# === Data Extraction from Figures via ShinyDigitise ===

# 1. Install required packages (run once)

if (!requireNamespace("devtools", quietly = TRUE)) {

install.packages("devtools")

}

if (!requireNamespace("promises", quietly = TRUE)) {

install.packages("promises")

}

# Install the ShinyDigitise package from GitHub (force reinstall if already present)

devtools::install\_github("EIvimeyCook/ShinyDigitise", force = TRUE)

# 2. Load libraries

library(ShinyDigitise)

library(promises)

# 3. Launch interactive digitising app and capture extracted data

# Point your file browser at the folder containing your figure images.

# Returns a data.frame of (x, y) coordinates and any grouping metadata.

figure\_dir <- "figure"

extracted\_df <- shinyDigitise(figure\_dir)

# 4. Inspect the extracted data

head(extracted\_df)

# 5. (Optional) Save extracted coordinates to disk for downstream analysis

output\_path <- file.path(figure\_dir, "digitised\_coordinates.csv")

write.csv(extracted\_df, output\_path, row.names = FALSE)

message("Digitised data saved to: ", output\_path)

# ===============================

# Alpha Diversity Meta‐Analysis

# ===============================

# 1) Install & load packages (run once)

if (!require("metafor")) install.packages("metafor")

if (!require("readxl")) install.packages("readxl")

if (!require("dplyr")) install.packages("dplyr")

if (!require("ggplot2")) install.packages("ggplot2")

library(metafor)

library(readxl)

library(dplyr)

library(ggplot2)

# 2) File paths

alpha\_fn <- "AlphaDiversity\_MetaAnalysis\_Input.xlsx"

output\_dir <- "results"

# Create results directory if it doesn't exist

if (!dir.exists(output\_dir)) dir.create(output\_dir, recursive = TRUE)

# 3) Read & prepare data

dat <- read\_excel(alpha\_fn) %>%

mutate(across(c(m1i, sd1i, n1i, m2i, sd2i, n2i), as.numeric))

# 4) Compute effect sizes (Hedges' g)

es <- escalc(

measure = "SMDH",

m1i = m1i, sd1i = sd1i, n1i = n1i,

m2i = m2i, sd2i = sd2i, n2i = n2i,

data = dat

)

# 5) Save the dataset with effect sizes & variances

write.csv(es,

file = file.path(output\_dir, "AlphaDiversity\_effect\_sizes.csv"),

row.names = FALSE)

# 6) Fit random-effects meta-analysis

mod <- rma.uni(yi, vi, data = es, method = "REML")

print(summary(mod))

# 7) Forest plot → PDF

pdf(file.path(output\_dir, "AlphaDiversity\_forest.pdf"),

width = 7, height = 5)

forest(mod,

slab = paste(es$Study\_ID),

xlab = "Hedges' g",

main = "Forest Plot: Alpha Diversity")

dev.off()

# 8) Funnel plot → PDF + Egger’s test

pdf(file.path(output\_dir, "AlphaDiversity\_funnel.pdf"),

width = 6, height = 5)

funnel(mod,

xlab = "Hedges' g",

main = "Funnel Plot: Alpha Diversity")

dev.off()

egger <- regtest(mod, model = "lm")

cat("\nEgger’s test for funnel asymmetry:\n")

print(egger)

# 9) Leave‐one‐out sensitivity analysis

loo\_df <- tibble(

Study\_Excluded = character(),

Estimate = numeric(),

CI\_lb = numeric(),

CI\_ub = numeric()

)

for (stud in unique(es$Study\_ID)) {

es\_sub <- filter(es, Study\_ID != stud)

sub\_mod <- rma.uni(yi, vi, data = es\_sub, method = "REML")

loo\_df <- loo\_df %>% add\_row(

Study\_Excluded = stud,

Estimate = sub\_mod$b,

CI\_lb = sub\_mod$ci.lb,

CI\_ub = sub\_mod$ci.ub

)

}

# Save LOO results

write.csv(loo\_df,

file = file.path(output\_dir, "AlphaDiversity\_results.csv"),

row.names = FALSE)

