

Figure 1 for Deleterious Alleles paper

```
#setwd("~/Documents/Github/pvpDiallel/")
knitr::opts_knit$set(root.dir=normalizePath('../'))
```

To reproduce the figures, we should set `pvpDiallel` as the root path, i.e. `setwd("~/Documents/Github/pvpDiallel/")`. And then use `knitr` package to knit a pdf file. Or simply click Knit PDF icon in RStudio. Note, to generate each panel into separate pdf files, we should turn `getpdf` into `TRUE` (i.e. `getpdf=TRUE`) when calling the plotting functions.

First of all, determine font size and set the `getpdf` option:

```
fs <- 1.3 # times bigger than default
getpdf <- FALSE # get figures in separated pdf [TRUE] or not [FALSE]
```

Figure 1a

```
plotH2 <- function(getpdf=TRUE, outfile, ...){
  H2 <- read.csv("data/DIalleleHeritability.csv")
  h <- read.csv("cache/loh_pBPHmax_median.csv")
  H2 <- merge(H2, h, by.x="Traits", by.y="trait")
  H2 <- H2[order(H2$h), ]

  barplot(H2[,4], names.arg = H2$Traits, ...)
  if(getpdf == TRUE){
    pdf(outfile, width=6, height=6)
    barplot(H2[,4], names.arg = H2$Traits, ...)
    box()
    dev.off()
  }
}

plotH2(getpdf=getpdf, outfile="graphs/Fig1a_v3.pdf",
       ylim=c(0, 1), col="antiquewhite3", ylab="Heritability",
       cex.axis=fs, cex.names=fs, cex.lab=fs)
```

