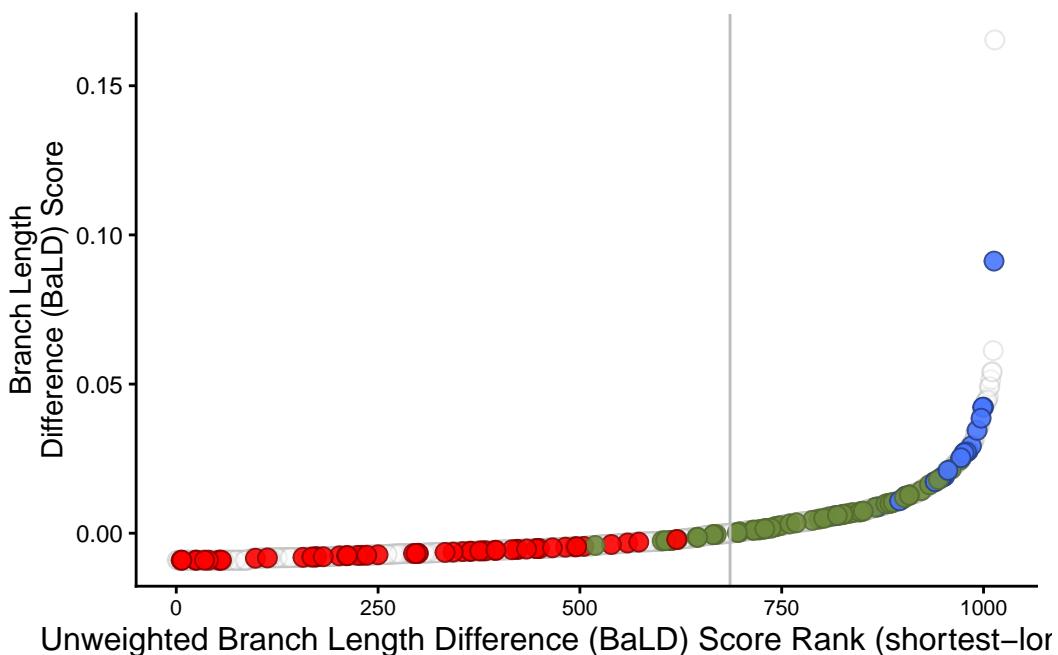
1 #View(rwibald_results_all_with_range)
2
3 #View(rwibald_results_all_key_only)
4
5 # Plot non_range_weighted score
6 rwibald_key_plot_non_range_weighted <- ggplot() +
7   #geom_point(data=rwibald_results_all_with_range, aes(x =
8   #  reorder(non_range_weighted_branch_length_observed_tree,
8   #  non_range_weighted_branch_length_observed_tree), y =
8   #  diff_non_range_weighted_observed_to_comparison_bl), fill="transparent", size = 3, colour=
8   #  "grey77", alpha = 0.4, pch = 21) +
9   geom_point(data=rwibald_results_all_with_range, aes(x = rank_BaLD, y =
9   #  diff_non_range_weighted_observed_to_comparison_bl), fill="transparent", size = 3, colour= "grey77",
9   #  alpha = 0.4, pch = 21) +
10  geom_point(data = rwibald_results_all_key_only, aes(x = rank_BaLD, y =
10  #  diff_non_range_weighted_observed_to_comparison_bl, fill = rwibald_type, color = rwibald_type), size
10  #  = 3, alpha = 0.9, pch = 21) +
11  scale_fill_manual(name = "RWiBaLD Category", values = c("paleo-endemic" = "royalblue1", "neo-endemic" =
11  #  "red", "meso-endemic" = "darkolivegreen4"), labels = c("neo-endemic","meso-endemic",
11  #  "paleo-endemic"), limits = c("neo-endemic","meso-endemic", "paleo-endemic"), na.value =
11  #  "transparent") +

```

```

11 scale_color_manual(name = "RWiBaLD Category", values = c("paleo-endemic" = "royalblue4", "neo-endemic"
12   = "red4", "meso-endemic" = "darkolivegreen"), labels = c("neo-endemic", "meso-endemic",
13   "paleo-endemic"), limits = c("neo-endemic", "meso-endemic", "paleo-endemic"), na.value =
14   "transparent") +
15 #geom_vline(xintercept = neo_cutoff, color = "red", linetype = "dashed") +
16 #geom_vline(xintercept = paleo_cutoff, color = "red", linetype = "dashed") +
17 xlab("Unweighted Branch Length Difference (BaLD) Score Rank (shortest-longest)") +
18 ylab("Branch Length\ndifference (BaLD) Score") +
19 geom_vline(xintercept = results$Negative_Records, color = "grey74", linetype = "solid") +
20 theme_classic() +
21 theme(plot.title = element_text(hjust = 0.5), legend.position="none", legend.text = element_text(size =
22   12), axis.title.x = element_text(size = 12), axis.text.x = element_text(size = 8))
23
24 print(rwibald_key_plot_non_range_weighted)

```



```

1 ggsave("quarto_outputs/figures/Figure2_C.png", plot = rwibald_key_plot_non_range_weighted, scale = 1,
2   width = 3000,
3   height = 1354,
4   units = "px",
5   dpi = 300)

```

13 Generate multi-panel figure 2ABC

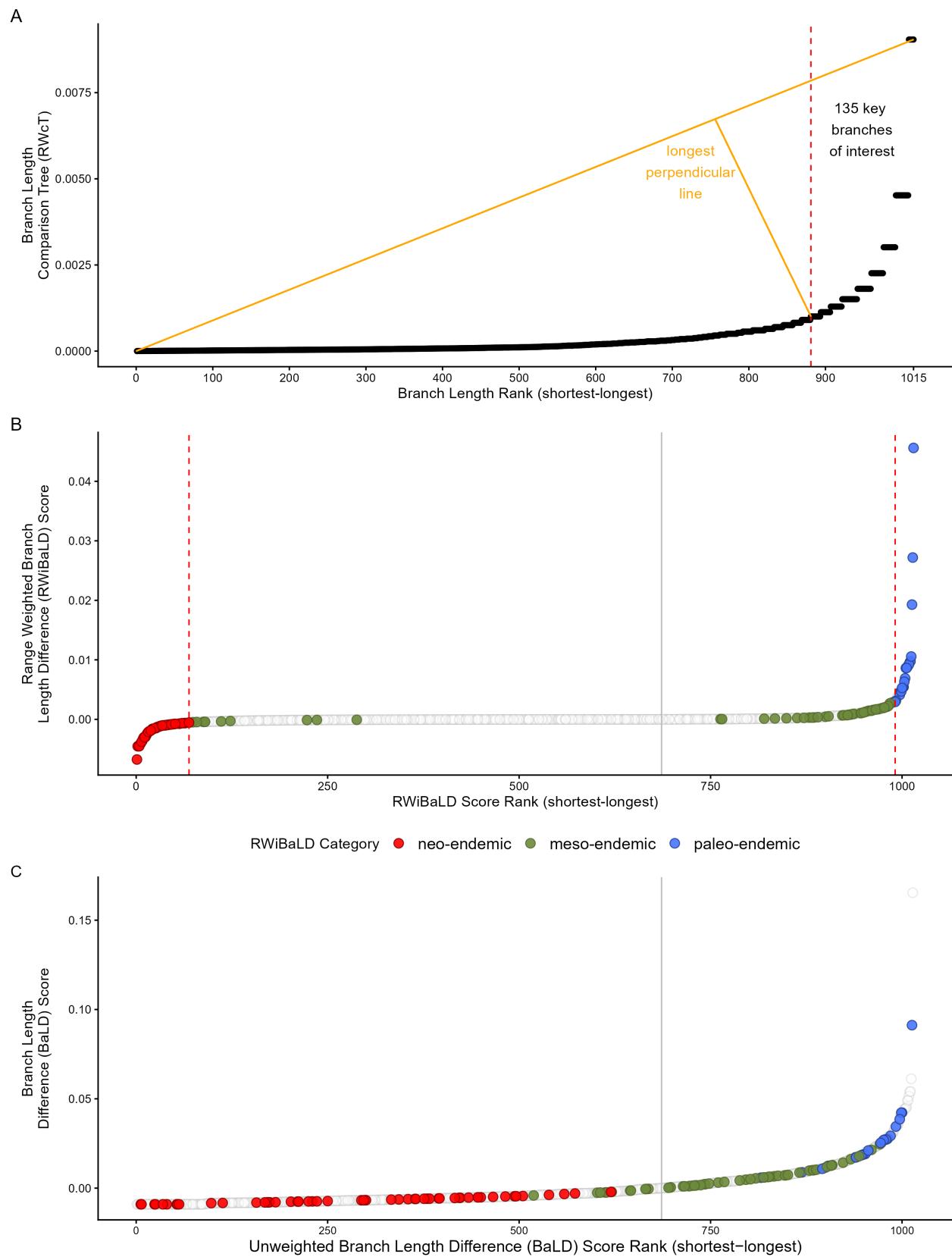
Here we generate a 2 up figure using patchwork

```

1 patchwork <- branch_length_by_rank / rwibald_key_plot / rwibald_key_plot_non_range_weighted +
2   plot_annotation(
3     title = 'Branches of Interest for Range Weighted Branch Length Difference (RWiBaLD)\n&\nRWiBaLD Scores
4     With Categories Calculated Using Elbow Statistic',
5     theme = theme(plot.title = element_text(hjust = 0.5)),
6     subtitle = '',
7     caption = '',
8     ...
9   )

```

```
6 tag_levels = 'A') +
7   theme(plot.tag.position = c(0, 1),
8     plot.tag = element_text(size = 12, hjust = 0, vjust = 0))
9
10 #print(patchwork)
11 #4062
12 ggsave("quarto_outputs/figures/Figure2_ABC.png", patchwork, width = 3000, height = 4062, units = "px")
```



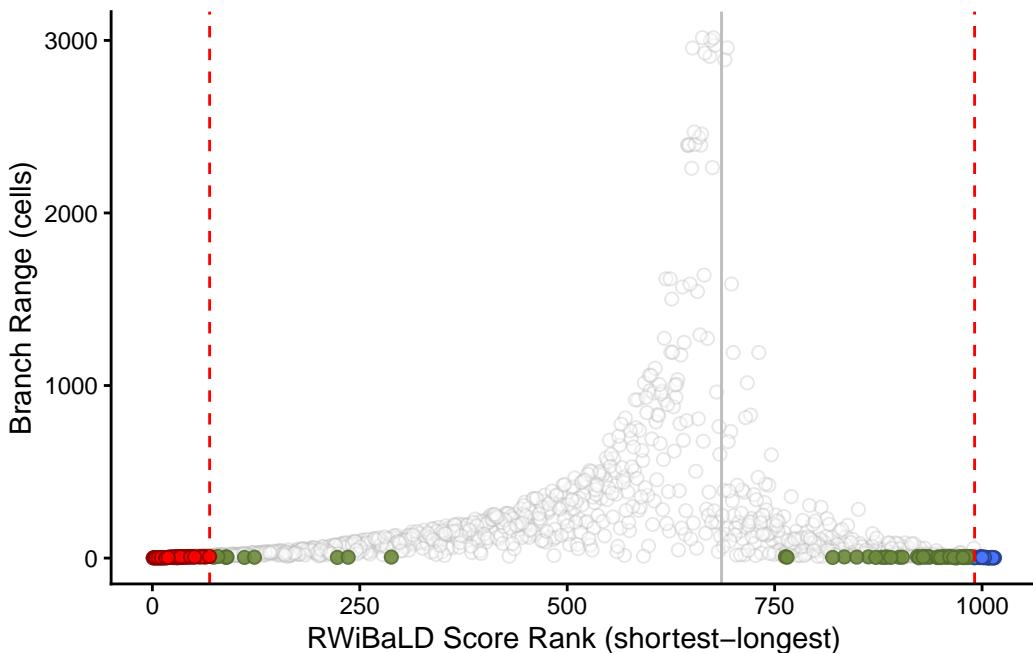
14 Generate Figure 3A

Here we create a plot of RWiBaLD score (x) by branch range (cells) (Figure 3A), coloured by RWiBaLD Categories: neo-endemic (red), meso-endemic (green) and paleo-endemic (blue).

