# Load the required package

library(data.table)

# Set working directory where all SNP files are located

setwd("…") # your working place

# Define a consistent vector of trait names (same order used later)

trait\_names <- c("LPNEB", "LEDS", "LB\_BHB", "NEFA", "LIGF-1", "GLU", "LM\_BHB",

"CIT", "LACE", "C10", "C14", "C16", "C18", "C18:1cis-9",

"SCFA", "MCFA", "LCFA", "FP", "PP", "MY")

# Step 1: Read all SNP files into a named list

snp\_list <- lapply(trait\_names, function(trait) {

fread(paste0("snp\_", trait))

})

names(snp\_list) <- trait\_names

# Step 2: Extract SNP information and effect columns

# - Assume column 3: SNP ID, 4: Chr, 5: Pos, 6: Effect, 7: Weight/Accuracy

# Get basic SNP information and LPNEB effect (columns 3–6)

snp\_info <- snp\_list[["LPNEB"]][, .SD, .SDcols = 3:6]

# Extract effect columns (column 6) from the remaining traits

snp\_effect\_matrix <- do.call(cbind, lapply(trait\_names[-1], function(trait) {

snp\_list[[trait]][[6]] # extract 6th column (effect)

}))

# Combine SNP info with all effect values

merged\_snp <- cbind(snp\_info, snp\_effect\_matrix)

# Step 3: Extract SNP weights (column 7)

snp\_weight\_matrix <- do.call(cbind, lapply(trait\_names, function(trait) {

snp\_list[[trait]][[7]]

}))

# Step 4: Assign column names to the final SNP effect data frame

colnames(merged\_snp) <- c("snp", "chr", "pos", trait\_names)

# Final dataset

snp\_df <- merged\_snp

# Step 5: Extract only the trait-by-SNP effect matrix for analysis

snp\_df1 <- snp\_df[, trait\_names, with = FALSE] # same as snp\_df[,4:23]

# Preview (optional)

head(snp\_df)

# SNP-Based Weighted Correlation Analysis for LPNEB and LEDS

# Supplementary Code (JDS Format)

library(pheatmap)

library(RColorBrewer)

# Set working directory to location of input files

setwd("…") # your working place

# Load SNP effect matrix (20 traits) and allele frequencies

snp\_effects <- fread("snp\_df1.csv") # Contains 20 columns of SNP effects

snp\_chr\_info <- fread("snp\_chr\_info.csv") # Contains chromosome IDs for each SNP (column: chr)

freq\_data <- fread("freqdata.count-NEB") # Allele frequency (column: freq)

# Combine data into a matrix

snp\_matrix <- cbind(as.matrix(snp\_effects), chr = snp\_chr\_info$chr, freq = freq\_data$V2)

# Function to compute 2pq-weighted correlation between two traits

calculate\_r12 <- function(snp1, snp2, freq) {

V1 <- 2 \* freq \* (1 - freq) \* snp1^2

V2 <- 2 \* freq \* (1 - freq) \* snp2^2

Cov12 <- 2 \* freq \* (1 - freq) \* snp1 \* snp2

r12 <- sum(Cov12) / sqrt(sum(V1) \* sum(V2))

return(r12)

}

# Chromosome-specific r12 calculation

calculate\_r12\_chr <- function(matrix, trait1, trait2, freq\_col, chr\_col, chr\_id) {

subset <- matrix[matrix[, chr\_col] == chr\_id, ]

calculate\_r12(subset[, trait1], subset[, trait2], subset[, freq\_col])

}

# Compute genome-wide r12 correlation matrix

compute\_r12\_matrix <- function(matrix, trait\_cols, freq\_col) {

n <- length(trait\_cols)

r\_mat <- matrix(NA, n, n)

diag(r\_mat) <- 1

for (i in 1:(n - 1)) {

for (j in (i + 1):n) {

r\_mat[i, j] <- calculate\_r12(matrix[, trait\_cols[i]], matrix[, trait\_cols[j]], matrix[, freq\_col])

r\_mat[j, i] <- r\_mat[i, j]

}

}

return(r\_mat)