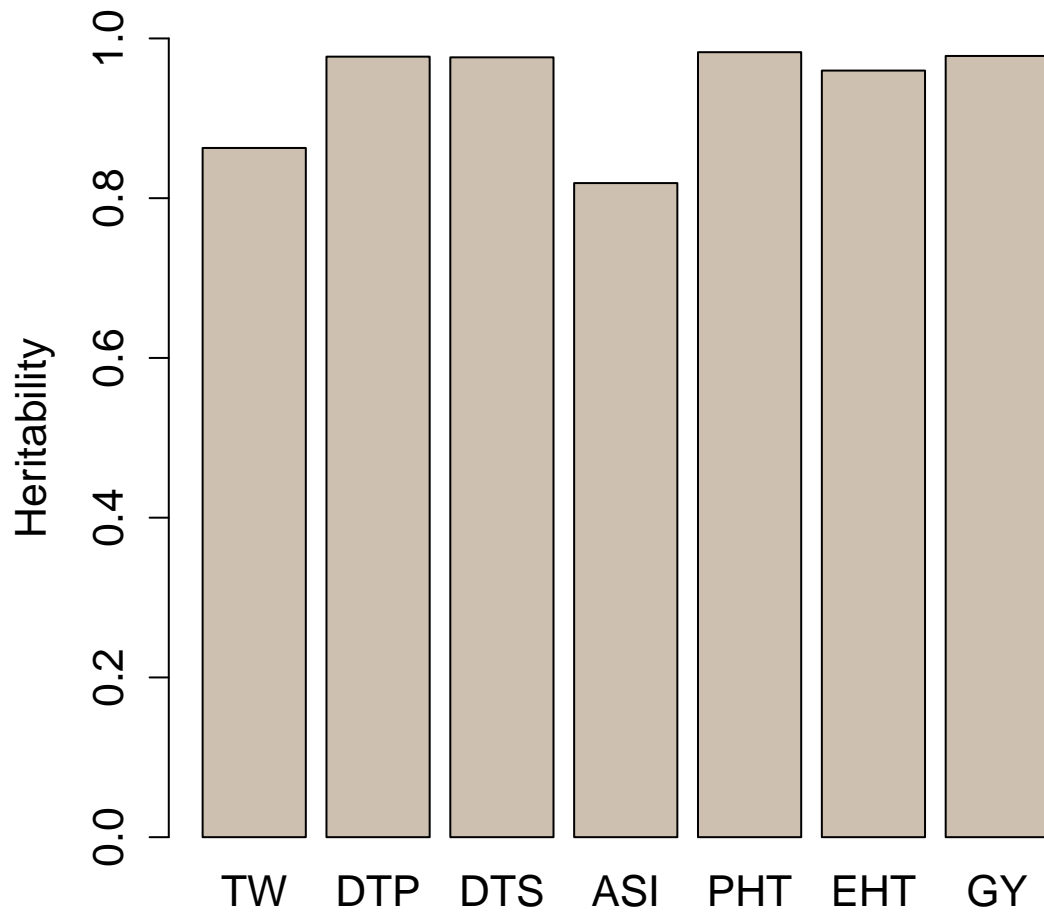


Figure 1b

```
plotloh <- function(getpdf=TRUE, outfile, ...){
  trait <- read.csv("cache/STable_heterosis.csv")
  bymed2 <- with(trait, reorder(trait, pBPH, median))
  boxplot(pBPH ~ bymed2, data=trait,
    xlab = "", ylab= "BPH (100%)", col="antiquewhite3",
    ...)
  if(getpdf == TRUE){
    pdf(outfile, width=6, height=6)
    boxplot(pBPH ~ bymed2, data=trait,
      xlab = "", ylab= "BPH (100%)", col="antiquewhite3",
      ...)
    dev.off()
  }
}

plotloh(getpdf=getpdf, outfile="graphs/Fig1b_v3.pdf",
  main="", cex.axis=fs, cex.lab=fs)
```