```

1 # Plot RWiBaLD rank X branch range
2 rwibald_key_plot_range <- ggplot() +
3   geom_point(data=rwibald_results_all_with_range, aes(x = rwibald_score_rank, y = range_cell_count),
4   ↪   fill="transparent", size = 2, colour= "grey77", alpha = 0.4, pch = 21) +
5   geom_point(data = rwibald_results_all_key_only, aes(x = rwibald_score_rank, y = range_cell_count, fill
6   ↪   = rwibald_type, color = rwibald_type), size = 2, alpha = 0.9, pch = 21) +
7   scale_fill_manual(name = "RWiBaLD Category", values = c("paleo-endemic" = "royalblue1","neo-endemic" =
8   ↪   "red", "meso-endemic" = "darkolivegreen4"), labels = c("neo-endemic","meso-endemic",
9   ↪   "paleo-endemic"), limits = c("neo-endemic","meso-endemic", "paleo-endemic"), na.value =
10  ↪   "transparent") +
11  scale_color_manual(name = "RWiBaLD Category", values = c("paleo-endemic" = "royalblue4", "neo-endemic"
12  ↪   = "red4", "meso-endemic" = "darkolivegreen"), labels = c("neo-endemic", "meso-endemic",
13  ↪   "paleo-endemic"), limits = c("neo-endemic", "meso-endemic", "paleo-endemic"), na.value =
14  ↪   "transparent") +
15  geom_vline(xintercept = neo_cutoff, color = "red", linetype = "dashed") +
16  geom_vline(xintercept = paleo_cutoff, color = "red", linetype = "dashed") +
17  geom_vline(xintercept = results$Negative_Records, color = "grey74", linetype = "solid") +
18  xlab("RWiBaLD Score Rank (shortest-longest)") +
19  ylab("Branch Range (cells)") +
20  theme_classic() +
21  theme(plot.title = element_text(hjust =1),
22    legend.position="none",
23    legend.text = element_text(size = rel(1)),
24    legend.title = element_text(size = rel(1)),
25    axis.title.x = element_text(size = rel(1)),
26    axis.title.y = element_text(size = rel(1)),
27    axis.text.x = element_text(size = rel(1)),
28    axis.text.y = element_text(size = rel(1)))
29
30 print(rwibald_key_plot_range)

```



```

1 ggsave("quarto_outputs/figures/Figure3_A.png", plot = rwibald_key_plot_range, scale = 1,
2       width = 2048,
3       height = 1024,
4       units = "px",
5       dpi = 300)

```

15 Generate Figure 3B

Now the same data as figure 3A but first removing all internal branches before plotting the data. Note: the neo / meso /paleo cutoffs are calculated on the whole dataset and the filtering to terminals done after that.

```

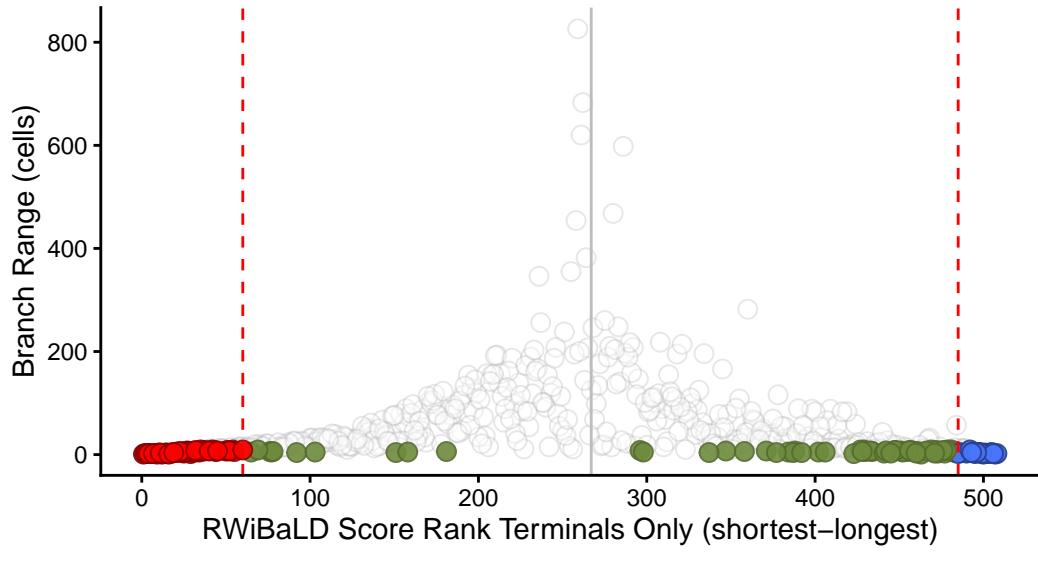
1 # For plotting purposes get only the rwibald branches of key interest and terminals only
2 rwibald_results_all_key_only_terminals <- rwibald_results_all_with_range %>%
3   filter(!grepl("^[0-9]", NAME)) %>%
4   mutate(rwibald_score_rank_terminals = rank(rwibald_score, ties.method = "first")) %>%
5   filter(bl_comparison_tree_rwibald_elbow_key == "key")
6
7 rwibald_results_all_with_range_terminals <- rwibald_results_all_with_range %>%
8   filter(!grepl("^[0-9]", NAME)) %>%
9   mutate(rwibald_score_rank_terminals = rank(rwibald_score, ties.method = "first"))
10
11 # Find the neo_cutoff
12 neo_cutoff_terms <- rwibald_results_all_with_range_terminals %>%
13   filter(rwibald_type == "neo-endemic") %>%
14   summarize(highest_rank = max(rwibald_score_rank_terminals, na.rm = TRUE)) %>%
15   pull(highest_rank)
16
17 # Find the paleo_cutoff
18 paleo_cutoff_terms <- rwibald_results_all_with_range_terminals %>%
19   filter(rwibald_type == "paleo-endemic") %>%
20   summarize(lowest_rank = min(rwibald_score_rank_terminals, na.rm = TRUE)) %>%
21   pull(lowest_rank)
22
23 # Subset the left and right dataframes
24 subset_left_df_terms <-
25   ↳ rwibald_results_all_with_range_terminals[rwibald_results_all_with_range_terminals$rwibald_score <= 0,
26   ↳ ]
27
28 # Records in negative data
29 negative_records_terms <- nrow(subset_left_df_terms)
30
31 # View(rwibald_results_all_key_only_terminals)
32 # View(rwibald_results_all_with_range_terminals)
33
34 # View(rwibald_results_all_with_range)
35
36 # Plot rwibald rank x rwibald score
37 # rwibald_key_plot_terminals <- ggplot() +
38 #   geom_point(data=rwibald_results_all_with_range_terminals, aes(x = rwibald_score_rank_terminals, y =
39 #     ↳ rwibald_score), fill="transparent", size =3, colour= "grey77", alpha = 0.4, pch=21) +
40 #   geom_point(data = rwibald_results_all_key_only_terminals, aes(x = rwibald_score_rank_terminals, y =
41 #     ↳ rwibald_score, fill = rwibald_type, color = rwibald_type), size =3, alpha = 0.9, pch=21) +
42 #   scale_fill_manual(name = "RWiBaLD Category", values = c("paleo-endemic" = "royalblue1", "neo-endemic"
43 #     ↳ = "red", "meso-endemic" = "darkolivegreen4"), labels = c("neo-endemic","meso-endemic",
44 #     ↳ "paleo-endemic"), limits = c("neo-endemic","meso-endemic", "paleo-endemic"), na.value =
45 #     ↳ "transparent") +
46 #   scale_color_manual(name = "RWiBaLD Category", values = c("paleo-endemic" = "royalblue4",
47 #     ↳ "neo-endemic" = "red4", "meso-endemic" = "darkolivegreen"), labels = c("neo-endemic", "meso-endemic",
48 #     ↳ "paleo-endemic"), limits = c("neo-endemic", "meso-endemic", "paleo-endemic"), na.value =
49 #     ↳ "transparent") +

```

```

40 #   geom_vline(xintercept = neo_cutoff, color = "red", linetype = "dashed") +
41 #   geom_vline(xintercept = paleo_cutoff, color = "red", linetype = "dashed") +
42 #   xlab("RWiBaLD Score Rank Terminals Only (shortest-longest)") +
43 #   ylab("Range Weighted Branch\nLength Difference (RWiBaLD) Score") +
44 #   theme_classic() +
45 #   theme(plot.title = element_text(hjust = 0.5), legend.position="bottom", legend.text =
46 #         element_text(size = 12))
47 #
48 # print(rwibald_key_plot_terminals)
49 #
50 # ggsave("quarto_outputs/figures/Figure3_B_terminals_only.png", plot = rwibald_key_plot_terminals, scale
51 #        = 1,
52 #        width = 2048,
53 #        height = 1024,
54 #        units = "px",
55 #        dpi = 300)
56 #
56 # Plot rwibald rank x branch range
57 rwibald_key_plot_range_terminals <- ggplot() +
58   geom_point(data=rwibald_results_all_with_range_terminals, aes(x = rwibald_score_rank_terminals, y =
59   range_cell_count), fill="transparent", size =3, colour= "grey77", alpha = 0.4, pch=21) +
60   geom_point(data = rwibald_results_all_key_only_terminals, aes(x = rwibald_score_rank_terminals, y =
61   range_cell_count, fill = rwibald_type, color = rwibald_type), size =3, alpha = 0.9, pch=21) +
62   scale_fill_manual(name = "RWiBaLD Category", values = c("paleo-endemic" = "royalblue1","neo-endemic" =
63   "red", "meso-endemic" = "darkolivegreen4"), labels = c("neo-endemic","meso-endemic",
64   "paleo-endemic"), limits = c("neo-endemic","meso-endemic", "paleo-endemic"), na.value =
65   "transparent") +
66   scale_color_manual(name = "RWiBaLD Category", values = c("paleo-endemic" = "royalblue4", "neo-endemic"
67   "red4", "meso-endemic" = "darkolivegreen"), labels = c("neo-endemic", "meso-endemic",
68   "paleo-endemic"), limits = c("neo-endemic", "meso-endemic", "paleo-endemic"), na.value =
69   "transparent") +
70   geom_vline(xintercept = neo_cutoff_terms, color = "red", linetype = "dashed") +
71   geom_vline(xintercept = paleo_cutoff_terms, color = "red", linetype = "dashed") +
72   geom_vline(xintercept = negative_records_terms, color = "grey74", linetype = "solid") +
73   xlab("RWiBaLD Score Rank Terminals Only (shortest-longest)") +
74   ylab("Branch Range (cells)") +
75   theme_classic() +
76   theme(plot.title = element_text(hjust = 0.5), legend.position="bottom", legend.text =
77   element_text(size=rel(1)))
78 #
79 print(rwibald_key_plot_range_terminals)

```



```

1 ggsave("quarto_outputs/figures/Figure3_B.png", plot = rwibald_key_plot_range_terminals, scale = 1,
2   width = 2048,
3   height = 1024,
4   units = "px",
5   dpi = 300)

