

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.

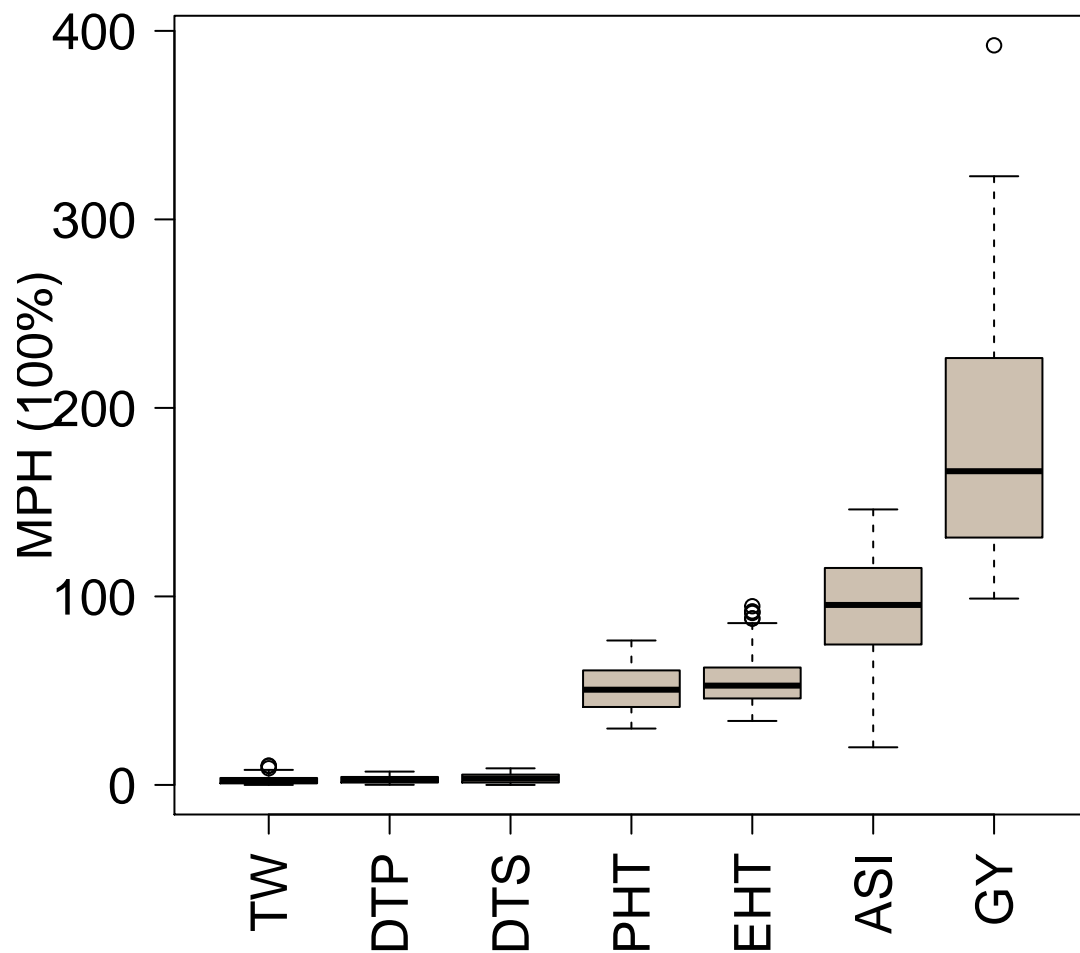
First of all, determine font size and set the `getpdf` option:

```
#par(mar=c(5,4,4,2))
par(font= 2, font.lab= 2, font.axis= 2)
fs <- 1.6 # times bigger than default
ht= 6; wt= 6 #figure height and weight
getpdf <- TRUE # get figures in separated pdf [TRUE] or not [FALSE]
```

Figure 1a

```
plotloh <- function(getpdf, outfile, ...){
  trait <- read.csv("data/hyb_heterosis.csv")
  trait$pMPH <- abs(trait$pMPH*100)
  bymed2 <- with(trait, reorder(trait, pMPH, median))
  boxplot(pMPH ~ bymed2, data=trait,
          xlab = "", ylab= "MPH (100%)", col="antiquewhite3",
          ...)
  if(getpdf == TRUE){
    pdf(outfile, width=ht, height=wt)
    par(mar=c(5,5,4,2))
    boxplot(pMPH ~ bymed2, data=trait,
            xlab = "", ylab= "MPH (100%)", col="antiquewhite3",
            ...)
    dev.off()
  }
}

plotloh(getpdf, outfile="graphs/Fig1a.pdf",
        main="", cex.axis=fs, cex.lab=fs, las=2)
```



```
## pdf
## 2
```