}

# Compute r12 correlation matrices for each chromosome

compute\_chrwise\_r12 <- function(matrix, trait\_cols, freq\_col, chr\_col, n\_chr = 29) {

chr\_list <- vector("list", n\_chr)

for (chr in 1:n\_chr) {

mat <- matrix(NA, length(trait\_cols), length(trait\_cols))

diag(mat) <- 1

for (i in 1:(length(trait\_cols) - 1)) {

for (j in (i + 1):length(trait\_cols)) {

r <- calculate\_r12\_chr(matrix, trait\_cols[i], trait\_cols[j], freq\_col, chr\_col, chr)

mat[i, j] <- r

mat[j, i] <- r

}

}

chr\_list[[chr]] <- mat

}

return(chr\_list)

}

# Plot heatmap function

plot\_r12\_heatmap <- function(matrix, title) {

pheatmap(

matrix,

cluster\_rows = FALSE,

cluster\_cols = FALSE,

display\_numbers = TRUE,

number\_color = "black",

number.cex = 3,

color = colorRampPalette(c("red", "white", "green"))(500),

breaks = seq(-1, 1, length.out = 501),

fontsize\_row = 16,

fontsize\_col = 16,

main = title

)

}

# Define trait columns (1:20)

trait\_indices <- 1:20

freq\_col <- 22 # Allele frequency column

chr\_col <- 21 # Chromosome column

# Genome-wide r12 matrix

r12\_total <- compute\_r12\_matrix(snp\_matrix, trait\_indices, freq\_col)

colnames(r12\_total) <- colnames(snp\_effects)

rownames(r12\_total) <- colnames(snp\_effects)

# Per-chromosome r12 matrices

chr\_r12\_list <- compute\_chrwise\_r12(snp\_matrix, trait\_indices, freq\_col, chr\_col)

# Extract LPNEB and LEDS row-wise correlation per chromosome

extract\_chrwise\_trait <- function(cor\_list, row\_index, trait\_names) {

mat <- do.call(rbind, lapply(cor\_list, function(m) m[row\_index, ]))

rownames(mat) <- paste0("BTA", 1:29)

colnames(mat) <- trait\_names

return(mat)

}

lpneb\_chr\_mat <- extract\_chrwise\_trait(chr\_r12\_list, 1, colnames(snp\_effects))

leds\_chr\_mat <- extract\_chrwise\_trait(chr\_r12\_list, 2, colnames(snp\_effects))

# Combine with genome-wide values as first row

combined\_lpneb <- rbind(r12\_total[1, ], lpneb\_chr\_mat)

combined\_leds <- rbind(r12\_total[2, ], leds\_chr\_mat)

# Save results

fwrite(as.data.frame(combined\_lpneb), "LPNEB\_chrwise\_r12.csv")

fwrite(as.data.frame(combined\_leds), "LEDS\_chrwise\_r12.csv")

# Plot heatmaps

plot\_r12\_heatmap(combined\_lpneb[, -1], title = "LPNEB Correlations with 19 Traits")

plot\_r12\_heatmap(combined\_leds[, -1], title = "LEDS Correlations with 19 Traits")

# Chromosome-wise and Genome-wide Independent Contribution Analysis for LPNEB and LEDS

# Standardized for Supplementary Code in JDS Submission

library(gridExtra)

# Load genome-wide trait correlation matrix

df <- fread("all\_chr\_weightcor\_diag1.csv")

cor\_matrix <- as.matrix(df)

rownames(cor\_matrix) <- colnames(cor\_matrix)

# Define traits of interest

selected\_traits <- c("LPNEB", "LEDS", "LB\_BHB", "NEFA", "LIGF-1", "LM\_BHB", "CIT", "C10:0", "PP", "MY")

# Function to compute independent contribution using C^-1 \* beta

compute\_contribution <- function(target, traits, mat) {

others <- setdiff(traits, target)

beta <- mat[target, others] %\*% solve(mat[others, others])

contrib <- beta %\*% mat[others, others] %\*% t(beta)

df <- data.frame(trait = others, contribution = as.vector(beta))

df$percent <- df$contribution / sum(abs(df$contribution))

return(df)