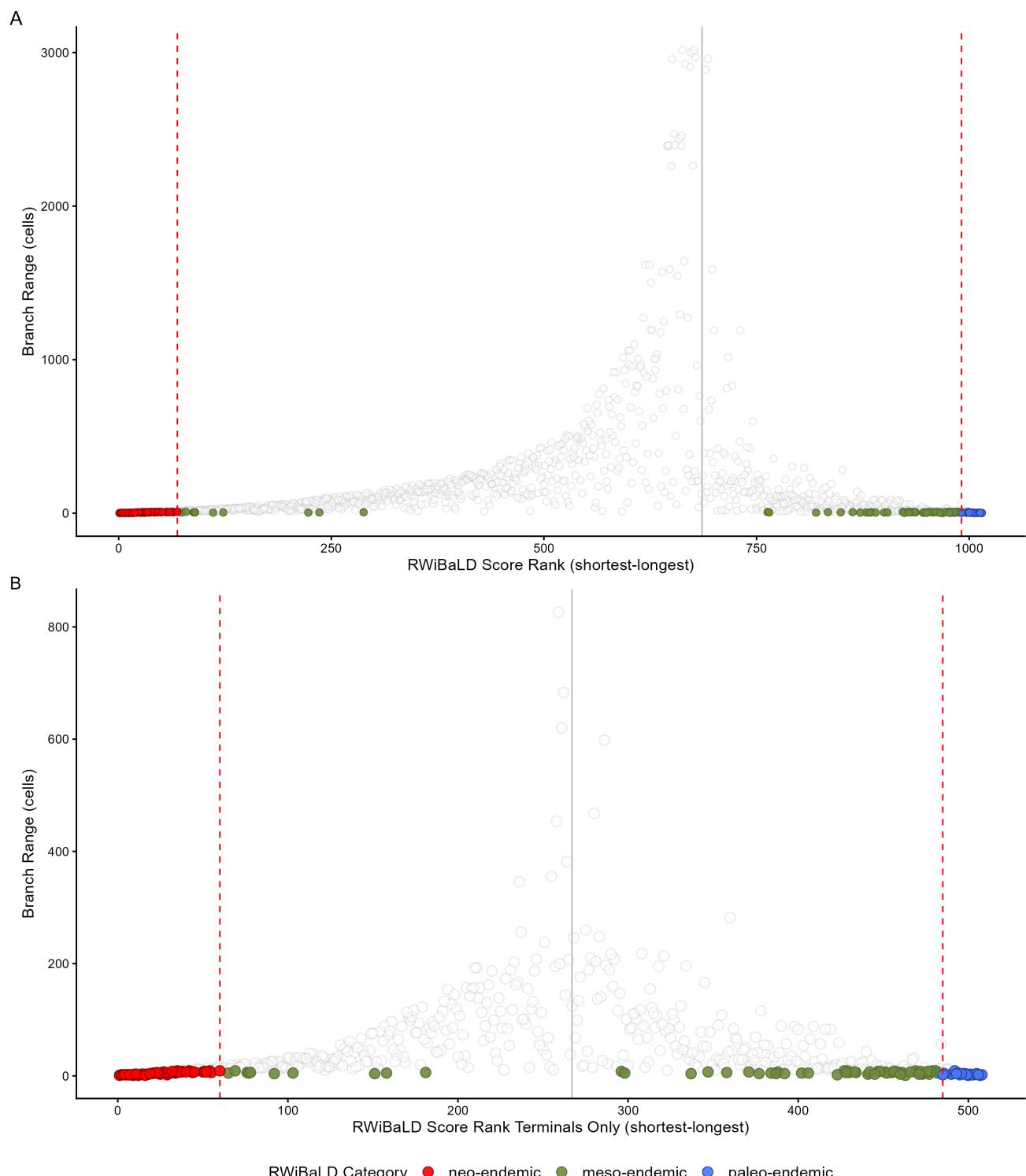
```

16 Generate multi-panel figure 3AB

```

1 patchwork <- rwibald_key_plot_range / rwibald_key_plot_range_terminals + plot_annotation(
2   theme = theme(plot.title = element_text(hjust = 0.5)),
3   subtitle = '',
4   caption = '',
5   tag_levels = 'A') +
6   theme(plot.tag.position = c(0, 1),
7     plot.tag = element_text(size = 12, hjust = 0, vjust = 0))
8
9 #print(patchwork)
10
11 ggsave("quarto_outputs/figures/Figure3_AB.png", patchwork, width = 3000, height = 3600, units = "px")

```



17 Read tree file & generate supplementary figure 1

Now we load the range weighted tree file exported from biodiverse and colour code it by the RWiBaLD categories.

```
1 # if need be check and install ggtree using this code
2 # if (!require("BiocManager", quietly = TRUE))
3 #   install.packages("BiocManager")
4 #
5 # BiocManager::install("ggtree")
6 #
7 data_dir <- "Acacia_biodiverse_exports/"
8 #
9 all_tree_data_nwk <- paste0(data_dir, "acacia_tree_TRIMMED1_EQ1_RW1.nwk")
10 #
11 myTree <- read.tree(file=all_tree_data_nwk)
12 #
13 rwibald_results_all <- rwibald_results_all %>%
14   mutate(rwibald_type = replace(rwibald_type, is.na(rwibald_type), "other"))
15 #
16 # View(rwibald_results_all)
17 #
18 rwibald_results_all$tree_cols <- as.factor(rwibald_results_all$rwibald_type)
19 #
20 # Remove quotes from labels so they match dataframe
21 myTree$tip.label <- gsub("'", "", myTree$tip.label)
22 myTree$node.label <- gsub("'", "", myTree$node.label)
23 #
24 # Create ggtree object
25 g <- ggtree(myTree) %<+% rwibald_results_all +
26   aes(color=tree_cols) +
27   scale_color_manual(values = c("paleo-endemic" = "royalblue1",
28                             "neo-endemic" = "red",
29                             "meso-endemic" = "darkolivegreen")) +
30   # geom_text2(aes(label = node),
31   #            hjust =0,      # adjust horizontal position
32   #            vjust = 0,      # adjust vertical position
33   #            size = 1,       # adjust text size
34   #            color = "blue") + # adjust text color
35   theme(legend.position="none") +
36   geom_tiplab(as_ylab=FALSE, size=1)
37 #
38 # Display the plot
39 #print(g)
40 #
41 ggsave("quarto_outputs/figures/SupFigure1_full_tree.png", g, width = 2048,
42       height = 4024, units = "px")
```

18 Generate tree in 2 halves supplementary figure 1A and 1B

```
1 data_dir <- "Acacia_biodiverse_exports/"
2 all_tree_data_nwk <- paste0(data_dir, "acacia_tree_TRIMMED1_EQ1_RW1.nwk")
3 #
4 myTree <- read.tree(file=all_tree_data_nwk)
5 #
6 rwibald_results_all <- rwibald_results_all %>%
7   mutate(rwibald_type = replace(rwibald_type, is.na(rwibald_type), "other"))
```

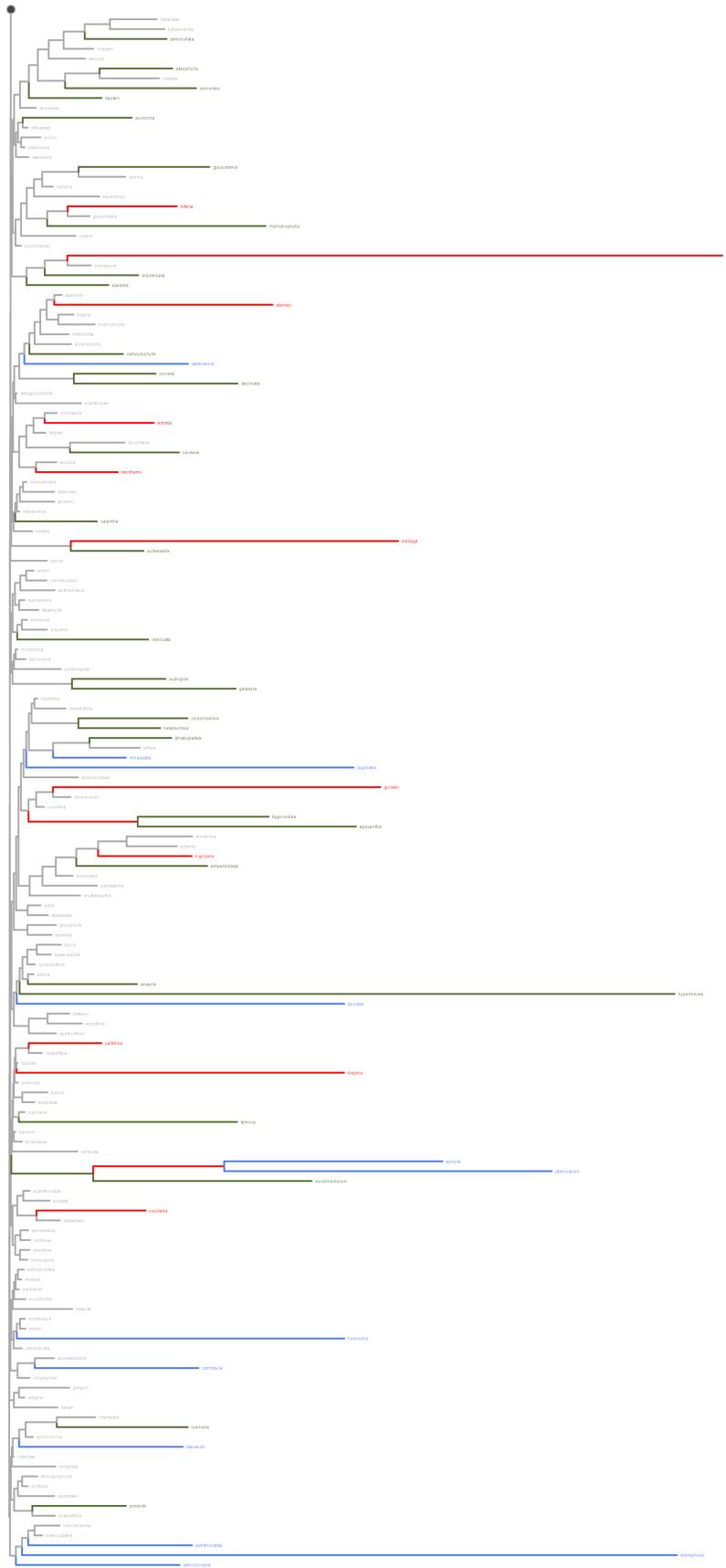
```

8 rwibald_results_all$tree_cols <- as.factor(rwibald_results_all$rwibald_type)
9
10 # Remove quotes from labels so they match the dataframe
11 myTree$tip.label <- gsub("'", "", myTree$tip.label)
12 myTree$node.label <- gsub("'", "", myTree$node.label)
13
14 # Calculate the halfway point
15 half <- length(myTree$tip.label) / 2
16
17 # Sort the tip labels for consistency
18 tip_labels <- myTree$tip.label
19 first_half_tips <- tip_labels[1:half]
20 second_half_tips <- tip_labels[(half + 1):length(tip_labels)]
21
22 # Identify MRCA nodes for each half
23 node_first_half <- getMRCA(myTree, first_half_tips)
24 node_second_half <- getMRCA(myTree, second_half_tips)
25
26
27 # Create the base ggtree object
28 g <- ggtree(myTree) %<+% rwibald_results_all +
29   aes(color = tree_cols) +
30   scale_color_manual(values = c("paleo-endemic" = "royalblue1",
31                             "neo-endemic" = "red",
32                             "meso-endemic" = "darkolivegreen",
33                             "other" = "gray66")) +
34   theme(legend.position = "none") +
35   geom_tiplab(as_ylab = FALSE, size=1)
36
37 #plot(g)
38
39 # Identify MRCA node
40 mrca_node <- MRCA(g, second_half_tips)
41
42 second_half_tree <- viewClade(tree_view = g, mrca_node) +
43   geom_point2(
44     aes(subset = (node == mrca_node)),
45     shape = 21,
46     size = 2,
47     fill = 'gray26',
48     nudge_x = 20,
49     position = position_nudge(y = -88))
50
51
52 # +
53 #   theme(panel.on top = FALSE,
54 #         panel.spacing.x = unit(c(1, 1, 1, 1), "cm"),
55 #         panel.background = element_rect(color = "green", fill="transparent", linewidth = 10),
56 #         plot.margin = unit(c(1, 1, 1, 1), "cm"))
57
58
59 #plot(second_half_tree)
60
61 # Save the figure showing the second half expanded and the first half collapsed
62 ggsave("quarto_outputs/figures/SupFigure_1A.png",
63        second_half_tree, width = 2048, height = 4024, units = "px")
64
65 #plot(g)

```

```
67 # Collapse the second half of the tree
68 g_half_collapsed <- collapse(g, node = node_second_half) +
69   # Add a point to indicate the collapsed node for reference
70   geom_point2(aes(subset=(node == node_second_half)), shape=21, size=2, fill='gray26')
71
72 #plot(g_halfCollapsed)
73
74 # Save the figure showing the first half expanded and the second half collapsed
75 ggsave("quarto_outputs/figures/SupFigure_1B.png",
76       g_half_collapsed, width = 2048, height = 4024, units = "px")
```





19 Load data & generate histograms for all cells

Next we load the data and generate histograms. This code generates a histogram for every cell in the dataset. For convenience the file names contain the CANAPE code and whether the histogram contains branches of key interest.