# 10) LOO plot with overall estimate → PDF

overall\_est <- as.numeric(mod$b)

p\_loo <- ggplot(loo\_df, aes(x = reorder(Study\_Excluded, Estimate), y = Estimate)) +

geom\_errorbar(aes(ymin = CI\_lb, ymax = CI\_ub), width = 0.2) +

geom\_point(size = 2) +

geom\_hline(yintercept = overall\_est,

linetype = "dashed",

color = "red") +

coord\_flip() +

theme\_minimal(base\_size = 12) +

labs(

title = "LOO Sensitivity:Alpha Diversity",

x = "Dropped Study",

y = "Hedges' g"

)

ggsave(filename = file.path(output\_dir, "AlphaDiversity\_LOO\_plot.pdf"),

plot = p\_loo,

width = 7, height = 5)

# 11) Orchard-style plot → PDF

# ==========================================

# Alpha Diversity Meta-Analysis Orchard Plot

# ==========================================

# 1) Install & load dependencies

if (!requireNamespace("devtools", quietly = TRUE)) install.packages("devtools")

if (!requireNamespace("readxl", quietly = TRUE)) install.packages("readxl")

if (!requireNamespace("dplyr", quietly = TRUE)) install.packages("dplyr")

if (!requireNamespace("metafor", quietly = TRUE)) install.packages("metafor")

# Install the latest version of orchaRd

if (!requireNamespace("orchaRd", quietly = TRUE)) {

devtools::install\_github("daniel1noble/orchaRd")

}

# Load libraries

library(readxl)

library(dplyr)

library(metafor)

library(orchaRd)

library(ggplot2)

# 2) File paths

input\_xlsx <- "AlphaDiversity\_MetaAnalysis\_Input.xlsx"

#output\_pdf <- "C:/Users/muhta/OneDrive/Desktop/Meta analysis/paper/results/AlphaDiversity\_OrchardPlot.pdf"

# 3) Read and prepare data

dat <- read\_excel(input\_xlsx) %>%

mutate(across(c(m1i, sd1i, n1i, m2i, sd2i, n2i), as.numeric))

# 4) Compute Hedges' g (yi) and variance (vi)

es <- escalc(

measure = "SMDH",

m1i = m1i, sd1i = sd1i, n1i = n1i,

m2i = m2i, sd2i = sd2i, n2i = n2i,

data = dat

)

es$Study\_ID <- dat$Study\_ID # keep Study\_ID for grouping

es$N <- es$n1i + es$n2i # calculate total sample size for each study

# 5) Fit random-effects meta-analysis

mod <- rma(yi, vi, data = es, method = "REML")

# 6) Generate & save the orchard plot

pdf(output\_pdf, width = 7, height = 5)

# ==========================================

# Alpha Diversity Meta-Analysis Orchard Plot

# ==========================================

# [Previous code for loading packages and preparing data remains the same until the plotting section]

# 6) Generate & save the orchard plot with skyblue points

pdf(output\_pdf, width = 7, height = 5)

# First create the basic plot and store it

orchard\_plot(

object = mod,

mod = "1", # intercept only model

group = "Study\_ID", # grouping variable

xlab = "Hedges' g (Alpha Diversity)",

# transparency of points

angle = 90,

g= FALSE,

# angle of y-axis labels

# size of prediction intervals

transfm = "none") # no transformation

dev.off()

# Step 0: Install necessary packages

install.packages(c("readxl", "metafor", "dplyr", "ggplot2", "writexl", "devtools"))

devtools::install\_github("daniel1noble/orchard")

# Load libraries

library(readxl)

library(dplyr)

library(metafor)

library(ggplot2)

library(writexl)

library(orchaRd)

# Step 1: Read Excel file

file\_path <- "relativeabundance.xlsx"

df <- read\_excel(file\_path)

# Step 2: Clean/rename columns if needed

df <- df %>%

rename(

Control\_Firmicutes = Control\_Firmicutes,

Control\_Bacteroidetes = Control\_Bacteroidetes,

Treatment\_Firmicutes = Treatment\_Firmicutes,

Treatment\_Bacteroidetes = treatement\_Bacteroidetes,

Control\_Sample = Control\_Sample,

Treatment\_Sample = Treatment\_Sample

)