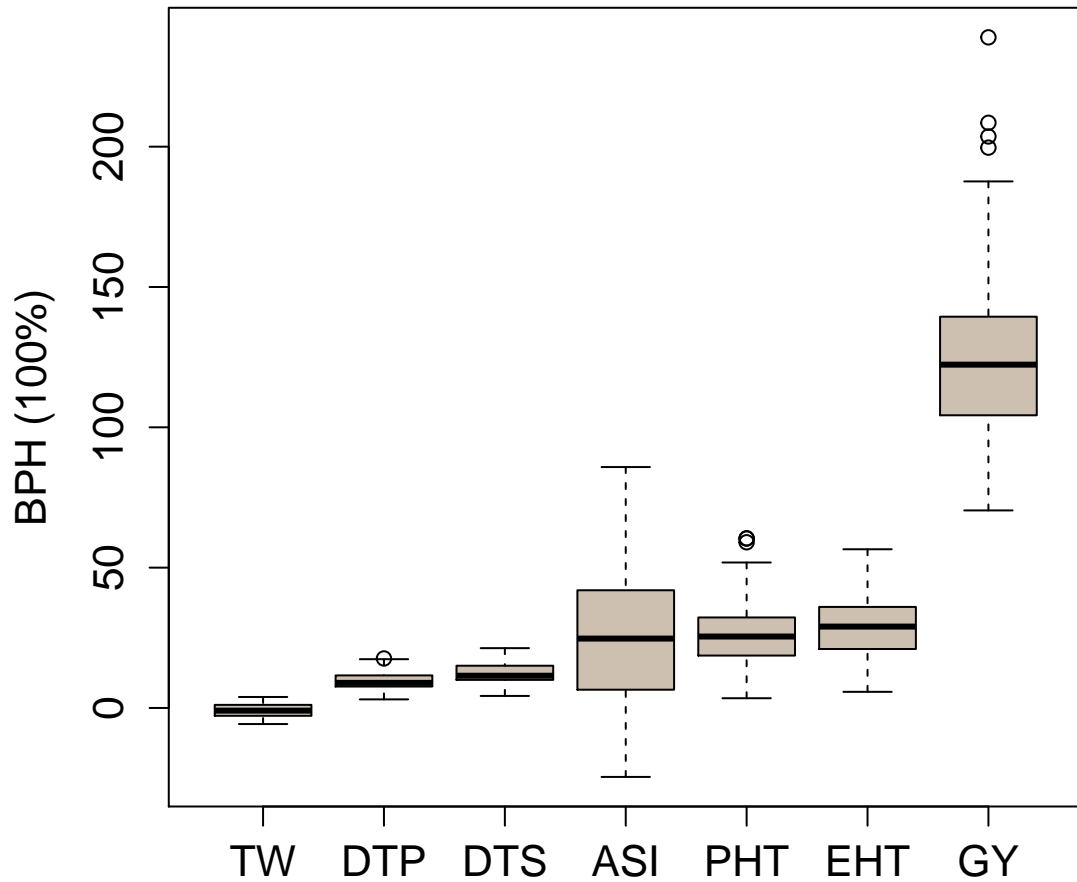


Figure 1c

```
plotReg <- function(getpdf=TRUE, outfile, ...){
  snptab <- read.csv("cache/daf_gerp2.csv")
  snptab <- snptab[order(snptab$GERP2),]
  plx <- predict(loess(snptab$meandaf ~ snptab$GERP2), se=T)
  x <- snptab$GERP2
  y <- snptab$meandaf

  plot(x, y, ...)
  lines(snptab$GERP2, plx$fit, col="cornflowerblue", lwd=2)
  lines(snptab$GERP2, plx$fit - qt(0.975,plx$df)*plx$se, lty=2, lwd=2, col="black")
  lines(snptab$GERP2, plx$fit + qt(0.975,plx$df)*plx$se, lty=2, lwd=2, col="black")

  if(getpdf == TRUE){
    pdf(outfile, width=6, height=6)
    plot(x, y, ...)
    lines(snptab$GERP2, plx$fit, col="cornflowerblue", lwd=2)
    lines(snptab$GERP2, plx$fit - qt(0.975,plx$df)*plx$se, lty=2, lwd=2, col="black")
    lines(snptab$GERP2, plx$fit + qt(0.975,plx$df)*plx$se, lty=2, lwd=2, col="black")

    dev.off()
  }
}
```

```

}

plotReg(getpdf=getpdf, outfile="graphs/Fig1c_v3.pdf",
       pch=16, col="antiquewhite3", xlab="GERP Score", ylab="Deleterious Allele Frequency",
       main="", cex.axis=fs, cex.lab=fs)