```
1 #function to change decimal places displayed on plot
2 scaleFUN <- function(x) sprintf("%.8g", x)
3
4 # Function to replace '-' with 'm' and ':' with '_'
5 clean_column_names <- function(df) {
6   new_names <- colnames(df) %>%
7     gsub("-", "neg", .) %>%
8     gsub(":", "_", .)
9   colnames(df) <- new_names
10  return(df)
11 }
12
13 # Clean column names using the function
14 canape_group_data_rwibald <- clean_column_names(canape_group_data_rwibald)
15
16 #View(canape_group_data_rwibald)
17
18 col_scheme <- c("paleo-endemic" = "royalblue1", "neo-endemic" = "red", "meso-endemic" = "darkolivegreen",
19   ↪ "other" = "lightgoldenrodyellow")
20 legend_labels <- c("neo-endemic" = "Neo Endemic", "paleo-endemic" = "paleo Endemic", "other" = "other",
21   ↪ "meso-endemic" = "Meso Endemic")
22
23 #data <- transposed_df
24 #View(canape_key_group_data_rwibald)
25
26 #Generate all histograms
27 for (i in 2:(ncol(canape_group_data_rwibald)-2)) {# skip first column and last 2 as they are
28   ↪ NAME,cell_count,rwibald_type
29 #for (i in 74:74) {
30   #i <- 74
31   data <- canape_group_data_rwibald[-(1:6),] # remove the rows not needed
32   col_name <- colnames(data)[i] # get the column name to generate the histogram of
33   colourByCol <- "rwibald_type" # set the column to use for colouring the histogram
34   number_of_Bins <- 41 # set number of bins to use
35   hist_data <- na.omit(as.numeric(data[[col_name]])) # extract that column data for plotting
36   hist_range <- range(hist_data,na.rm=TRUE) # calculate the range of all values in the column ie. min/max
37   ↪ value
38
39 #calculate the total range of the whole data set to get a consistent x scale on all histograms.
40 dataForRange <- data %>%
41   select(-first(colnames(.)), -last(colnames(.))) %>%
42   mutate(across(everything(), as.numeric))
43 totalRange <- range(dataForRange, na.rm=TRUE) #lowest and highest values in the dataset
44 totalRangeGap <- max(dataForRange, na.rm = TRUE) - min(dataForRange, na.rm = TRUE) # distance between
45   ↪ range above
46
47   range <- max(hist_data) - min(hist_data)
48   binSize <- range/number_of_Bins
49   farthest_number <- max(abs(hist_range)) # calculate the number farthest from 0
50   x_min <- -(farthest_number)-2*(binSize) # set xmin so zero is centred
51   x_max <- (farthest_number)+2*(binSize) # set xmax so zero is centred
52
53 #check to see if any of the branches in the histogram are RWIBALD significant to add to the file names
54 result <- data %>%
```

```

50 select (!!sym(col_name), !!sym(colourByCol)) %>%
51 filter (!is.na (!!sym(col_name))) %>%
52 summarise(result = ifelse(any (!!sym(colourByCol) %in% c("paleo-endemic", "neo-endemic",
53   ↪ "meso-endemic")), "key", "zero-key")) %>%
54 pull(result)
55
56 #View(result)
57 # setup the file names
58 CANAPE_CODE <- canape_group_data_rwibald["CANAPE_CODE", col_name]
59 cell <- str_replace_all(col_name, "[[:]", "_")
60 filename <- paste0("quarto_outputs/histograms/", cell, "_CC_", CANAPE_CODE, "_", result, "_",
61   ↪ numberOfBins, "bins.png")
62
63 #print(filename)
64 # get a subset of datafram for the histogram
65 plot_data <- data %>%
66 select (!!sym(col_name), !!sym(colourByCol)) %>%
67 filter (!is.na (!!sym(col_name))) %>%
68 arrange (!!sym(colourByCol))
69
70 #View(plot_data)
71
72 plot <- ggplot(plot_data, aes(x = as.numeric(.data[[col_name]])), fill = forcats::fct_rev(rwibald_type)))
73   +
74 geom_histogram(color = "black", linewidth= 0.2, bins = numberofBins) +
75 scale_fill_manual(values = col_scheme) +
76 xlim(x_min, x_max) +
77 coord_cartesian(ylim = c(0, 15)) +
78 labs(x = "RWiBaLD Score", y = "Frequency", fill = "Branch Category") +
79 theme_bw()
80
81 plot <- plot +
82 stat_bin(
83   aes(label = after_stat(if_else (condition = count>15, as.character(count), "")),
84   bins= as.numeric(numberofBins),
85   position=position_stack(vjust=0.1),
86   pad=TRUE,
87   geom = "text",
88   color = "black",
89   size = 2,
90   y = 14
91 )
92
93 CairoPNG(width = 1024, height = 600, file = filename, canvas="white", bg = "white", units="px", dpi=96,
94   ↪ title = "")
95 print(plot)
96 dev.off()
97 }
```

20 Load data & generate 12 histograms for figure 4A-L

This code filters out only the 12 histograms included in Figure 4 and generates the ggplot2 objects of them.

```

1 data <- t(canape_group_data)
2 #View(data)
3
4 #Transpose the datafram and set the first row as column names
```

```

5 transposed_df <- setNames(data.frame(data[-1,]), data[1,])
6
7 #View(transposed_df)
8
9 #cells to keep
10 cols_to_keep <- c("NAME", "-1825000:-2975000", "75000:-2525000",
11   ↵ "625000:-3475000", "-1425000:-3225000", "1775000:-3725000", "-1425000:-3475000", "-1225000:-3775000",
12   ↵ "-1425000:-3275000", "1225000:-1275000", "-1175000:-3275000", "-1075000:-2825000",
13   ↵ "-1075000:-2375000")
14
15 #Calculate RWiBaLD significants and Statistics etc.
16 canape_group_data_rwibald <- transposed_df %>%
17   tibble::rownames_to_column(var = "NAME") %>%
18   select(all_of(cols_to_keep)) %>%
19   left_join(select(rwibald_results_all, NAME, rwibald_type), by = 'NAME') %>%
20   mutate(rwibald_type = ifelse(is.na(rwibald_type), "other", rwibald_type))
21
22 # Check if the column is already a factor
23 if (!is.factor(canape_group_data_rwibald$rwibald_type)) {
24   # If not a factor, convert and specify levels
25   canape_group_data_rwibald$rwibald_type <- factor(canape_group_data_rwibald$rwibald_type, levels =
26   ↵ c("neo-endemic", "meso-endemic", "paleo-endemic", "other"))
27 }
28
29 #View(canape_group_data_rwibald)
30
31 # Function to replace '-' with 'm' and ':' with '_'
32 clean_column_names <- function(df) {
33   new_names <- colnames(df) %>%
34     gsub("-", "neg", .) %>%
35     gsub(":", "_", .)
36   colnames(df) <- new_names
37   return(df)
38 }
39
40 # Clean column names using the function
41 canape_group_data_rwibald <- clean_column_names(canape_group_data_rwibald)
42
43 rownames(canape_group_data_rwibald) <- canape_group_data_rwibald$NAME
44
45 #View(canape_group_data_rwibald)
46
47 # Write data to file
48 write.csv(canape_group_data_rwibald, paste0("quarto_outputs/transposed_df_figure_only.csv"),
49   ↵ row.names=FALSE)
50
51 #function to change decimal places displayed on plot
52 scaleFUN <- function(x) sprintf("%.8g", x)
53
54 #View(canape_group_data_rwibald)
55
56 col_scheme <- c("paleo-endemic" = "royalblue1",
57   "neo-endemic" = "red",
58   "meso-endemic" = "darkolivegreen",
59   "other" = "lightgoldenrodyellow"
60   )
61
62 legend_labels <- c("neo-endemic" = "neo-endemic",
63   "paleo-endemic" = "paleo-endemic",
64   )

```

```

59         "meso-endemic" = "meso-endemic",
60         "other" = "other"
61     )
62
63 legend_order <- c("neo-endemic", "meso-endemic", "paleo-endemic", "other")
64
65 #data <- transposed_df
66 #View(canape_group_data_rwibald)
67
68 # Create an empty list to store plots
69 plot_list <- list()
70
71 #Generate histograms
72 for (i in 2:(ncol(canape_group_data_rwibald)-1)) {# skip first column and last 2 as they are
    ↵ NAME,rwibald_type
73   #i <- 2
74   local({ # have to make this local to allow multiple plots in patchwork
75     i <- i
76   data <- canape_group_data_rwibald[-(1:6),] # remove the rows not needed
77   col_name <- colnames(data)[i] # get the column name to generate the histogram of
78   colourByCol <- "rwibald_type" # set the column to use for colouring the histogram
79   numberOfBins <- 41 # set number of bins to use
80
81   # have to show only one legend as the 'collect' feature in patchwork does not collect the legends as
    ↵ desired
82   if(i == 3){
83     showLegend <- TRUE
84   } else {
85     showLegend <- FALSE
86   }
87
88   hist_data <- na.omit(as.numeric(data[[col_name]])) # extract that column data for plotting
89   hist_range <- range(hist_data,na.rm=TRUE) # calculate the range of all values in the column ie. min/max
    ↵ value
90
91   #calculate the total range of the whole data set to get a consistent x scale on all histograms.
92   dataForRange <- data %>%
93     select(-first(colnames(.)), -last(colnames(.))) %>%
94     mutate(across(everything(), as.numeric))
95   totalRange <- range(dataForRange, na.rm=TRUE)#lowest and highest values in the dataset
96   totalRangeGap <- max(dataForRange, na.rm = TRUE) - min(dataForRange, na.rm = TRUE) # distance between
    ↵ range above
97
98   range <- max(hist_data) - min(hist_data)
99   binSize <- range/numberOfBins
100   farthest_number <- max(abs(hist_range)) # calculate the number farthest from 0
101   x_min <- -(farthest_number)-2*(binSize) # set xmin so zero is centred
102   x_max <- (farthest_number)+2*(binSize) # set xmax so zero is centred
103
104   # get a subset of dataframe for the histogram
105   plot_data <- data %>%
106     mutate (!!sym(col_name) := as.numeric (!!sym(col_name))) %>%
107     select (!!sym(col_name), !!sym(colourByCol)) %>%
108     filter (!is.na (!!sym(col_name))) %>%
109     arrange (!!sym(colourByCol))
110
111   #View(plot_data)
112   #str(plot_data)
113

```

```

114 plot <- ggplot(plot_data, aes(x = .data[[col_name]], fill = forcats::fct_rev(rwibald_type))) +
115   geom_histogram(color = "black", linewidth = 0.2, bins = number_ofBins, show.legend = showLegend) +
116   scale_fill_manual(values = col_scheme, labels = legend_labels, breaks = legend_order, drop = FALSE,
117     na.value = "transparent") +
118   #xlim(x_min, x_max) +
119   xlim(-0.011, 0.011) +
120   coord_cartesian(ylim = c(0, 13)) +
121   scale_y_continuous(labels = scales::label_number(accuracy = 1)) +
122   labs(x = "RWiBaLD Score", y = "Frequency", fill = "RWiBaLD Category") +
123   theme_bw()
124
124 plot <- plot +
125   geom_rect(
126     aes(xmin = -0.0005, xmax = 0.0005, ymin = 12.2, ymax = 12.8),
127     color = "white",
128     fill = "white"
129   ) +
130   stat_bin(
131     aes(label = after_stat(if_else (condition = count>12, as.character(count), ""))),
132     bins= as.numeric(number_ofBins),
133     position=position_stack(vjust=0.1),
134     pad=TRUE,
135     geom = "text",
136     color = "black",
137     size = 2.5,
138     y = 12.5
139   )
140
141 #print(plot)
142
143 # Dynamic variable name for the plot
144 plot_name <- paste0("plot_", i - 1) # Subtracting 1 to start numbering from 1
145 assign(plot_name, plot)
146
147 # Add the plot to the list
148 plot_list[[i - 1]] <- get(plot_name)
149 #rm(plot)
150 }
151
152 }
153
154 #print(plot_list[[1]])
155
156 #save the plotlist for later on
157 saveRDS(plot_list, "quarto_outputs/plotlist.rds")

```

21 Generate multi-panel histogram figure 4A-L

This code compiles all 12 histograms generated above into one figure.