# Step 3: Proportions & F:B ratio

df <- df %>%

mutate(

p\_firm\_control = Control\_Firmicutes / (Control\_Firmicutes + Control\_Bacteroidetes),

p\_firm\_treat = Treatment\_Firmicutes / (Treatment\_Firmicutes + Treatment\_Bacteroidetes),

p\_bact\_control = Control\_Bacteroidetes / (Control\_Firmicutes + Control\_Bacteroidetes),

p\_bact\_treat = Treatment\_Bacteroidetes / (Treatment\_Firmicutes + Treatment\_Bacteroidetes),

fb\_control = Control\_Firmicutes / Control\_Bacteroidetes,

fb\_treat = Treatment\_Firmicutes / Treatment\_Bacteroidetes

)

# Step 4: Log response ratios

df <- df %>%

mutate(

yi\_firm = log(p\_firm\_treat / p\_firm\_control),

yi\_bact = log(p\_bact\_treat / p\_bact\_control),

yi\_fb = log(fb\_treat / fb\_control)

)

# Step 5: Delta-method SEs

df <- df %>%

mutate(

se\_firm = sqrt(((1 - p\_firm\_treat) / (Treatment\_Sample \* p\_firm\_treat))^2 +

((1 - p\_firm\_control) / (Control\_Sample \* p\_firm\_control))^2),

se\_bact = sqrt(((1 - p\_bact\_treat) / (Treatment\_Sample \* p\_bact\_treat))^2 +

((1 - p\_bact\_control) / (Control\_Sample \* p\_bact\_control))^2),

se\_fb = sqrt(

((1 / Treatment\_Bacteroidetes)^2 + (1 / Treatment\_Firmicutes)^2) / Treatment\_Sample +

((1 / Control\_Bacteroidetes)^2 + (1 / Control\_Firmicutes)^2) / Control\_Sample

)

)

# Step 6: Meta-analysis

res\_firm <- rma(yi = yi\_firm, sei = se\_firm, data = df, method = "REML")

res\_bact <- rma(yi = yi\_bact, sei = se\_bact, data = df, method = "REML")

res\_fb <- rma(yi = yi\_fb, sei = se\_fb, data = df, method = "REML")

# Step 7: Egger's test

egger\_firm <- regtest(res\_firm, model = "rma")

egger\_bact <- regtest(res\_bact, model = "rma")

egger\_fb <- regtest(res\_fb, model = "rma")

print(egger\_firm)

print(egger\_bact)

print(egger\_fb)

# Step 8: Trim-and-fill

tf\_firm <- trimfill(res\_firm)

tf\_bact <- trimfill(res\_bact)

tf\_fb <- trimfill(res\_fb)

print(tf\_firm)

print(tf\_bact)

print(tf\_fb)

# Step 9: Funnel plots

pdf("Firmicutes\_Funnel.pdf")

funnel(res\_firm, main = "Funnel Plot: Firmicutes")

dev.off()

pdf("Bacteroidetes\_Funnel.pdf")

funnel(res\_bact, main = "Funnel Plot: Bacteroidetes")

dev.off()

pdf("FB\_Ratio\_Funnel.pdf")

funnel(res\_fb, main = "Funnel Plot: F:B Ratio")

dev.off()

#Step 10: Orchard plot

# Firmicutes Orchard Plot

pdf("Firmicutes\_Orchard.pdf")

orchard\_plot(res\_firm, mod = "1", group = "Study\_ID",

angle = 90, xlab = "Log Ratio (Firmicutes)", transfm = "none")

dev.off()

# Bacteroidetes Orchard Plot

pdf("Bacteroidetes\_Orchard.pdf")

orchard\_plot(res\_bact, mod = "1", group = "Study\_ID",

angle = 90, xlab = "Log Ratio (Bacteroidetes)", transfm = "none")

dev.off()

# F:B Ratio Orchard Plot

pdf("FB\_Ratio\_Orchard.pdf")

orchard\_plot(res\_fb, mod = "1", group = "Study\_ID",

angle = 90, xlab = "Log Ratio (Firmicutes:Bacteroidetes)", transfm = "none")

dev.off()

