# Statistical Inference for High-Dimensional Models with Applications to Imaging Genetics

# Purpose: simulate SNP matrix (10,000), imaging features (300), and binary disease label

set.seed(123)

# sample size and dimensions

n <- 200 # number of subjects (adjustable)

p\_snp <- 10000 # number of SNPs

p\_img <- 300 # number of imaging features

# --- Simulate genotypes ---

# simulate minor allele frequencies for each SNP

maf <- runif(p\_snp, min = 0.05, max = 0.5)

# genotype counts per subject (0,1,2) assuming HWE-like Binomial(2, maf)

# we create a matrix of size n x p\_snp

geno\_vec <- rbinom(n \* p\_snp, size = 2, prob = rep(maf, each = n))

G <- matrix(geno\_vec, nrow = n, ncol = p\_snp)

colnames(G) <- paste0("SNP", seq\_len(p\_snp))

rownames(G) <- paste0("ID", seq\_len(n))

# --- Define causal SNPs and effects on imaging ---

k\_causal\_snp <- 10 # number of causal SNPs

causal\_snps <- sample(seq\_len(p\_snp), k\_causal\_snp)

# each causal SNP will influence a small subset of imaging features

effects\_snp\_to\_img <- matrix(0, nrow = k\_causal\_snp, ncol = p\_img)

for (i in seq\_len(k\_causal\_snp)) {

affected\_imgs <- sample(seq\_len(p\_img), size = 5) # each SNP affects 5 imaging features

effects\_snp\_to\_img[i, affected\_imgs] <- rnorm(length(affected\_imgs), mean = 0.3, sd = 0.15)

}

rownames(effects\_snp\_to\_img) <- paste0("SNP", causal\_snps)

colnames(effects\_snp\_to\_img) <- paste0("IMG", seq\_len(p\_img))

# --- Simulate imaging features as linear functions of the causal SNPs + noise ---

# X\_img = G\_causal %\*% effects + Gaussian noise

G\_causal <- G[, causal\_snps, drop = FALSE] # n x k\_causal\_snp

# imaging matrix (n x p\_img)

Img\_mean\_part <- G\_causal %\*% effects\_snp\_to\_img

Noise <- matrix(rnorm(n \* p\_img, mean = 0, sd = 1), nrow = n, ncol = p\_img)

Img <- Img\_mean\_part + Noise

colnames(Img) <- paste0("IMG", seq\_len(p\_img))

rownames(Img) <- rownames(G)

# --- Simulate binary disease label depending on a few imaging features ---

k\_causal\_img\_for\_disease <- 8

causal\_imgs\_for\_disease <- sample(seq\_len(p\_img), k\_causal\_img\_for\_disease)

beta\_img\_to\_disease <- numeric(p\_img)

beta\_img\_to\_disease[causal\_imgs\_for\_disease] <- rnorm(k\_causal\_img\_for\_disease, mean = 0.6, sd = 0.25)

# linear predictor and logistic model

linpred <- Img %\*% beta\_img\_to\_disease

prob\_disease <- 1 / (1 + exp(-linpred))

# add intercept to control baseline prevalence

intercept <- -1.5

prob\_disease <- 1 / (1 + exp(-(intercept + linpred)))

Y <- rbinom(n, size = 1, prob = prob\_disease)

# --- Package into data.frames and save ---

SNPs\_df <- as.data.frame(G)

Imgs\_df <- as.data.frame(Img)

meta\_df <- data.frame(ID = rownames(G), disease = Y)

# Save for later steps

saveRDS(list(SNPs = SNPs\_df, Imaging = Imgs\_df, meta = meta\_df,

causal\_snps = causal\_snps,

effects\_snp\_to\_img = effects\_snp\_to\_img,

causal\_imgs\_for\_disease = causal\_imgs\_for\_disease,

beta\_img\_to\_disease = beta\_img\_to\_disease),

file = "simulated\_imaging\_genetics\_data\_step1.rds")

cat("Simulation complete. Objects saved to 'simulated\_imaging\_genetics\_data\_step1.rds'\n")