```

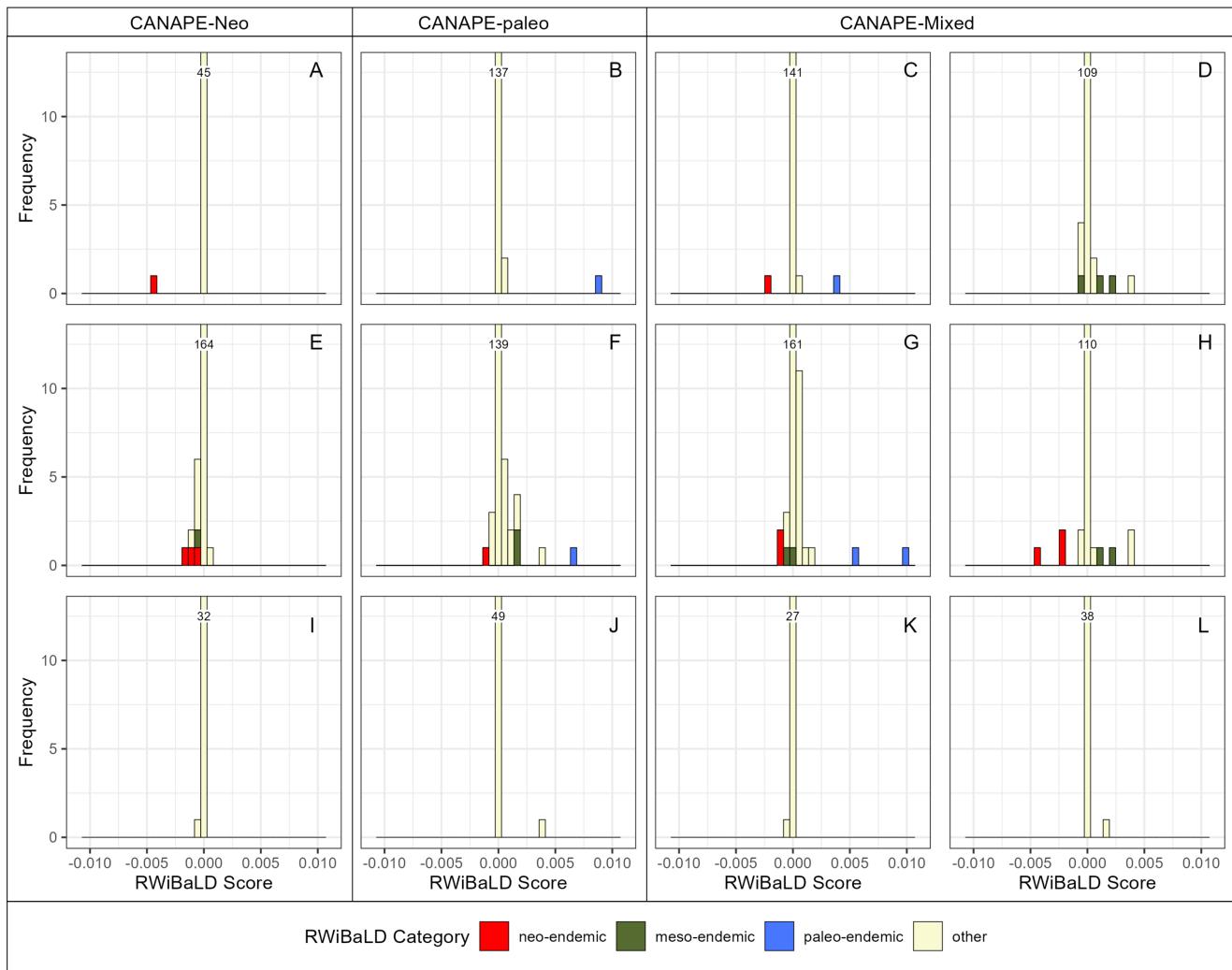
1 plot_list <- readRDS("quarto_outputs/plotlist.rds")
2
3 patchwork <- plot_list[[1]] + xlab(NULL) +
4   guides(x = "none") +
5   plot_list[[2]] + xlab(NULL) + ylab(NULL) +
6   guides(x = "none", y = "none") +
7   plot_list[[3]] + xlab(NULL) + ylab(NULL) +

```

```

8     guides(x = "none", y = "none") +
9     plot_list[[4]] + xlab(NULL) + ylab(NULL) +
10    guides(x = "none", y = "none") +
11    plot_list[[5]] + xlab(NULL) + guides(x = "none") +
12    plot_list[[6]] + xlab(NULL) + ylab(NULL) +
13    guides(x = "none", y = "none") +
14    plot_list[[7]] + xlab(NULL) + ylab(NULL) +
15    guides(x = "none", y = "none") +
16    plot_list[[8]] + xlab(NULL) + ylab(NULL) +
17    guides(x = "none", y = "none") +
18    plot_list[[9]] +
19    plot_list[[10]] + ylab(NULL) + guides(y = "none") +
20    plot_list[[11]] + ylab(NULL) + guides(y = "none") +
21    plot_list[[12]] + ylab(NULL) + guides(y = "none") +
22    plot_annotation(
23      subtitle = 'CANAPE-Neo'                                CANAPE-paleo
24      ↵  CANAPE-Mixed',
25      caption = '',
26      tag_levels = 'A') +
27      plot_layout(ncol = 4, guides = 'collect') &
28      theme(legend.position = "bottom", plot.tag.position = c(0.9, 0.9),
29             plot.tag = element_text(size = 12, hjust = 0, vjust = 0))
30
31 #print(patchwork)
32 ggsave("quarto_outputs/figures/Figure4_A.png", patchwork, width = 3000, height = 2400, units = "px")

```



22 Load results data & generate data files for biome maps

Now we take the RWiBaLD results and split the taxa in the original specimen data from the [Categorical Analysis of Neo And Paleo-Endemism \(CANAPE\)](#) paper according to these RWiBaLD categories so we can overlay them on the biome map published by Crisp et al. in 2004.

```

1 original_paper_specimen_data_file <-  

2   "doi_10_5061_dryad_dv4qk__v20150514/Point_distribution_Australian_Phyletic_Diversity_Acacia.csv"  

3  

4 original_paper_specimen_data <- read.table(original_paper_specimen_data_file, header=T, sep=",")  

5  

6 #View(original_paper_specimen_data)  

7 #View(rwibald_results_all_with_range)  

8  

9 # Summarize the data in the rwibaled_type column  

10 summary_df <- rwibald_results_all_with_range %>%  

11   group_by(rwibaled_type) %>%  

12   summarise(count = n()) # Count of each rwibaled_type  

13  

14 # Create a table using gt  

summary_table <- summary_df %>%
```

Summary of rwibaled_type

Count of each type

rwibald_type	count
meso-endemic	59
neo-endemic	55
other	880
paleo-endemic	21

```

15 gt() %>%
16   tab_header(
17     title = "Summary of rwibaled_type",
18     subtitle = "Count of each type"
19   )
20
21 # Print the table
22 summary_table
23
24 # Extract the 'NAME' values where 'rwibald_type' is 'meso-endemic'
25 meso_names <- rwibald_results_all_with_range[rwibald_results_all_with_range$rwibald_type ==
26   "meso-endemic", "NAME"]
27
28 # Filter the 'original_paper_specimen_data' dataframe
29 meso_taxa <- original_paper_specimen_data[original_paper_specimen_data$Species %in% meso_names, ]
30
31 # Write data to file
32 write.csv(meso_taxa, "quarto_outputs/Meso_endemics.csv" ,row.names=FALSE)
33
34 # Extract the 'NAME' values where 'rwibald_type' is 'neo-endemic'
35 neo_names <- rwibald_results_all_with_range[rwibald_results_all_with_range$rwibald_type == "neo-endemic",
36   "NAME"]
37
38 # Filter the 'original_paper_specimen_data' dataframe
39 neo_taxa <- original_paper_specimen_data[original_paper_specimen_data$Species %in% neo_names, ]
40
41 # Write data to file
42 write.csv(neo_taxa, "quarto_outputs/Neo_endemics.csv" ,row.names=FALSE)
43
44 # Extract the 'NAME' values where 'rwibald_type' is 'paleo-endemic'
45 paleo_names <- rwibald_results_all_with_range[rwibald_results_all_with_range$rwibald_type ==
46   "paleo-endemic", "NAME"]
47
48 # Filter the 'original_paper_specimen_data' dataframe
49 paleo_taxa <- original_paper_specimen_data[original_paper_specimen_data$Species %in% paleo_names, ]

```

23 Two tailed relative phylogenetic diversity (RPD) & CANAPE functions

This is the function for calculating Categorical Analysis of Neo- and Paleo-endemism (CANAPE) from ranked P scores of Biodiverse analysis.

```

1 #Standard 2 tailed test for RPD
2 significance_fun <- function(x){
3   if (x >= 0.99) {
4     return("Very Highly Sig")
5   } else if (x >= 0.975){
6     return ("Highly Sig")
7   } else if (x <= 0.01){
8     return ("Very Sig Low")
9   } else if (x <= 0.025){
10    return ("Sig Low")
11  } else {
12    return("Not Sig")
13  }
14}
15
16 #two pass test for RPE
17 # x=P_PE_WE_P, y=P_PHYLO_RPE_NULL2, z=P_PHYLO_RPE2
18 significance_super_fun <- function(x, y, z){
19   if (x > 0.95 || y > 0.95) {
20     if (z <= 0.025){
21       return ("Neo")
22     } else if (z >= 0.975){
23       return ("paleo")
24     } else if (x >= 0.99 || y >= 0.99){
25       return ("Super")
26     } else {
27       return("Mixed")
28     }
29   } else {
30     return("Not Sig")
31   }
32 }
```

24 Load CANAPE & randomisation results

Here we load the re-created biodiverse results from the original paper, calculate CANAPE and re-generate the CANAPE map. Note: these results may differ slightly from the original paper as we re-ran the randomisation in biodiverse.

```

1 data_dir <- "Acacia_biodiverse_exports/"
2
3 # The CANAPE results file calculated in Biodiverse directly and exported
4 biodiverse_canape_results_file <- paste0(data_dir, "Acacia_Rand1_CANAPE_Export.csv")
5
6 # The spatial & randomisation results files calculated in Biodiverse and exported
7 biodiverse_observed_data_file <- paste0(data_dir, "Acacia_SPATIAL_RESULTS_Export.csv")
8 biodiverse_rand_results_file <- paste0(data_dir, "Acacia_Rand1_SPATIAL_RESULTS_Export.csv")
9
10 biodiverse_canape_results <- read.table(biodiverse_canape_results_file, header=T,sep=",", check.names =
11   FALSE )
12
13 biodiverse_observed_spatial_results <- read.table(biodiverse_observed_data_file, header=T,sep=",")
14 biodiverse_rand_spatial_results <- read.table(biodiverse_rand_results_file, header=T,sep=",")
15
16 biodiverse_results_concatenated <- cbind(biodiverse_observed_spatial_results,
17   biodiverse_rand_spatial_results)
18
19 #View(biodiverse_results_concatenated)
```

```

18 #####
19 #Create new columns in dataframe and
20 #populate them using the functions above
21 #####
22 #####
23
24 targets <- c("PHYLO_RPD2", "PD_P", "PE_WE_P", "PD_P_per_taxon", "PHYLO_RPE2")
25
26 for (name in targets) {
27   colname <- paste0("P_", name) # prepend the P_ since we want the proportions, saves some typing above
28   new_colname = paste0(colname, "_SIG")
29   trait_index <- match (colname, colnames(biodiverse_results_concatenated))
30   # Apply the function to every row of column with index "trait_index"
31   # and generate a column in the dataframe showing significant cells
32   if (!is.na(trait_index)) {
33     biodiverse_results_concatenated[[new_colname]] <- apply
34     ↪ (biodiverse_results_concatenated[trait_index], MARGIN=c(1), significance_fun)
35   } else {
36     print (paste("Cannot find index", colname, "in data frame"))
37   }
38 }
39 biodiverse_results_concatenated$P_PHYLO_RPE2_CANAPE_SIG <- sapply(
40   1:nrow(biodiverse_results_concatenated),
41   function(x) significance_super_fun(
42     biodiverse_results_concatenated$P_PE_WE_P[x],
43     biodiverse_results_concatenated$P_PHYLO_RPE_NULL2[x],
44     biodiverse_results_concatenated$P_PHYLO_RPE2[x]
45   )
46 )
47
48 #View(biodiverse_results_concatenated)

```

25 Generate CANAPE map figure 5A

Here we generate the CANAPE map, indicating cell locations from the histograms in figure 4A-L. CANAPE identifies geographic concentrations of high PE, and gives a summary classification of the type of endemism dominating in a location. The RWiBaLD histograms (figure 4A-L) identify the specific branches that contribute the most to PE in a given cell, and what type of endemism they represent.