# Step 11: Forest plots

forest(res\_firm, slab = paste(df$Study\_ID, df$Treatment), xlab = "Log Ratio (Firmicutes)", main = "Forest Plot: Firmicutes")

forest(res\_bact, slab = paste(df$Study\_ID, df$Treatment), xlab = "Log Ratio (Bacteroidetes)", main = "Forest Plot:Bacteroidetes")

forest(res\_fb, slab = paste(df$Study\_ID, df$Treatment), xlab = "Log Ratio (F:B Ratio)", main = "Forest Plot: Firmicutes:Bacteroidetes Ratio")

# Step 12: Leave-one-out

loo\_firm <- leave1out(res\_firm)

loo\_bact <- leave1out(res\_bact)

loo\_fb <- leave1out(res\_fb)

# Add study labels

loo\_firm\_df <- as.data.frame(loo\_firm)

loo\_firm\_df$Study <- paste(df$Study\_ID, df$Treatment)

loo\_bact\_df <- as.data.frame(loo\_bact)

loo\_bact\_df$Study <- paste(df$Study\_ID, df$Treatment)

loo\_fb\_df <- as.data.frame(loo\_fb)

loo\_fb\_df$Study <- paste(df$Study\_ID, df$Treatment)

# Step 13: Save LOO plots

pdf("Firmicutes\_LOO.pdf")

ggplot(loo\_firm\_df, aes(x = reorder(Study, estimate), y = estimate)) +

geom\_point() +

geom\_errorbar(aes(ymin = ci.lb, ymax = ci.ub), width = 0.2) +

geom\_hline(yintercept = res\_firm$b[1], linetype = "dashed", color = "red") +

coord\_flip() +

labs(title = "Leave-One-Out (Firmicutes)", x = "Excluded Comparison", y = "Effect Size") +

theme\_minimal()

dev.off()

pdf("Bacteroidetes\_LOO.pdf")

ggplot(loo\_bact\_df, aes(x = reorder(Study, estimate), y = estimate)) +

geom\_point() +

geom\_errorbar(aes(ymin = ci.lb, ymax = ci.ub), width = 0.2) +

geom\_hline(yintercept = res\_bact$b[1], linetype = "dashed", color = "red") +

coord\_flip() +

labs(title = "Leave-One-Out (Bacteroidetes)", x = "Excluded Comparison", y = "Effect Size") +

theme\_minimal()

dev.off()

pdf("FB\_Ratio\_LOO.pdf")

ggplot(loo\_fb\_df, aes(x = reorder(Study, estimate), y = estimate)) +

geom\_point() +

geom\_errorbar(aes(ymin = ci.lb, ymax = ci.ub), width = 0.2) +

geom\_hline(yintercept = res\_fb$b[1], linetype = "dashed", color = "red") +

coord\_flip() +

labs(title = "Leave-One-Out (F:B Ratio)", x = "Excluded Comparison", y = "Effect Size") +

theme\_minimal()

dev.off()

# Step 14: Save data outputs

output\_list <- list(

Effect\_Sizes = df %>% select(Study\_ID, Treatment, yi\_firm, se\_firm, yi\_bact, se\_bact, yi\_fb, se\_fb),

LOO\_Firmicutes = loo\_firm\_df,

LOO\_Bacteroidetes = loo\_bact\_df,

LOO\_FB\_Ratio = loo\_fb\_df

)

write\_xlsx(output\_list, path = "logresponse\_Analysis\_Results.xlsx")

# Step 14: Summarize meta-analysis results

summary\_stats <- function(res, label) {

s <- summary(res)

data.frame(

Measure = label,

Estimate = round(s$b[1], 4),

CI\_Lower = round(s$ci.lb, 4),

CI\_Upper = round(s$ci.ub, 4),

Tau2 = round(s$tau2, 4),

I2 = round(s$I2, 2),

Q = round(s$QE, 4),

Q\_pval = signif(s$QEp, 4),

stringsAsFactors = FALSE

)

}

summary\_df <- bind\_rows(

summary\_stats(res\_firm, "Firmicutes"),

summary\_stats(res\_bact, "Bacteroidetes"),

summary\_stats(res\_fb, "F:B Ratio")

)

# Step 15: Add this to the Excel output

output\_list$Meta\_Analysis\_Summary <- summary\_df

# Overwrite Excel file with new sheet included

write\_xlsx(output\_list, path = "logresponse\_Analysis\_Results.xlsx")