Load necessary libraries:

```
library(ggplot2)
library(ggvis)

##
## Attaching package:  'ggvis'
## The following object is masked from 'package:ggplot2':
##
##    resolution

library(dplyr)

##
## Attaching package:  'dplyr'
## The following objects are masked from 'package:stats':
##
##    filter, lag
## The following objects are masked from 'package:base':
##
##    intersect, setdiff, setequal, union

library(RColorBrewer)
```

Function of H_{adm} in terms of H_1, H_2, γ_1, and F_{12}:

```
Hadm <- function(H1, H2, gamma1, F12)
{
  gamma1^2*H1+(1-gamma1)^2*H2+
    gamma1*(1-gamma1)*(H1+H2)*(1+F12)/(1-F12)
}
```

Get some realistic values of H_1 and H_2 as means of expected heterozygosities for the European and Native American populations in the Pemberton et al 2013 *G3* paper:

```
HetsEuro <- c(0.724,0.728,0.731,0.718,0.730,0.727,0.723,0.735)
length(HetsEuro)

## [1] 8

H1 <- mean(HetsEuro)

HetsNAm <- c(0.625,0.561,0.507,0.676,0.617,
             0.671,0.695,0.696,0.665,0.644,
             0.641,0.665,0.582,0.623,0.660,
             0.667,0.649,0.483,0.624,0.669,
             0.580,
             0.643,0.640,0.671,0.590,0.587,
             0.625,0.625,0.633)
length(HetsNAm)

## [1] 29

H2 <- mean(HetsNAm)

H1

## [1] 0.727

H2

## [1] 0.628069
```

Get the region where $\underline{\gamma_{\text{max}}} = \gamma^*$ in terms of H_1 and H_2:

```
abs(H1-H2)/(2*(H1+H2)+abs(H1-H2))

## [1] 0.03521844
```

Get values of H_{adm} for $\gamma_1 \in [0, 1]$ and various values of F_{12}:

```
minFst <- (2-H1-H2-2*sqrt((1-H1)*(1-H2)))/
  (2+H1+H2-2*sqrt((1-H1)*(1-H2)))
maxFst <- (2-H1-H2)/(2+H1+H2)
FstGammaStr <- abs(H1-H2)/(2*(H1+H2+abs(H1-H2)))

HadmDF <- expand.grid(gamma1=seq(0,1,by = 0.01),
                      Fst=c(minFst,0.02,FstGammaStr,0.06,0.10,
                            0.14,0.18,maxFst))
HadmDF$Hadm <- NA
for(i in 1:nrow(HadmDF))
{
  HadmDF$Hadm[i] <- Hadm(H1 = H1, H2 = H2,
                         gamma1 = HadmDF$gamma1[i],
                         F12 = HadmDF$Fst[i])
}

HadmDF$Fst <- round(HadmDF$Fst, 3)
```

Make a 2nd data frame with just the boundary for the region where the maximum of H_{adm} is at γ^*:

```
HadmDFB1 <- HadmDF[HadmDF$Fst == 0.034,]
HadmDFB2 <- HadmDF[HadmDF$Fst == 0.192,]
identical(HadmDFB1[,1], HadmDFB2[,1])

## [1] TRUE

HadmDF2 <- data.frame(gamma1 = HadmDFB1$gamma1,
                      HadmB1 = HadmDFB1$Hadm,
                      HadmB2 = HadmDFB2$Hadm)
```

Add some more things to data frame to work with ggplot:

```
HadmDF$Fst <- as.factor(HadmDF$Fst)

HadmDF <- HadmDF %>%
  group_by(Fst) %>%
  mutate(HadmMax = (max(Hadm) == Hadm))
```