}

# Calculate independent contributions genome-wide

df\_LPNEB <- compute\_contribution("LPNEB", setdiff(selected\_traits, "LEDS"), cor\_matrix)

df\_LEDS <- compute\_contribution("LEDS", setdiff(selected\_traits, "LPNEB"), cor\_matrix)

# Load chromosome-wise correlation matrices (assumed precomputed)

# correlation\_matrices is a list of 29 matrices, each 20x20

# Each matrix must match selected\_traits order

# Function for chromosome-specific contributions

compute\_chr\_contrib <- function(chr\_matrix, target\_row, trait\_indices) {

beta <- chr\_matrix[target\_row, trait\_indices] %\*% solve(chr\_matrix[trait\_indices, trait\_indices])

contrib <- beta %\*% chr\_matrix[trait\_indices, trait\_indices] %\*% t(beta)

df <- as.data.frame(t(beta))

return(list(contrib = as.numeric(contrib), beta = df))

}

# Define trait indices used in chromosome contribution analysis

trait\_indices <- c(3:5, 7:8, 10, 19:20) # Corresponds to selected traits index

trait\_names\_chr <- selected\_traits[trait\_indices]

# Loop through chromosomes

results <- lapply(1:29, function(chr) {

mat <- correlation\_matrices[[chr]]

lpneb <- compute\_chr\_contrib(mat, 1, trait\_indices)

leds <- compute\_chr\_contrib(mat, 2, trait\_indices)

lpneb$beta$traits <- trait\_names\_chr

leds$beta$traits <- trait\_names\_chr

list(lpneb = lpneb$beta, leds = leds$beta, lpneb\_total = lpneb$contrib, leds\_total = leds$contrib)

})

# Extract and reshape results

lpneb\_df <- do.call(cbind, lapply(1:29, function(i) {

df <- results[[i]]$lpneb

df$Chromosome <- i

return(df)

}))

leds\_df <- do.call(cbind, lapply(1:29, function(i) {

df <- results[[i]]$leds

df$Chromosome <- i

return(df)

}))

# Extract effects and format

extract\_effect\_matrix <- function(df) {

index <- seq(1, ncol(df), 3)

mat <- df[, c(2, index)]

colnames(mat)[2:30] <- paste0("BTA", 1:29)

rownames(mat) <- mat$traits

return(as.data.frame(t(mat[, -1])))

}

lpneb\_mat <- extract\_effect\_matrix(lpneb\_df)

leds\_mat <- extract\_effect\_matrix(leds\_df)

# Normalize rows to show relative contribution per chromosome

row\_normalize <- function(df) df / rowSums(abs(df), na.rm = TRUE)

lpneb\_norm <- as.data.frame(row\_normalize(lpneb\_mat))

leds\_norm <- as.data.frame(row\_normalize(leds\_mat))

# Add genome-wide contributions as first row

lpneb\_all <- rbind("All\_BTA" = setNames(as.vector(df\_LPNEB$percent), df\_LPNEB$trait), lpneb\_norm)

leds\_all <- rbind("All\_BTA" = setNames(as.vector(df\_LEDS$percent), df\_LEDS$trait), leds\_norm)

# Global min/max for consistent color scale

all\_vals <- c(unlist(lpneb\_all), unlist(leds\_all))

breaks\_range <- seq(-0.5, 0.5, length.out = 501)

color\_gradient <- colorRampPalette(c("red", "white", "green"))(500)

# Plot heatmaps

p1 <- pheatmap(lpneb\_all, cluster\_rows = FALSE, cluster\_cols = FALSE, display\_numbers = TRUE,

number\_color = "black", number.cex = 2.0, fontsize\_row = 16, fontsize\_col = 16,

color = color\_gradient, breaks = breaks\_range, main = "Normalized Contribution to LPNEB", silent = TRUE)

p2 <- pheatmap(leds\_all, cluster\_rows = FALSE, cluster\_cols = FALSE, display\_numbers = TRUE,

number\_color = "black", number.cex = 2.0, fontsize\_row = 16, fontsize\_col = 16,

color = color\_gradient, breaks = breaks\_range, main = "Normalized Contribution to LEDS", silent = TRUE)

# Display side-by-side

grid.arrange(p1$gtable, p2$gtable, ncol = 2)