```

1 #loadfonts()
2
3 myFont <- choose_font(c("HelvLight", "Arial", "sans"), quiet = TRUE) #load a font if available
4
5 map_text <- "Categorical Analysis of Neo- And Paleo- Endemism"
6 sigplot <- "P_PHYLO_RPE2_CANAPE_SIG"
7 col_scheme <- c("paleo" = "royalblue1", "Not Sig" = "snow2", "Neo" = "red", "Super" = "#9D00FF", "Mixed"=
8   ↪ "#CB7FFF")
9 legend_order <- c("Neo", "paleo", "Not Sig", "Mixed", "Super")
10 legend_labels <- c("Neo"="Neo", "paleo"="Paleo", "Not Sig"="Not significant", "Mixed"="Mixed",
11   ↪ "Super"="Super")
12
13 biodiverse_results_concatenated[, sigplot] <- factor(biodiverse_results_concatenated[, sigplot],
14   ↪ levels=legend_order)
15 Axis_0 <- "Axis_0"
16 Axis_1 <- "Axis_1"
17
18 map_shape_file <- paste0("shape_files/coastline_albers.shp")

```

```

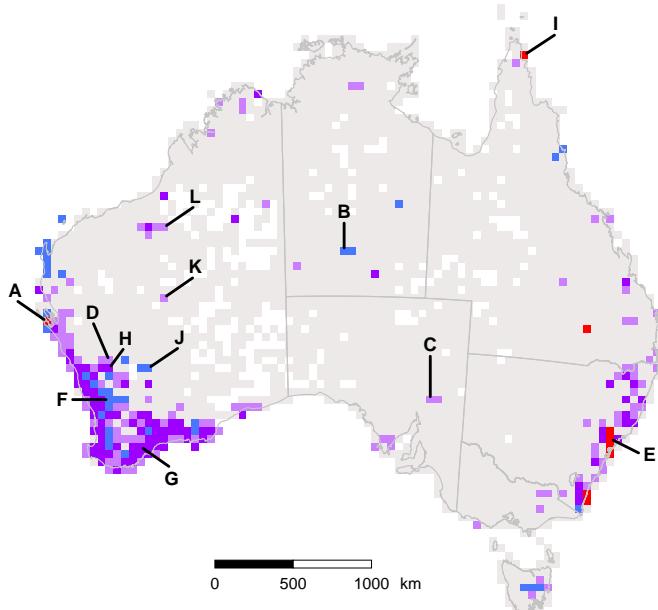
16 map_data    <- st_read(map_shape_file)
17
18 cols_to_keep <- c("-1825000:-2975000", "75000:-2525000",
19   ↵ "625000:-3475000", "-1425000:-3225000", "1775000:-3725000", "-1425000:-3475000", "-1225000:-3775000",
20   ↵ "-1425000:-3275000", "1225000:-1275000", "-1175000:-3275000", "-1075000:-2825000",
21   ↵ "-1075000:-2375000")
22
23
24 # Initialize empty vectors for x and y
25 x <- c()
26 y <- c()
27
28 # Split each element of cols_to_keep and append to x and y for plotting on the map
29 for (val in cols_to_keep) {
30   parts <- strsplit(val, ":")[[1]]
31   x <- c(x, as.numeric(parts[1]))
32   y <- c(y, as.numeric(parts[2]))
33 }
34
35 # Print the result
36 #print(x)
37 #print(y)
38
39 labels <- c("A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L")
40
41 # Create data frame
42 histograms <- data.frame(x = x, y = y, label = labels)
43
44 # Plot the sf object with ggplot2, coloring by the BIOME column
45 map_plot_CANAPE <- ggplot() +
46   geom_tile(data=biodiverse_results_concatenated, aes_string(x=Axis_0, y=Axis_1, fill=sigplot)) +
47   geom_sf(data = map_data, colour = "grey77" , fill="transparent") +
48   scale_fill_manual(values = col_scheme, labels=legend_labels, name="CANAPE", guide =
49     ↵ guide_legend(direction = "horizontal", title.position = "bottom", title.hjust=0.5, title.vjust=0.5,
50     ↵ label.position="bottom", label.hjust = 0.5, label.vjust = 0.1, lineheight=0.5))++
51   geom_text_repel(data=histograms, aes(x = x, y = y, label = labels), fontface = "bold", size = 2.5,
52     ↵ nudge_x = c(-200000, 0, 0, -100000, 250000, -300000, 200000, 100000, 200000, 200000, 200000,
53     ↵ 200000), nudge_y = c(200000, 250000, 350000, 300000, -100000, 0, -200000, 200000, 200000,
54     ↵ 200000, 200000)) +
55   annotate("rect", xmin = -750000, xmax = -250000, ymin = -4500000, ymax = -4550000, fill = "black",
56     ↵ colour = "black", alpha = 1, linewidth = 0.1) +
57   annotate("rect", xmin = -250000, xmax = 250000, ymin = -4500000, ymax = -4550000, fill = "white",
58     ↵ colour = "black", alpha = 1, linewidth = 0.1) +
59   annotate("text", label = "0", x = -750000, y = -4650000, size=rel(2), face = 'plain', family =
60     ↵ myFont) +
61   annotate("text", label = "500", x = -250000, y = -4650000, size=rel(2), face = 'plain', family =
62     ↵ myFont) +
63   annotate("text", label = "1000", x = 250000, y = -4650000, size=rel(2), face = 'plain', family =
64     ↵ myFont) +
65   annotate("text", label = "km", x = 500000, y = -4650000, size=rel(2), face = 'plain', family =
66     ↵ myFont) +
67   theme(text = element_text(family = myFont),
68     strip.background = element_blank(),
69     title = element_text(colour = 'black', angle = 0, size=rel(1), face = 'plain', family = myFont),
70     axis.line=element_blank(),axis.text.x=element_blank(),
71     axis.text.y=element_blank(),axis.ticks=element_blank(),
72     axis.title.x=element_blank(), axis.title.y=element_blank(),
73     legend.position="none",
74     legend.direction='horizontal',
75     legend.text = element_text(colour = 'black', angle = 0, size=rel(1), face = 'plain', family =
76     ↵ myFont),
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
917
918
919
919
920
921
922
923
924
925
926
927
927
928
929
929
930
931
932
933
934
935
936
937
937
938
939
939
940
941
942
943
944
945
946
946
947
948
948
949
949
950
951
952
953
954
955
956
956
957
958
958
959
959
960
961
962
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
1552
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1560
1561
1561
1562
1562
1563
1563
1564
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1570
1571
1571
1572
1572
1573
1573
1574
1574
1575
1575
1576
1576
1577
1577
1578
1578
1579
1579
1580
1580
1581
1581
1582
1582
1583
1583
1584
1584
1585
1585
1586
1586
1587
1587
1588
1588
1589
1589
1590
1590
1591
1591
1592
1592
1593
1593
1594
1594
1595
1595
1596
1596
1597
1597
1598

```

```

61 panel.grid = element_blank(),
62 panel.background=element_blank(),#element_rect(colour = "black", fill="white", size = 1),
63 panel.border = element_blank(),
64 plot.background=element_blank(),#element_rect(colour = "black", fill="white", size = 1),
65 plot.margin=unit(c(0,0,0,0),"line"))
66
67 print(map_plot_CANAPE)

```



```

1 ggsave('quarto_outputs/figures/Figure5_A.png', map_plot_CANAPE, width = 1024, height = 1024, units =
  ↵ "px", bg = "white")

```

```

Reading layer `coastline_albers` from data source
  `C:\E\gitRepos\rwibald\shape_files\coastline_albers.shp` using driver `ESRI Shapefile'
Simple feature collection with 44 features and 4 fields
Geometry type: POLYGON
Dimension:     XY
Bounding box:  xmin: -1887741 ymin: -4840771 xmax: 2121462 ymax: -1029671
Projected CRS: GDA94 / Australian Albers

```

26 Generate biome map with neo-endemics figure 5B

Here we show the specimen data for all of the neo-endemic taxa categorised by RWiBaLD overlaid on the biome map published by [Crisp et al.](#) in 2004.

```

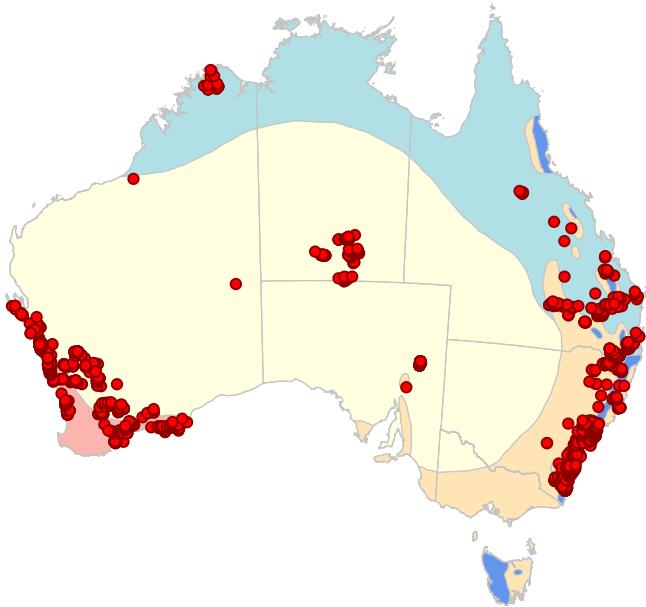
1 #loadfonts()
2
3 #myFont <- choose_font(c("HelvLight", "Arial", "sans"), quiet = TRUE) #load a font if available
4
5 col_scheme <- c("aseasonal wet" = "cornflowerblue",
6                 "eremeant" = "lightyellow1",
7                 "monsoonal tropics" = "powderblue",
8                 "southeastern temperate" = "moccasin",

```

```

9         "southwestern temperate"= "#fbb4ae")
10 legend_order <-c("aseasonal wet",
11                  "eremeans",
12                  "monsoonal tropics",
13                  "southeastern temperate",
14                  "southwestern temperate")
15 legend_labels <- c("aseasonal wet" = "aseasonal wet",
16                      "eremeans" = "eremeans",
17                      "monsoonal tropics" = "monsoonal tropics",
18                      "southeastern temperate"="southeastern temperate",
19                      "southwestern temperate"="southwestern temperate")
20
21 map_shape_file1 <- paste0("shape_files/coastline_albers.shp")
22 map_data   <- st_read(map_shape_file1)
23 #geom_sf(data = map_data, colour = "grey77" , fill="transparent") +
24
25 map_shape_file <- paste0("shape_files/biomes_crisp_3577/biomes_crisp_3577.shp")
26 # Read the shapefile as an sf object
27 biomes_sf <- st_read(map_shape_file)
28
29 csv_file <- paste0("quarto_outputs/Neo_endemics.csv")
30 # Read the CSV file into a data frame
31 csv_data <- read.csv(csv_file)
32 # Convert data frame to sf object
33 csv_sf <- st_as_sf(csv_data, coords = c("x_metres_EPSG_3577_Albers_Equal_Area",
34                                         "y_metres_EPSG_3577_Albers_Equal_Area"), crs = st_crs(biomes_sf))
35
36 # Plot the sf object with ggplot2, coloring by the BIOME column
37 map_plot_Neo <- ggplot() +
38   geom_sf(data = biomes_sf, aes(fill = BIOME), colour = "grey77") +
39   geom_sf(data = map_data, colour = "grey77" , fill="transparent") +
40   geom_sf(data = csv_sf, fill= "red", color = "red4", shape = 21) +
41   scale_fill_manual(values = col_scheme, labels=legend_labels, name="Biomes", guide =
42     guide_legend(direction = "horizontal", title.position = "bottom", title.hjust=0.5, title.vjust=0.5,
43     label.position="bottom", label.hjust = 0.5, label.vjust = 0.1, lineheight=0.5))++
44   theme(text = element_text(family = myFont),
45         strip.background = element_blank(),
46         title = element_text(colour = 'black', angle = 0, size=rel(1), face = 'plain', family = myFont),
47         axis.line=element_blank(),axis.text.x=element_blank(),
48         axis.text.y=element_blank(),axis.ticks=element_blank(),
49         axis.title.x=element_blank(), axis.title.y=element_blank(),
50         legend.position="none",
51         legend.direction='horizontal',
52         legend.text = element_text(colour = 'black', angle = 0, size=rel(1), face = 'plain', family =
53           myFont),
54         legend.key = element_rect(color = "black", size = 0),
55         panel.grid = element_blank(),
56         panel.background=element_blank(),#element_rect(colour = "black", fill="white", size = 1),
57         panel.border = element_blank(),
58         plot.background=element_blank(),#element_rect(colour = "black", fill="white", size = 1),
59         plot.margin=unit(c(0,0,0,0),"line"))
60
61
62 print(map_plot_Neo)

```



```
1 ggsave("quarto_outputs/figures/Figure5_B.png", map_plot_Neo, width = 3000, height = 2600, units = "px")
```

```
Reading layer `coastline_albers' from data source
`C:\E\gitRepos\rwibald\shape_files\coastline_albers.shp' using driver `ESRI Shapefile'
Simple feature collection with 44 features and 4 fields
Geometry type: POLYGON
Dimension:     XY
Bounding box:  xmin: -1887741 ymin: -4840771 xmax: 2121462 ymax: -1029671
Projected CRS: GDA94 / Australian Albers
Reading layer `biomes_crisp_3577' from data source
`C:\E\gitRepos\rwibald\shape_files\biomes_crisp_3577\biomes_crisp_3577.shp'
using driver `ESRI Shapefile'
Simple feature collection with 5 features and 3 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: -1887414 ymin: -4840588 xmax: 2121442 ymax: -1087816
Projected CRS: GDA94 / Australian Albers
```

27 Generate biome map with meso-endemics figure 5C

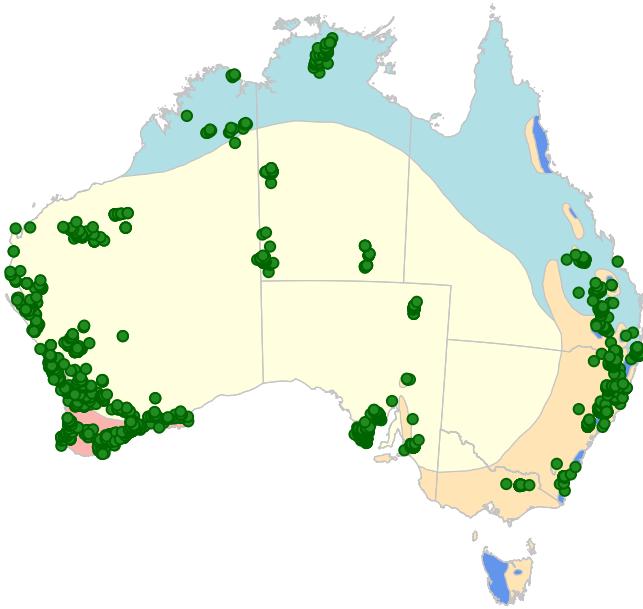
Here we show the specimen data for all of the meso-endemic taxa categorised by RWiBaLD overlaid on the biome map published by [Crisp et al.](#) in 2004.