Specify colors:

```
##the default colors are just equally spaced colors on the color wheel
##from: http://stackoverflow.com/questions/8197559/emulate-ggplot2-default-color-palette
gg_color_hue <- function(n) {
  hues = seq(15, 375, length=n+1)
  hcl(h=hues, l=65, c=100)[1:n]
}

grey <- grey.colors(7)[3]

##ggCols5 <- gg_color_hue(5)
ggCols5 <- brewer.pal(7, "BuPu")[-(1:2)]
```

Get the values of Hadm over all values of gamma1, Fst in a tight grid:

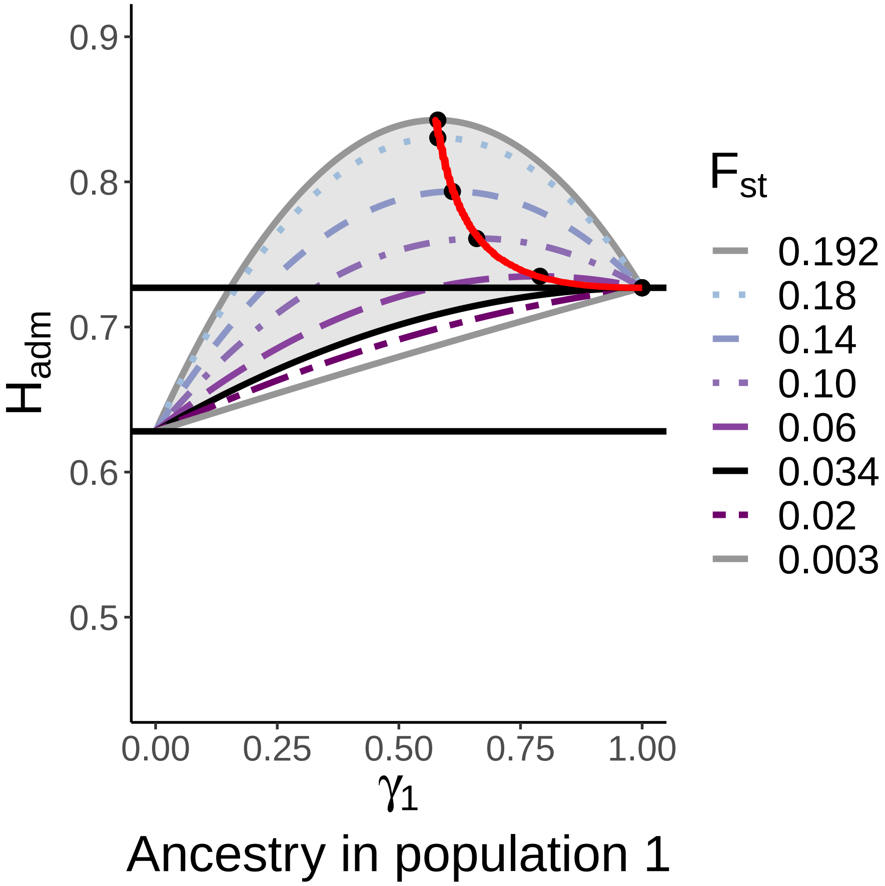
```
Hadm_all_allow <-
  expand.grid(gamma1=seq(0,1,by = 0.005),
              Fst=seq(minFst,maxFst,length.out=200))
Hadm_all_allow$Hadm <- NA
for(i in 1:nrow(Hadm_all_allow))
{
  Hadm_all_allow$Hadm[i] <-
    Hadm(H1 = H1, H2 = H2,
         gamma1 = Hadm_all_allow$gamma1[i],
         F12 = Hadm_all_allow$Fst[i])
}
head(Hadm_all_allow)
```

```
##   gamma1         Fst      Hadm
## 1  0.000 0.002808579 0.6280690
## 2  0.005 0.002808579 0.6286016
## 3  0.010 0.002808579 0.6291338
## 4  0.015 0.002808579 0.6296657
## 5  0.020 0.002808579 0.6301972
## 6  0.025 0.002808579 0.6307283

tail(Hadm_all_allow)

##       gamma1       Fst      Hadm
## 40195  0.975 0.1922259 0.7402469
## 40196  0.980 0.1922259 0.7376620
## 40197  0.985 0.1922259 0.7350449
## 40198  0.990 0.1922259 0.7323955
## 40199  0.995 0.1922259 0.7297139
## 40200  1.000 0.1922259 0.7270000

##now for each value of Fst, get the maximum value
Hadm_max <- data.frame(Fst=seq(minFst,maxFst,length.out=200),
                       gamma1_argmax=NA,
                       Hadm_max=NA)
for(i in 1:nrow(Hadm_max))
{
  ##get all rows of Hadm_all_allow with this value of Fst
  current_Fst <- Hadm_max$Fst[i]
  subset_Fst <-
    Hadm_all_allow[Hadm_all_allow$Fst == current_Fst,]
  ##get argmax and max
  current_argmax <- which.max(subset_Fst$Hadm)
  Hadm_max$gamma1_argmax[i] <- subset_Fst$gamma1[current_argmax]
  Hadm_max$Hadm_max[i] <- subset_Fst$Hadm[current_argmax]
}
```