Figure 1b

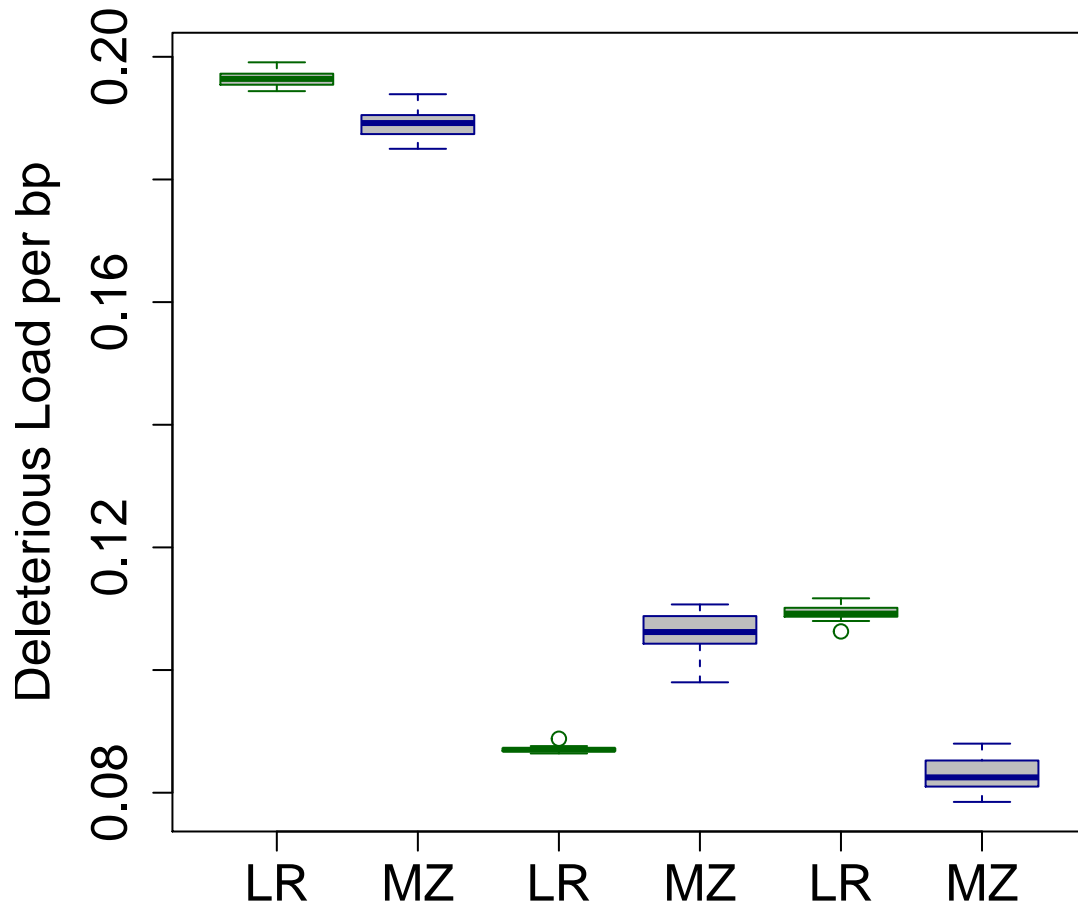
```
plot_load <- function(getpdf, outfile, ...){
  dres <- read.table("data/sup_deleterious_hmp3.txt", header=T)

  boxplot(DR ~ geno*ordered, data=dres, border=c("darkgreen","darkblue"), col="grey", ...)
  #text(cex=fs, x=-.25, y=-1.25, H2$Traits, xpd=TRUE, srt=45, pos=2)
  if(getpdf == TRUE){
    pdf(outfile, width=ht, height=wt)
    par(mar=c(5,5,4,2))
    boxplot(DR ~ geno*ordered, data=dres, border=c("darkgreen","darkblue"), col="grey", ...)

    box()
    dev.off()
  }
}

plot_load(getpdf, outfile="graphs/Fig1d_v3.pdf",
  names=c("LR", "MZ", "LR", "MZ", "LR", "MZ"),
```

```
xlab="", ylab="Deleterious Load per bp",
main="", cex.axis=fs, cex.lab=fs)
```



```
## pdf
## 2
```

