GDRsimulationStats

Nora Bull

18 11 2021

# Willow statistics simulation

## Function to calculate and save p-values for simulated GDR

GDRsim\_pval\_calcSave <- function(GDRnull\_file, pvalSaveFile) {  
   
 "  
 Input:  
 GDRnull\_file: path to file with random GDR values   
 Format: CSV  
   
 pval\_save\_file: desired filepath to store values  
 Format: CSV  
 "  
   
 # Read data  
 GDRnull\_data <- read.csv(GDRnull\_file)  
 col = colnames(GDRnull\_data)  
 data = unlist(strsplit(col, '\_'))  
 G = gsub("G", "", data[1])  
 perm = gsub("P", "", data[2])  
 GDR = as.double(gsub("GDR", "", data[3]))  
 uniq\_pvals = length(unique(GDRnull\_data[,1]))  
   
 # Create empirical cumulative distribution function (ecdf)  
 P = ecdf(GDRnull\_data[,1])  
   
 # Calculate p value for GDR  
 p\_val = P(GDR)  
  
 # Save p val to file  
 #GDR <- as.character(GDR)  
 cat(c(G, perm, GDR, uniq\_pvals, p\_val), file = pvalSaveFile, sep = ",", append = TRUE)  
 cat('\n', file = pvalSaveFile, append=TRUE)  
}

# Run p-value calculation

# Create file to save p-values:  
save\_file <- "C:/Users/norab/Master/thesis\_data/simulation/simNull\_pvals\_24.11.21.csv"  
file.create(save\_file)

## [1] TRUE

cat(c("group\_size,permutations,GDR,n\_uniq\_pvals,pval\n"), file = save\_file, append=TRUE)  
  
# Create list of files to process (each containing random GDRs for one simulated GDR values)  
simNull\_files <- list.files(path="C:/Users/norab/Master/thesis\_data/simulation/simData/", full.names=TRUE, recursive=FALSE)  
  
# Apply GDRsim\_pval\_calcSave function to all simNull\_files  
run <- sapply(simNull\_files, GDRsim\_pval\_calcSave, save\_file)

# Multiple testing correction

# Read data  
all\_data <- read.csv(save\_file, sep=",")  
  
# adjust  
all\_data$pval\_adj\_holm <- p.adjust(all\_data$pval)  
all\_data$pval\_adj\_bonferroni <- p.adjust(all\_data$pval, method = "bonferroni")  
all\_data$hommel <- p.adjust(all\_data$pval, method = "hommel")  
  
# Save  
save\_to <- 'C:/Users/norab/Master/thesis\_data/simulation/simNull\_alldata\_24.11.21.csv'  
write.csv(all\_data, save\_to, row.names = FALSE)

# Plot CDF for four simulated instances

Retrieve ecdf and p\_val for 4 files:

The 4 scenarios of simulated GDR value distributions are plotted: G = number of group samples P = number of permuted samples

1. Low G, low P
2. Low G, high P
3. High G, low P
4. High G, low P

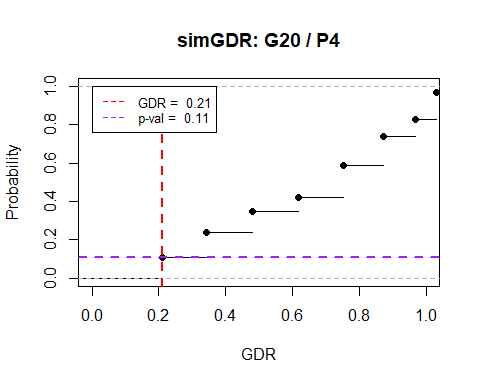
Low = ~20 % permuted samples High = ~80 % permuted samples

## Function to plot ecdf and

ecdf\_pval <- function(GDRnullFile, title) {  
   
 # Read data  
 GDRnullData <- read.csv(GDRnullFile)  
 col = colnames(GDRnullData)  
 data = unlist(strsplit(col, '\_'))  
 G = gsub("G", "", data[1])  
 perm = gsub("P", "", data[2])  
 GDR = as.double(gsub("GDR", "", data[3]))  
   