# Purpose: Test associations between imaging features (300) and SNPs (10,000)

# using univariate linear regressions + multiple testing corrections (FDR, Bonferroni, Holm)

# Load simulated data from Step 1

data\_list <- readRDS("simulated\_imaging\_genetics\_data\_step1.rds")

SNPs <- data\_list$SNPs

Imgs <- data\_list$Imaging

meta <- data\_list$meta

n <- nrow(SNPs)

p\_snp <- ncol(SNPs)

p\_img <- ncol(Imgs)

# --- STEP 2.1: Test association between each imaging feature and each SNP ---

# We'll test each SNP against the average imaging value (as a simplified first test)

# In next sub-steps, we can refine to feature-wise association matrices.

img\_mean <- rowMeans(Imgs) # average cortical thickness per subject

# Function to test one SNP vs imaging mean via linear regression

get\_pval <- function(x, y) {

fit <- lm(y ~ x)

summary(fit)$coefficients[2, 4] # p-value for SNP coefficient

}

# Apply to all SNPs (this will take time for large data — use first 2000 SNPs demo)

max\_snp <- 2000

pvals <- sapply(SNPs[, 1:max\_snp], get\_pval, y = img\_mean)

# --- STEP 2.2: Multiple testing corrections ---

# Bonferroni correction

bonf\_p <- p.adjust(pvals, method = "bonferroni")

# Holm correction

holm\_p <- p.adjust(pvals, method = "holm")

# Benjamini-Hochberg FDR

bh\_p <- p.adjust(pvals, method = "BH")

# --- STEP 2.3: Summarize results ---

summary\_table <- data.frame(

SNP = names(pvals),

raw\_p = pvals,

bonf\_p = bonf\_p,

holm\_p = holm\_p,

bh\_p = bh\_p

)

sig\_bonf <- sum(bonf\_p < 0.05)

sig\_holm <- sum(holm\_p < 0.05)

sig\_bh <- sum(bh\_p < 0.05)

cat("\nSummary of significant SNPs (alpha = 0.05):\n")

cat("Bonferroni:", sig\_bonf, "\nHolm:", sig\_holm, "\nBH (FDR):", sig\_bh, "\n")

# --- STEP 2.4: Visualization of -log10(p-values) ---

plot(-log10(pvals), type = "h", main = "SNP associations with imaging (mean feature)",

xlab = "SNP index", ylab = "-log10(p-value)")

abline(h = -log10(0.05 / max\_snp), col = "red", lty = 2) # Bonferroni line

cat("\nUnivariate tests complete. Next: feature-wise associations or LASSO high-dimensional modeling.\n")

# Purpose: Identify SNPs associated with imaging features

# using LASSO and Elastic Net regression (via glmnet)

# --- Load packages ---

library(glmnet)

# --- Load the simulated data from Step 1 ---

data\_list <- readRDS("simulated\_imaging\_genetics\_data\_step1.rds")

SNPs <- as.matrix(data\_list$SNPs)

Imgs <- as.matrix(data\_list$Imaging)

meta <- data\_list$meta

n <- nrow(SNPs)

p\_snp <- ncol(SNPs)

p\_img <- ncol(Imgs)

cat("Loaded data with", n, "subjects,", p\_snp, "SNPs and", p\_img, "imaging features.\n")

# --- STEP 3.1: Choose an imaging phenotype ---

# We'll model the first imaging feature (can change to another column if desired)

y <- Imgs[, 1]

X <- SNPs

# --- STEP 3.2: LASSO Regression ---

set.seed(123)

lasso\_fit <- cv.glmnet(X, y, alpha = 1, nfolds = 5, standardize = TRUE)

# Plot cross-validation curve

plot(lasso\_fit)

title("LASSO Cross-Validation Curve", line = 2.5)

# Extract coefficients at lambda.min

coef\_lasso <- coef(lasso\_fit, s = "lambda.min")

selected\_lasso <- which(coef\_lasso != 0)

selected\_snp\_lasso <- rownames(coef\_lasso)[selected\_lasso]

cat("\nLASSO selected", length(selected\_snp\_lasso) - 1, "SNPs (excluding intercept).\n")

# --- STEP 3.3: Elastic Net Regression ---

set.seed(123)

elastic\_fit <- cv.glmnet(X, y, alpha = 0.5, nfolds = 5, standardize = TRUE)