Figure 1c

```
plotReg <- function(getpdf, 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=wt, height=ht)
    par(mar=c(5,5,4,2))
  }
}
```

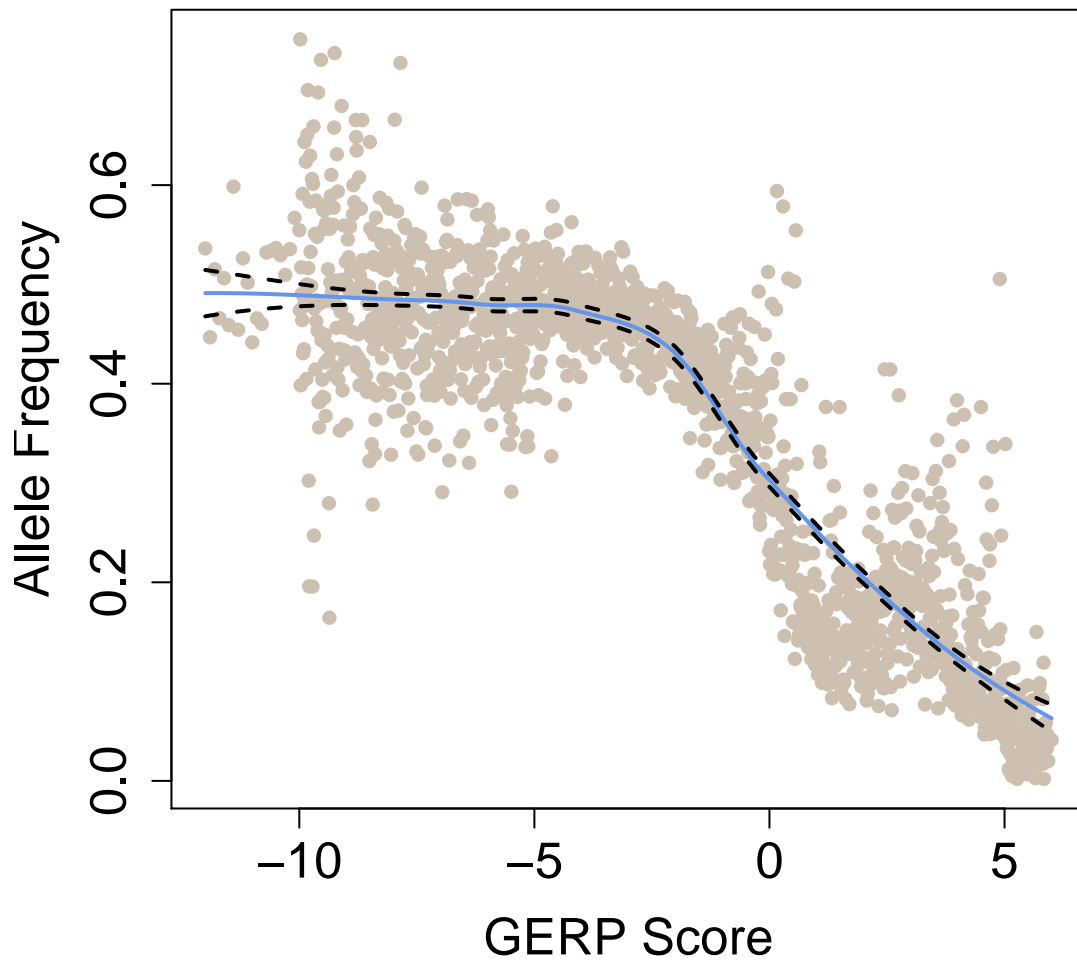
```

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, outfile="graphs/Fig1b_v3.pdf",
  pch=16, col="antiquewhite3", xlab="GERP Score", ylab="Allele Frequency",
  main="", cex.axis=fs, cex.lab=fs)

