Jan Graffelman Script

Jerome Goudet and Bruce Weir

2022-04-14

knitr::opts\_chunk$set(echo = TRUE)#,results="hide")#,fig.show="hide")  
#knitr::opts\_chunk$set(echo = TRUE)

setwd("C:/Users/bsweir/Dropbox/awork/6SUMMERINSTITUTE/PlantAndFoodNZ2022/2.HardyWeinberg/Software")  
getwd()

## [1] "C:/Users/bsweir/Dropbox/awork/6SUMMERINSTITUTE/PlantAndFoodNZ2022/2.HardyWeinberg/Software"

# AMD Data

#fn <- “c:/other/data/AMD/AMD.txt”

AMD <- read.table(“AMD.txt”,header=TRUE)

#AMD <- read.table(fn,header=TRUE)

dim(AMD)

control1<-AMD[which(AMD$Chr==1),101:150] #extract data from chrom1 only and for the controls

ncontrols<-dim(control1)[2] ncontrols head(control1)

cAA<-apply(control1,1,fun<-function(x) sum(x==1)) #homAA

cAB<-apply(control1,1,fun<-function(x) sum(x==2)) #het

cBB<-apply(control1,1,fun<-function(x) sum(x==3)) #homBB

c00<-apply(control1,1,fun<-function(x) sum(x==0)) #missing

control.counts<-cbind(cAA,cAB,cBB)

dim(control.counts)

head(control.counts)

#using HardyWeinberg package

library(HardyWeinberg)

# HWExactStats is thought for larger sets of markers.

pvalues <- HWExactStats(control.counts) pvalues

colnames(control.counts) <- c(“AA”,“AB”,“BB”) head(control.counts)

# Your alternative with “apply”, slightly corrected.

pvalc<-apply(control.counts,1,function(x) HardyWeinberg::HWExact(as.vector(x),verbose=TRUE)$pval)

head(cbind(pvalues,pvalc))

pvalues

qqunif <- function (x, logplot = FALSE, lbs = 1:length(x), texton = FALSE, mm = NULL, main = “Q-Q plot for a uniform distribution”, plotlines = 0, xlab = “Expected p-value”, ylab = “Observed p-value”, colvec = rep(“black”, length(x)), …) { xs <- sort(x, index.return = TRUE) colvec <- colvec[xsx n <- length(x) epvals <- (1:n - 0.5)/n if (!logplot) { plot(epvals, pvals, xlab = “Expected p-value”, ylab = “Observed p-value”, xlim = c(0, 1), ylim = c(0, 1), main = main, col = colvec, …) if (texton) text(epvals, pvals, lbs) qp <- quantile(pvals, c(0.25, 0.75)) qe <- quantile(epvals, c(0.25, 0.75)) slope <- diff(qp)/diff(qe) int <- qp[1] - slope \* qe[1] if (plotlines == 1) { abline(0, 1, lwd = 1, col = “green”) } if (plotlines == 2) { abline(0, 1, lwd = 1, col = “green”) abline(int, slope, lwd = 1, col = “red”) } } else { lpvals <- -log10(pvals) lepvals <- -log10(epvals) if (is.null(mm)) mm <- max(c(lpvals, lepvals)) plot(lepvals, lpvals, xlab = xlab, ylab = ylab, ylim = c(0, mm), main = main, col = colvec, …) if (texton) text(lepvals, lpvals, lbs) qp <- quantile(lpvals, c(0.25, 0.75)) qe <- quantile(lepvals, c(0.25, 0.75)) slope <- diff(qp)/diff(qe) int <- qp[1] - slope \* qe[1] if (plotlines == 1) { abline(0, 1, lwd = 1, col = “black”) } if (plotlines == 2) { abline(0, 1, lwd = 1, col = “green”) abline(int, slope, lwd = 1, col = “red”) } } return(list(pvals = pvals, epvals = epvals)) }

out.qq <- qqunif(pvalues) out.qq <- qqunif(pvalues,logplot=TRUE,plotlines=1)