 # Make ecfd  
 P = ecdf(GDRnullData[,1])  
 # calc p-value  
 p\_val = P(GDR)  
   
 # Plot  
 title = paste0("simGDR: ", title)  
 plot(P, main = title, xlim=c(0,1), ylab="Probability", xlab="GDR")   
   
 abline(v=GDR, col="red", lwd = 2, lty = 2)  
 abline(h=p\_val, col="purple", lwd = 2, lty = 2)  
 GDRlab = paste("GDR = ",round(GDR, 2))  
 p\_valLab = paste("p-val = ", p\_val)  
 legend(x=0, y=1,c(GDRlab, p\_valLab), col=c("red", "purple"), lty=2, cex=0.8)  
   
 print(p\_val)  
 return(GDR)}

## Plot

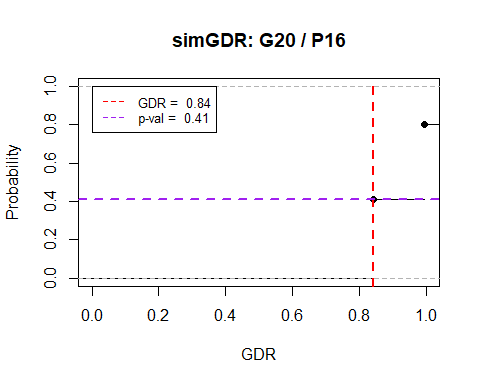
# Files to plot  
randSimG20\_P4 <- "C:/Users/norab/Master/thesis\_data/simulation/simData/randSimG20\_P4.csv"  
randSimG20\_P16 <- "C:/Users/norab/Master/thesis\_data/simulation/simData/randSimG20\_P16.csv"  
randSimG190\_P38 <- "C:/Users/norab/Master/thesis\_data/simulation/simData/randSimG190\_P38.csv"  
randSimG190\_P152 <- "C:/Users/norab/Master/thesis\_data/simulation/simData/randSimG190\_P152.csv"  
  
# Plot  
ecdf\_pval(randSimG20\_P4, title = "G20 / P4")



## [1] 0.11

## [1] 0.2105263

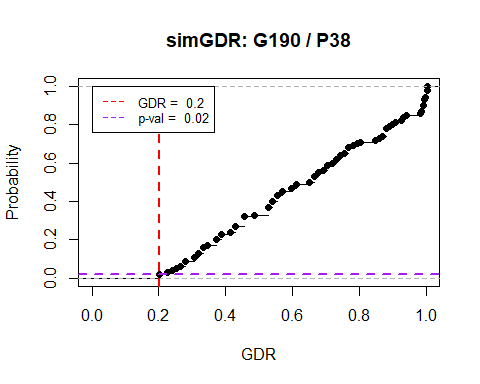
ecdf\_pval(randSimG20\_P16, title = "G20 / P16")



## [1] 0.41

## [1] 0.8421053

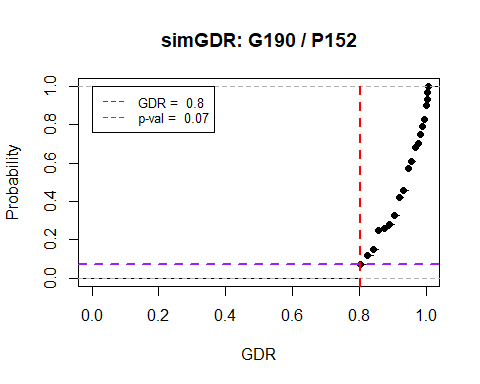
ecdf\_pval(randSimG190\_P38, title = "G190 / P38")



## [1] 0.02

## [1] 0.2010582

ecdf\_pval(randSimG190\_P152, title = "G190 / P152")



## [1] 0.07

## [1] 0.8042328