```
1 #loadfonts()
2
3 #myFont <- choose_font(c("HelvLight", "Arial", "sans"), quiet = TRUE) #load a font if available
4
5 map_text <- "Categorical Analysis of Neo- And Paleo- Endemism"
6
7 col_scheme <- c("aseasonal wet" = "cornflowerblue",
8                 "eremeann" = "lightyellow1",
9                 "monsoonal tropics" = "powderblue",
10                "southeastern temperate" = "moccasin",
```

```

11         "southwestern temperate"= "#fbb4ae")
12 legend_order <-c("aseasonal wet",
13                  "eremean",
14                  "monsoonal tropics",
15                  "southeastern temperate",
16                  "southwestern temperate")
17 legend_labels <- c("aseasonal wet" = "aseasonal wet",
18                      "eremean" = "eremean",
19                      "monsoonal tropics" = "monsoonal tropics",
20                      "southeastern temperate"="southeastern temperate",
21                      "southwestern temperate"="southwestern temperate")
22
23 map_shape_file1 <- paste0("shape_files/coastline_albers.shp")
24 map_data   <- st_read(map_shape_file1)
25 #geom_sf(data = map_data, colour = "grey77" , fill="transparent") +
26
27 map_shape_file <- paste0("shape_files/biomes_crisp_3577/biomes_crisp_3577.shp")
28 # Read the shapefile as an sf object
29 biomes_sf <- st_read(map_shape_file)
30
31 csv_file <- paste0("quarto_outputs/Meso_endemics.csv")
32 # Read the CSV file into a data frame
33 csv_data <- read.csv(csv_file)
34 # Convert data frame to sf object
35 csv_sf <- st_as_sf(csv_data, coords = c("x_metres_EPSG_3577_Albers_Equal_Area",
36                                         "y_metres_EPSG_3577_Albers_Equal_Area"), crs = st_crs(biomes_sf))
37
38 # Plot the sf object with ggplot2, coloring by the BIOME column
39 map_plot_Meso <- ggplot() +
40   geom_sf(data = biomes_sf, aes(fill = BIOME), color = "grey77") +
41   geom_sf(data = map_data, colour = "grey77" , fill="transparent") +
42   geom_sf(data = csv_sf, fill= "forestgreen", color = "darkgreen", shape = 21) +
43   scale_fill_manual(values = col_scheme, labels=legend_labels, name="Biomes", guide =
44     guide_legend(direction = "horizontal", title.position = "bottom", title.hjust=0.5, title.vjust=0.5,
45     label.position="bottom", label.hjust = 0.5, label.vjust = 0.1, lineheight=0.5))++
46   theme(text = element_text(family = myFont),
47         strip.background = element_blank(),
48         title = element_text(colour = 'black', angle = 0, size=rel(1), face = 'plain', family = myFont),
49         axis.line=element_blank(),axis.text.x=element_blank(),
50         axis.text.y=element_blank(),axis.ticks=element_blank(),
51         axis.title.x=element_blank(), axis.title.y=element_blank(),
52         legend.position="none",
53         legend.direction='horizontal',
54         legend.text = element_text(colour = 'black', angle = 0, size=rel(1), face = 'plain', family =
55           myFont),
56         legend.key = element_rect(color = "black", size = 0),
57         panel.grid = element_blank(),
58         panel.background=element_blank(),#element_rect(colour = "black", fill="white", size = 1),
59         panel.border = element_blank(),
60         plot.background=element_blank(),#element_rect(colour = "black", fill="white", size = 1),
61         plot.margin=unit(c(0,0,0,0),"line"))
62
63 print(map_plot_Meso)

```



```
1 ggsave("quarto_outputs/figures/Figure5_C.png", map_plot_Meso, width = 3000, height = 2600, units = "px")
```

```
Reading layer `coastline_albers' from data source
`C:\E\gitRepos\rwibald\shape_files\coastline_albers.shp' using driver `ESRI Shapefile'
Simple feature collection with 44 features and 4 fields
Geometry type: POLYGON
Dimension:     XY
Bounding box:  xmin: -1887741 ymin: -4840771 xmax: 2121462 ymax: -1029671
Projected CRS: GDA94 / Australian Albers
Reading layer `biomes_crisp_3577' from data source
`C:\E\gitRepos\rwibald\shape_files\biomes_crisp_3577\biomes_crisp_3577.shp'
using driver `ESRI Shapefile'
Simple feature collection with 5 features and 3 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: -1887414 ymin: -4840588 xmax: 2121442 ymax: -1087816
Projected CRS: GDA94 / Australian Albers
```

28 Generate biome map with paleo-endemics figure 5D

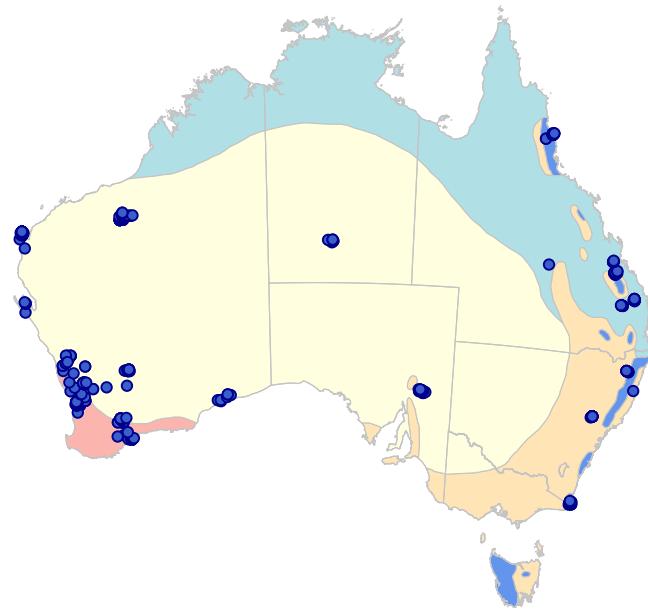
Here we show the specimen data for all of the paleo-endemic taxa categorised by RWiBaLD overlaid on the biome map published by [Crisp et al.](#) in 2004.

```
1 #loadfonts()
2
3 #myFont <- choose_font(c("HelvLight", "Arial", "sans"), quiet = TRUE) #load a font if available
4
5 map_text <- "Categorical Analysis of Neo- And Paleo- Endemism"
6
7 col_scheme <- c("aseasonal wet" = "cornflowerblue",
8                 "eremeann" = "lightyellow1",
9                 "monsoonal tropics" = "powderblue",
10                "southeastern temperate" = "moccasin",
```

```

11         "southwestern temperate"= "#fbb4ae")
12 legend_order <-c("aseasonal wet",
13                  "eremean",
14                  "monsoonal tropics",
15                  "southeastern temperate",
16                  "southwestern temperate")
17 legend_labels <- c("aseasonal wet" = "aseasonal wet",
18                      "eremean" = "eremean",
19                      "monsoonal tropics" = "monsoonal tropics",
20                      "southeastern temperate"="southeastern temperate",
21                      "southwestern temperate"="southwestern temperate")
22
23 map_shape_file1 <- paste0("shape_files/coastline_albers.shp")
24 map_data  <- st_read(map_shape_file1)
25 #geom_sf(data = map_data, colour = "grey77" , fill="transparent") +
26
27 map_shape_file <- paste0("shape_files/biomes_crisp_3577/biomes_crisp_3577.shp")
28
29 # Read the shapefile as an sf object
30 biomes_sf <- st_read(map_shape_file)
31
32 csv_file <- paste0("quarto_outputs/Paleo_endemics.csv")
33
34 # Read the CSV file into a data frame
35 csv_data <- read.csv(csv_file)
36
37 # Convert data frame to sf object
38 csv_sf <- st_as_sf(csv_data, coords = c("x_metres_EPSG_3577_Albers_Equal_Area",
39                                         "y_metres_EPSG_3577_Albers_Equal_Area"), crs = st_crs(biomes_sf))
40
41 # Plot the sf object with ggplot2, coloring by the BIOME column
42 map_plot_Paleo  <- ggplot() +
43   geom_sf(data = biomes_sf, aes(fill = BIOME), color = "grey77") +
44   geom_sf(data = map_data, colour = "grey77" , fill="transparent") +
45   geom_sf(data = csv_sf, fill= "royalblue3", color = "darkblue", shape = 21) +
46   scale_fill_manual(values = col_scheme, labels=legend_labels, name="Biomes", guide =
47     guide_legend(direction = "horizontal", title.position = "bottom", title.hjust=0.5, title.vjust=0.5,
48     label.position="bottom", label.hjust = 0.5, label.vjust = 0.1, lineheight=0.5))++
49   theme(text = element_text(family = myFont),
50         strip.background = element_blank(),
51         title = element_text(colour = 'black', angle = 0, size=rel(1), face = 'plain', family = myFont),
52         axis.line=element_blank(),axis.text.x=element_blank(),
53         axis.text.y=element_blank(),axis.ticks=element_blank(),
54         axis.title.x=element_blank(), axis.title.y=element_blank(),
55         legend.position="none",
56         legend.direction='horizontal',
57         legend.text = element_text(colour = 'black', angle = 0, size=rel(0.5), face = 'plain', family =
58           myFont),
59         legend.key = element_rect(color = "black", size = 0),
60         panel.grid = element_blank(),
61         panel.background=element_blank(),#element_rect(colour = "black", fill="white", size = 1),
62         panel.border = element_blank(),
63         plot.background=element_blank(),#element_rect(colour = "black", fill="white", size = 1),
64         plot.margin=unit(c(0,0,0,0),"line"))
65
66
67 print(map_plot_Paleo)

```



```
1 ggsave("quarto_outputs/figures/Figure5_D.png", map_plot_Paleo, width = 3000, height = 2600, units = "px")
```

Reading layer `coastline_albers` from data source
`C:\E\gitRepos\rwibald\shape_files\coastline_albers.shp` using driver `ESRI Shapefile'
Simple feature collection with 44 features and 4 fields
Geometry type: POLYGON
Dimension: XY
Bounding box: xmin: -1887741 ymin: -4840771 xmax: 2121462 ymax: -1029671
Projected CRS: GDA94 / Australian Albers
Reading layer `biomes_crisp_3577` from data source
`C:\E\gitRepos\rwibald\shape_files\biomes_crisp_3577\biomes_crisp_3577.shp`
using driver `ESRI Shapefile'
Simple feature collection with 5 features and 3 fields
Geometry type: MULTIPOLYGON
Dimension: XY
Bounding box: xmin: -1887414 ymin: -4840588 xmax: 2121442 ymax: -1087816
Projected CRS: GDA94 / Australian Albers

29 Generate 4 up map figure 5ABCD

Here we compile the maps into one figure 5A-D

```
1 patchwork_map <- map_plot_CANAPE +  

2     map_plot_Neo +  

3     map_plot_Meso +  

4     map_plot_Paleo  

5  

6 patchwork_map <- patchwork_map +  

7     plot_annotation(subtitle = '',
8                     caption = '',
9                     tag_levels = 'A') +  