# Plot cross-validation curve

plot(elastic\_fit)

title("Elastic Net Cross-Validation Curve", line = 2.5)

# Extract coefficients

coef\_elastic <- coef(elastic\_fit, s = "lambda.min")

selected\_elastic <- which(coef\_elastic != 0)

selected\_snp\_elastic <- rownames(coef\_elastic)[selected\_elastic]

cat("Elastic Net selected", length(selected\_snp\_elastic) - 1, "SNPs (excluding intercept).\n")

# --- STEP 3.4: Compare results ---

common\_snp <- setdiff(intersect(selected\_snp\_lasso, selected\_snp\_elastic), "(Intercept)")

cat("\nCommon SNPs between LASSO and Elastic Net:", length(common\_snp), "\n")

print(common\_snp)

# --- STEP 3.5: Visualize coefficient magnitudes ---

par(mfrow = c(1, 2))

plot(coef\_lasso[-1], main = "LASSO Coefficients",

ylab = "Effect size", xlab = "SNP index", pch = 16)

plot(coef\_elastic[-1], main = "Elastic Net Coefficients",

ylab = "Effect size", xlab = "SNP index", pch = 16)

par(mfrow = c(1, 1))

cat("\n✅ Step 3 complete: LASSO and Elastic Net modeling done.\nNext step: Apply Knockoff Filter for controlled FDR.\n")

# Purpose: Apply Knockoff Filter for controlled FDR in SNP selection

# --- Load packages ---

library(knockoff)

library(glmnet)

# --- Load simulated data ---

data\_list <- readRDS("simulated\_imaging\_genetics\_data\_step1.rds")

SNPs <- as.matrix(data\_list$SNPs)

Imgs <- as.matrix(data\_list$Imaging)

meta <- data\_list$meta

n <- nrow(SNPs)

p\_snp <- ncol(SNPs)

cat("Loaded data with", n, "subjects and", p\_snp, "SNPs.\n")

# --- STEP 4.1: Choose an imaging phenotype again (same as Step 3) ---

y <- Imgs[, 1]

X <- scale(SNPs) # standardized SNPs

# --- STEP 4.2: Construct knockoff variables ---

# The knockoff package automatically builds fake copies ("knockoffs")

# that preserve correlations among predictors but are independent of y.

knockoff\_X <- create.gaussian(X, mu = rep(0, p\_snp), Sigma = cor(X))

# --- STEP 4.3: Define a test statistic function using LASSO ---

stat\_lasso <- function(X, Xk, y) {

# Fit LASSO on combined (X + knockoffs)

X\_combined <- cbind(X, Xk)

fit <- glmnet(X\_combined, y, alpha = 1, standardize = TRUE)

# Compute feature importance as max lambda where coefficient becomes nonzero

W <- stat.glmnetCoefDiff(X, Xk, y)

return(W)

}

# --- STEP 4.4: Apply the knockoff filter ---

set.seed(123)

W\_stats <- stat\_lasso(X, knockoff\_X, y)

selected\_knockoff <- knockoff.filter(X, y, fdr = 0.1, statistic = stat.glmnetCoefDiff)

cat("\nKnockoff Filter selected", length(selected\_knockoff$selected),

"SNPs controlling FDR at 10%.\n")

# --- STEP 4.5: View selected SNPs ---

print(selected\_knockoff$selected[1:20]) # show first few if many

cat("\n✅ Step 4 complete: Knockoff Filter applied.\n")

# --- STEP 4.6: Optional visualization ---

W\_sorted <- sort(W\_stats, decreasing = TRUE)

plot(W\_sorted, type = "h", main = "Knockoff Statistics (W)",

ylab = "Importance (W)", xlab = "SNP index")

abline(h = 0, col = "red", lty = 2)

# Purpose: Compare classical, LASSO/ENet, and Knockoff methods

# --- Load required libraries ---

library(ggplot2)

library(UpSetR)

# --- Load previously saved data ---

data\_list <- readRDS("simulated\_imaging\_genetics\_data\_step1.rds")

SNPs <- as.matrix(data\_list$SNPs)

Imgs <- as.matrix(data\_list$Imaging)