Make some plots:

```
head(HadmDF$Fst)

## [1] 0.003 0.003 0.003 0.003 0.003 0.003
## Levels: 0.003 0.02 0.034 0.06 0.1 0.14 0.18 0.192

##rewrite this as a function of gamma1 primarily
Hadm <- function(gamma1, H1, H2, F12)
{
  gamma1^2*H1+(1-gamma1)^2*H2+
    gamma1*(1-gamma1)*(H1+H2)*(1+F12)/(1-F12)
}

HadmDF$Fst <- factor(HadmDF$Fst, levels=rev(levels(HadmDF$Fst)))

ggplot(HadmDF) +
  geom_ribbon(data=HadmDF2, aes(x=gamma1, ymin=HadmB1, ymax=HadmB2),
              fill="grey", alpha=0.4) +
  geom_line(aes(x=gamma1, y=Hadm, group=Fst, col=Fst,
                lty=Fst), size=1.2) +
  geom_point(aes(x=gamma1, y=Hadm, size=HadmMax)) +
  geom_hline(yintercept=c(H1,H2), size=1.2) +
  geom_line(data=Hadm_max,
            aes(x=gamma1_argmax, y=Hadm_max),
            color="red", size=1.2) +
  ylim(c(0.45,0.9)) +
  scale_color_manual(values=c(grey, ggCols5[1:4],
                              "#000000",
                              ggCols5[5], grey),
                     guide_legend(title=quote(Fst)),
                     labels = c("0.192","0.18","0.14","0.10",
                                "0.06","0.034","0.02","0.003")) +
  scale_size_manual(values = c(NA, 2.5)) +
  scale_linetype_manual(values = c(1,3,2,4,5,1,6,1),
                        guide_legend(title=quote(Fst)),
                        labels = c("0.192","0.18","0.14","0.10",
                                   "0.06","0.034","0.02","0.003")) +
  theme(axis.line = element_line(colour = "black"),
        plot.title = element_text(size = 16, hjust = 0.5),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank(),
        legend.key = element_blank(),
        legend.text.align = 0,
        text = element_text(size=20),
        axis.title.x = element_text(size=20),
        axis.title.y = element_text(size=20),
        axis.text.x = element_text(size=14),
        axis.text.y = element_text(size=14)) +
  ylab(expression(H[adm])) +
  ##xlab(expression(gamma[1])) +
  xlab(expression(atop(gamma[1], "Ancestry in population 1"))) +
  guides(size=FALSE,
         color=guide_legend(title=expression(F[st])),
         linetype=guide_legend(title=expression(F[st])))
```

```
## Warning:  Removed 800 rows containing missing values (geom_point).
```


H_{adm}
γ_1
Ancestry in population 1
F_{st}