10    plot_layout(ncol = 2, guides = 'collect') &  

11    theme(legend.position = "bottom",
```

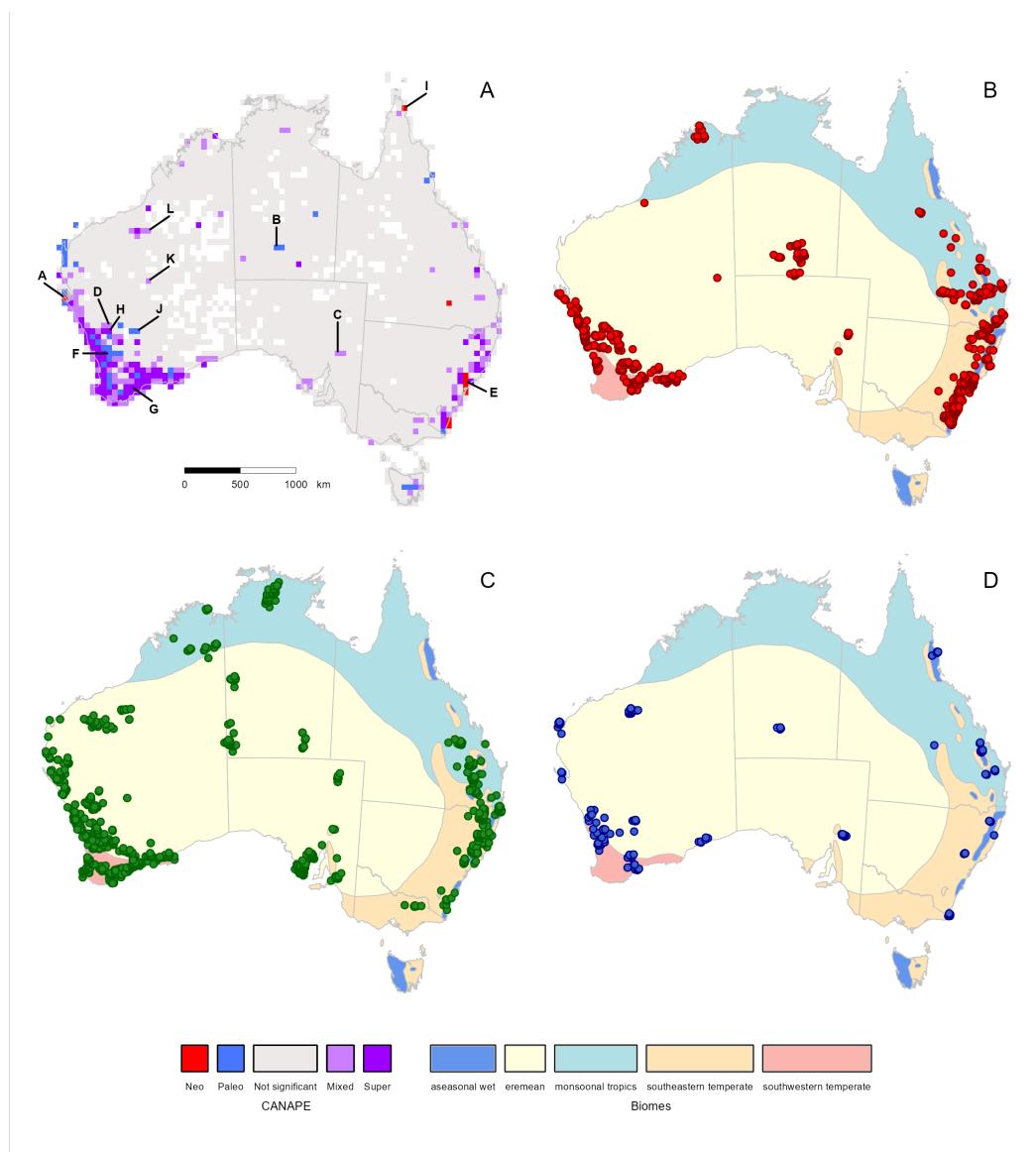
```

12
13
14
15
16
17
18
19
20
21
22
23
24
  legend.title=element_text(size=rel(0.6)),
  legend.text=element_text(size=rel(0.5)),
  legend.title.position = "bottom",
  legend.label.position= "bottom",
  legend.key = element_rect(color = "black", size = 0.5),
  legend.key.height = unit(0.5, "cm"),
  legend.key.width = unit(0.5, "cm"),
  plot.tag.position = c(0.9, 0.9),
  plot.tag = element_text(size = 12, hjust = 0, vjust = 0))

# print(patchwork_map)

ggsave('quarto_outputs/figures/Figure5_ABCD.png', patchwork_map, width = 3000, height = 2600, units =
  "px")

```



30 Table of RWiBaLD results supplementary table 1

RWiBaLD results for each branch on the phylogeny of Australian Acacia.

```
1 RWiBaLD_results_csv <- paste0("quarto_outputs/Acacia_RWiBaLD_results_all_with_range.csv")
2
3 rwibald_results_all_with_range <- read.table(RWiBaLD_results_csv, header=T,sep=",")
4
5 #View(rwibald_results_all_with_range)
6
7 #colnames(rwibald_results_all_with_range)
8
9 gt_table <- rwibald_results_all_with_range %>%
10   select("NAME", "branch_length_comparison_tree", "branch_length_observed_tree",
11         "rwibald_score", "rwibald_type", "range_cell_count")
12
13 # print table using gt
14 rwibald_results_all_with_range_table <- gt(gt_table) %>%
15   tab_options(
16     table.width = pct(100),
17     table.layout = "auto",
18     table.align = "left",
19     table.margin.left = px(5),
20     table.margin.right = px(5),
21     table.font.size = px(8),
22     column_labels.font.size = px(10),
23     heading.align = "center",
24     heading.title.font.size = px(12),
25     quarto.use_bootstrap = TRUE
26   ) %>%
27   opt_row_striping() %>%
28   tab_header(
29     title = md("RWiBaLD Results")
30   ) %>%
31   tab_style(
32     style = cell_text(align = "center"),
33     locations = cells_body()
34   )
35
36 rwibald_results_all_with_range_table
```

RWiBaLD Results

NAME	branch_length_comparison_tree	branch_length_observed_tree	rwibald_score	rwibald_type	range_cell_count
1	8.777963e-05	2.234812e-04	1.357015e-04	other	103
10	8.951784e-05	3.583689e-05	-5.368095e-05	other	101
100	4.109683e-05	7.239508e-06	-3.385732e-05	other	220
101	2.366833e-05	3.361011e-05	9.941780e-06	other	382
102	2.318283e-05	1.884262e-05	-4.340207e-06	other	390
103	2.137424e-05	4.697007e-05	2.559583e-05	other	423
104	2.318283e-04	9.137027e-04	6.818744e-04	other	39
105	6.027535e-04	1.624921e-03	1.022168e-03	other	15
106	1.532424e-04	1.057493e-04	-4.749313e-05	other	59
107	9.935497e-05	3.545809e-04	2.552259e-04	other	91
108	9.827502e-05	3.429999e-05	-6.397503e-05	other	92
109	8.449815e-05	1.130098e-04	2.851166e-05	other	107
11	8.777963e-05	2.896118e-05	-5.881846e-05	other	103
110	1.891486e-05	9.029480e-06	-9.885378e-06	other	478
111	7.534419e-04	1.343083e-03	5.896411e-04	other	12
112	1.614518e-04	4.342113e-05	-1.180307e-04	other	56
113	1.051314e-04	1.208033e-04	1.567189e-05	other	86
114	8.777963e-05	6.943275e-09	-8.777269e-05	other	103
115	7.862002e-05	9.885975e-06	-6.873404e-05	other	115
116	7.175637e-05	1.346953e-05	-5.828684e-05	other	126
117	7.008761e-05	4.083340e-05	-2.925421e-05	other	129
118	7.008761e-05	5.616836e-05	-1.391925e-05	other	129
119	6.954848e-05	1.778353e-05	-5.176495e-05	other	130
12	1.335495e-05	1.097784e-05	-2.377112e-06	other	677
120	1.597403e-05	7.837726e-06	-8.136307e-06	other	566
121	1.504376e-05	1.467145e-05	-3.723155e-07	other	601
122	9.881205e-06	4.100078e-06	-5.781126e-06	other	915
123	9.859654e-06	1.414163e-06	-8.445490e-06	other	917
124	3.117690e-04	3.328489e-04	2.107982e-05	other	29
125	2.511473e-04	3.887985e-04	1.376512e-04	other	36
126	2.054841e-04	4.573766e-05	-1.597465e-04	other	44
127	2.825407e-04	6.909660e-05	-2.134441e-04	other	32
128	2.825407e-04	2.411957e-04	-4.134496e-05	other	32
129	5.022946e-04	2.914361e-04	-2.108585e-04	other	18
13	1.310334e-05	5.646846e-06	-7.456490e-06	other	690
130	1.130163e-04	3.901460e-05	-7.400168e-05	other	80
131	1.004589e-04	4.157142e-05	-5.888750e-05	other	90
132	7.862002e-05	1.278658e-04	4.924578e-05	other	115
133	7.662121e-05	3.954025e-05	-3.708095e-05	other	118
134	3.863804e-05	1.794236e-06	-3.684381e-05	other	234
135	4.346780e-05	4.264787e-05	-8.199310e-07	other	208
136	6.551668e-05	4.443585e-05	-2.108084e-05	other	138
137	2.807858e-05	5.238707e-05	2.430849e-05	other	322
138	2.539692e-05	7.418464e-05	4.878772e-05	other	356
139	4.758580e-04	6.521900e-05	-4.106390e-04	other	19
14	1.145919e-05	4.368943e-06	-7.090249e-06	other	789
140	3.767209e-04	2.539890e-04	-1.227320e-04	other	24
141	3.477424e-04	5.308590e-04	1.831166e-04	other	26
142	2.916549e-04	4.352484e-04	1.435935e-04	other	31
143	2.054841e-04	2.656276e-05	-1.789214e-04	other	44
144	1.506884e-04	1.191929e-08	-1.506765e-04	other	60
145	1.238535e-04	5.947281e-05	-6.438065e-05	other	73
146	1.221798e-04	3.874051e-05	-8.343925e-05	other	74
147	8.145317e-05	9.120355e-05	9.750379e-06	other	111
148	1.936039e-05	8.736721e-06	-1.062367e-05	other	467
149	1.361642e-05	7.233721e-06	-6.382699e-06	other	664
15	1.090628e-05	2.498268e-05	1.407641e-05	other	829
150	1.302781e-05	3.732148e-06	-9.295665e-06	other	694
151	3.931001e-04	1.185664e-04	-2.745337e-04	other	23
152	2.318283e-04	6.839497e-05	-1.634333e-04	other	39
153	1.586193e-04	1.557460e-04	-2.873300e-06	other	57
154	7.008761e-05	6.309607e-06	-6.377801e-05	other	129
155	6.551668e-05	1.174379e-05	-5.377289e-05	other	138
156	6.504534e-05	4.422731e-05	-2.081803e-05	other	139
157	3.631045e-05	1.614817e-05	-2.016228e-05	other	249
158	5.758791e-05	5.237361e-05	-5.214300e-06	other	157
159	4.589494e-05	1.856155e-05	-2.733338e-05	other	197
16	6.901757e-05	9.388402e-05	2.486645e-05	other	131
160	4.000576e-05	2.116269e-05	-1.884307e-05	other	226
161	2.620667e-05	2.072920e-09	-2.620460e-05	other	345
162	1.369894e-04	5.770269e-05	-7.928674e-05	other	66
163	1.027421e-04	1.694995e-05	-8.579212e-05	other	88
164	8.001152e-05	3.946034e-05	-4.055118e-05	other	113
165	5.318413e-05	4.206808e-09	-5.317992e-05	other	170
166	4.224908e-05	3.435192e-06	-3.881388e-05	other	214
167	9.132629e-05	2.496945e-05	-6.635684e-05	other	99
168	2.825407e-04	4.848574e-05	-2.340550e-04	other	32
169	1.674315e-04	2.781283e-05	-1.396187e-04	other	54
170	6.192673e-05	8.906534e-05	2.713862e-05	other	146
171	1.205507e-04	3.798806e-05	-8.256264e-05	other	75
172	1.205507e-04	5.692448e-05	-6.362621e-05	other	75
173	1.174195e-04	1.042297e-04	-1.632098e-05	other	75
174	1.063683e-04	4.595375e-05	-7.146576e-05	other	77
175	8.777963e-05	1.635680e-06	-1.047326e-04	other	85
176	5.079383e-05	6.665401e-06	-8.111423e-05	other	103
177	4.967749e-05	4.017738e-09	-5.078982e-05	other	178
178	3.631045e-05	1.645418e-05	-3.322331e-05	other	182
179	2.816605e-05	2.872118e-09	-3.630758e-05	other	249
180	2.379290e-04	1.232424e-05	-1.584181e-05	other	321
181	2.226922e-05	2.067711e-06	-2.102229e-04	other	38
182	1.583415e-05	9.804553e-06	-6.029602e-06	other	406
183	1.382462e-05	2.249130e-06	-1.157549e-05	other	571
184	5.795707e-05	6.484471e-05	6.899068e-06	other	654
185	5.413953e-05	3.121126e-05	-2.292827e-05	other	156
186	8.371576e-05	6.061842e-05	-2.309734e-05	other	167
187	8.072591e-05	6.385333e-09	-8.071953e-05	other	108
188	7.930967e-05	2.326218e-05	-5.604749e-05	other	112
189	7.265293e-05	2.070162e-05	-2.023142e-05	other	114