```

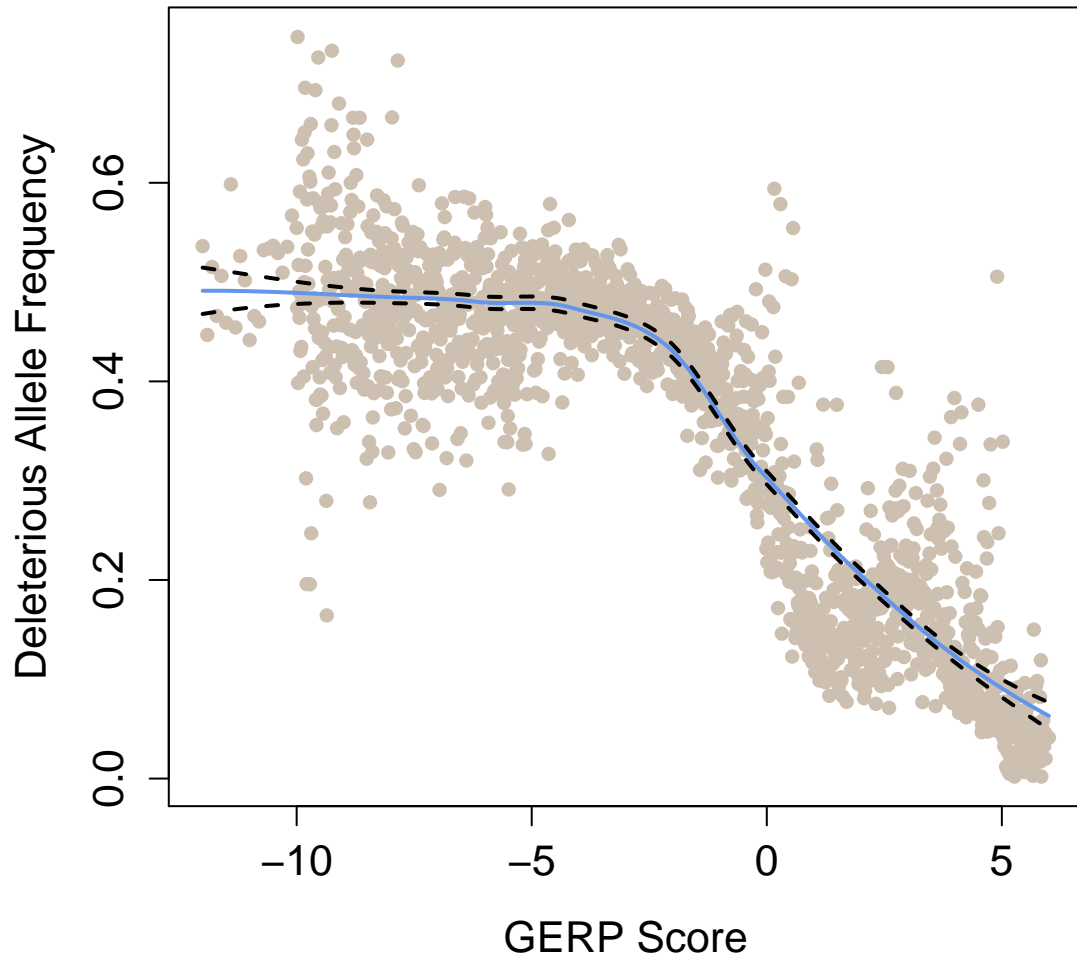


Figure 1d

To run this chunk, we need to install `beanplot` package.

```

library("beanplot")
plotbeans <- function(getpdf=TRUE, outfile, ...){

  res <- read.csv("cache/mgerp_cm.csv")
  cutoff <- quantile(res$gen, c(0.25, 0.5, 0.75))
  res$sq <- 0
  res[res$gen < cutoff[1], ]$sq <- 1
  res[res$gen >= cutoff[1] & res$gen < cutoff[2], ]$sq <- 2
  res[res$gen >= cutoff[2] & res$gen < cutoff[3], ]$sq <- 3
  res[res$gen >= cutoff[3], ]$sq <- 4
}

```

```

beanplot(mgerp ~ sq, data = res, kernel="cosine", ll = 0.04, cex=fs, side = "no", cut=10, ylim=c(0.5,
  border = NA, col=list("#cd5b45", "antiquewhite3", "antiquewhite3", "antiquewhite3"),
  xaxt="n", ...)
axis(side = 1, at = 1:4, labels = c("<25%", "25-50%", "50-75%", " >75%"), cex.axis=fs)

if(getpdf == TRUE){
  pdf(outfile, width=6, height=6)
  beanplot(mgerp ~ sq, data = res, kernel="cosine", ll = 0.04, cex=1.5, side = "no", cut=10, ylim=c(0
    border = NA, col=list("#cd5b45", "antiquewhite3", "antiquewhite3", "antiquewhite3"),
    xaxt="n", ...)
  axis(side = 1, at = 1:4, labels = c("<25%", "25-50%", "50-75%", " >75%"), cex.axis=1.3)
  dev.off()
}
}

plotbeans(getpdf=getpdf, outfile="graphs/Fig1d_v3.pdf",
  ylab="GERP Score", xlab="Quantiles of Recombination Rate (cM/Mb)",
  cex.axis=fs, cex.lab=fs)

```

log="y" selected