```



```

## pdf
## 2

```

Figure 1d

```

#####
plotmgerp <- function(mgerp, getpdf, outfile, ...){

```

```

#mgerp <- read.csv("cache/mgerp_cm.csv")

mgerp <- mgerp[order(mgerp$gen),]
plx <- predict(loess(mgerp$mgerp ~ mgerp$gen), se=T)
x <- mgerp$gen
y <- mgerp$mgerp

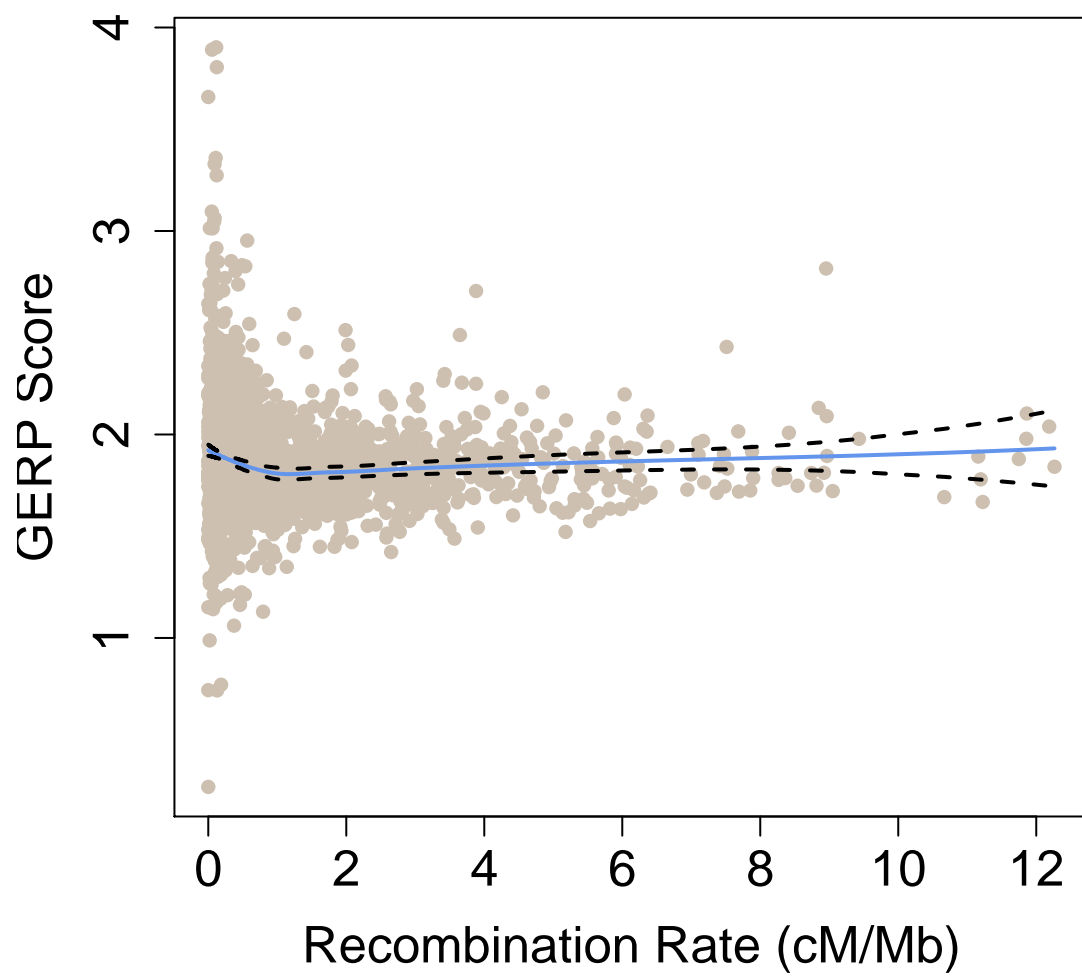
plot(x, y, ...)
lines(mgerp$gen, plx$fit, col="cornflowerblue", lwd=2)
lines(mgerp$gen, plx$fit - qt(0.975,plx$df)*plx$se, lty=2, lwd=2, col="black")
lines(mgerp$gen, plx$fit + qt(0.975,plx$df)*plx$se, lty=2, lwd=2, col="black")

if(getpdf == TRUE){
  pdf(outfile, width=wt, height=ht)
  par(mar=c(5,5,4,2))
  plot(x, y, ...)
  lines(mgerp$gen, plx$fit, col="cornflowerblue", lwd=2)
  lines(mgerp$gen, plx$fit - qt(0.975,plx$df)*plx$se, lty=2, lwd=2, col="black")
  lines(mgerp$gen, plx$fit + qt(0.975,plx$df)*plx$se, lty=2, lwd=2, col="black")

  dev.off()
}
}

mgerp <- read.csv("cache/mgerp_cm.csv")
plotmgerp(mgerp, getpdf=TRUE, outfile="graphs/Fig1d.pdf",
  pch=16, col="antiquewhite3", xlab="Recombination Rate (cM/Mb)", ylab="GERP Score",
  main="", cex.axis=fs, cex.lab=fs)

```



pdf
2