# --- Load results from previous steps ---

# (These should have been saved in previous steps, or you can reassign them manually)

# Example placeholders — replace these with your actual variable names if different

selected\_univariate <- which(p.adjust(cor(Imgs[, 1], SNPs)^2, method = "BH") < 0.05)

selected\_lasso <- selected\_features # from Step 3

selected\_knockoff <- selected\_knockoff$selected # from Step 4

# --- STEP 5.1: Compare selection overlap ---

comparison\_list <- list(

Univariate = selected\_univariate,

LASSO = selected\_lasso,

Knockoff = selected\_knockoff

)

# --- STEP 5.2: Create an UpSet plot (Venn for large sets) ---

upset(fromList(comparison\_list),

nsets = 3,

order.by = "freq",

mainbar.y.label = "Intersection Size",

sets.x.label = "Features Selected per Method")

# --- STEP 5.3: Summary statistics ---

cat("Summary of selected SNPs:\n")

cat("Univariate FDR-BH:", length(selected\_univariate), "SNPs\n")

cat("LASSO:", length(selected\_lasso), "SNPs\n")

cat("Knockoff:", length(selected\_knockoff), "SNPs\n")

# --- STEP 5.4: Visualization of selection frequencies ---

# Count how many methods selected each SNP

all\_indices <- 1:ncol(SNPs)

selection\_count <- rowSums(sapply(comparison\_list, function(s) all\_indices %in% s))

selection\_df <- data.frame(

SNP = all\_indices,

SelectedBy = selection\_count

)

ggplot(selection\_df, aes(x = SNP, y = SelectedBy)) +

geom\_col() +

theme\_minimal() +

labs(

title = "SNPs Selected Across Methods",

x = "SNP Index",

y = "Number of Methods that Selected SNP"

# Purpose: Visualize discovered imaging–genetic associations

# --- Load required libraries ---

library(pheatmap)

library(ggplot2)

# --- Load simulated data ---

data\_list <- readRDS("simulated\_imaging\_genetics\_data\_step1.rds")

SNPs <- as.matrix(data\_list$SNPs)

Imgs <- as.matrix(data\_list$Imaging)

# --- Combine results from previous steps ---

# Replace variable names if yours differ

selected\_univariate <- which(p.adjust(cor(Imgs[, 1], SNPs)^2, method = "BH") < 0.05)

selected\_lasso <- selected\_features # from Step 3

selected\_knockoff <- selected\_knockoff$selected # from Step 4

# Union of all selected SNPs

all\_selected <- unique(c(selected\_univariate, selected\_lasso, selected\_knockoff))

cat("Total unique SNPs selected across methods:", length(all\_selected), "\n")

# --- STEP 6.1: Compute correlations between selected SNPs and imaging features ---

corr\_mat <- cor(SNPs[, all\_selected, drop = FALSE], Imgs)

# --- STEP 6.2: Plot a heatmap ---

pheatmap(corr\_mat,

main = "Imaging–Genetic Association Heatmap",

cluster\_rows = TRUE,

cluster\_cols = TRUE,

color = colorRampPalette(c("blue", "white", "red"))(100),

fontsize = 8)

# --- STEP 6.3: Summarize associations ---

avg\_corr <- rowMeans(abs(corr\_mat))

summary\_df <- data.frame(

SNP\_Index = all\_selected,

Avg\_Abs\_Corr = avg\_corr

)

summary\_df <- summary\_df[order(-summary\_df$Avg\_Abs\_Corr), ]

head(summary\_df)

# --- STEP 6.4: Plot strongest SNPs ---

top\_snps <- head(summary\_df, 20)

ggplot(top\_snps, aes(x = factor(SNP\_Index), y = Avg\_Abs\_Corr)) +

geom\_col(fill = "steelblue") +

coord\_flip() +

theme\_minimal() +

labs(

title = "Top 20 SNPs Most Associated with Imaging Features",

x = "SNP Index",

y = "Average |Correlation|"

)

# --- Optional: Save summary ---

write.csv(summary\_df, "final\_imaging\_genetic\_summary.csv", row.names = FALSE)

cat("✅ Step 6 complete — visualization and summary saved.\n")