

Generate plot in Figure 4

Simina Boca, Jeff Leek

2017-06-13

Load the relevant libraries:

```
library(ggplot2)
library(reshape2)
library(dplyr)
library(scales)
library(qvalue)
library(xtable)
```

Load the data and the results and add them to the data object:

```
load("BMI_GIANT_GWAS.RData")

tot <- BMI_GIANT_GWAS

load("BMI_GIANT_GWAS_results_logistic.RData")

tot$fitted0.8 <- fitted0.8
tot$fitted0.9 <- fitted0.9
tot$fitted.final.smooth <- fitted.final.smooth

load("BMI_GIANT_GWAS_results_Scott_theoretical.RData")

tot$ScottT <- pi0EstScott

load("BMI_GIANT_GWAS_results_Scott.RData")

tot$ScottE <- pi0EstScott

load("BMI_GIANT_GWAS_bootstrap_CIs_logistic.RData")

dim(tot)

## [1] 2500573      14
dim(perc.05.95)

## [1] 2500573      2
head(tot)

## # A tibble: 6 × 14
##       SNP     A1     A2 Freq_MAF_Hapmap      b      se      p      N
##   <chr> <chr> <chr>        <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 rs1000000 G     A        0.3667 -0.0001  0.0043  0.98140 233572
## 2 rs10000010 T     C        0.4250  0.0022  0.0029  0.43840 339148
## 3 rs10000012 G     C        0.1917 -0.0096  0.0053  0.07009 236095
## 4 rs10000013 A     C        0.1667  0.0096  0.0043  0.02558 236048
## 5 rs10000017 C     T        0.2333  0.0038  0.0045  0.39840 235308
## 6 rs10000023 G     T        0.4083  0.0023  0.0037  0.53420 236022
```

```

## # ... with 6 more variables: Freq_MAF_Int_Hapmap <fctr>, fitted0.8 <dbl>,
## #   fitted0.9 <dbl>, fitted.final.smooth <dbl>, ScottT <dbl>, ScottE <dbl>
head(perc.05.95)

##      [,1]      [,2]
## 1 0.8946947 0.9062979
## 2 0.6841069 0.7169933
## 3 0.8973993 0.9056517
## 4 0.8976166 0.9070543
## 5 0.9038571 0.9139008
## 6 0.8667469 0.8766078

head(rownames(perc.05.95))

## [1] "1" "2" "3" "4" "5" "6"
identical(as.integer(rownames(perc.05.95)), 1:nrow(perc.05.95))

## [1] TRUE
tot$lower.CI <- perc.05.95[,1]
tot$upper.CI <- perc.05.95[,2]

```

Get range of estimates:

```

range(tot$fitted.final.smooth)

## [1] 0.6966151 1.0000000

range(tot$fitted0.8)

## [1] 0.7042345 1.0000000

range(tot$fitted0.9)

## [1] 0.6890391 1.0000000

```

Only use random sample of 50,000 SNPs in plot:

```

##get random sample of 50,000 SNPs
set.seed(310841)
d1 = tot %>% sample_n(5e4)

```

Create long data frame (for ggplot):

```

colnames(d1)[colnames(d1)=="ScottT"] <- "fitted.ScottT"
colnames(d1)[colnames(d1)=="ScottE"] <- "fitted.ScottE"

d3 <- melt(d1,
            id.vars=colnames(d1)[-grep("fitted", colnames(d1))],
            value.name = "pi0", variable.name = "lambda")
d3$lambda <- as.character(d3$lambda)
d3$lambda[d3$lambda=="fitted0.8"] <- "lambda=0.8"
d3$lambda[d3$lambda=="fitted0.9"] <- "lambda=0.9"
d3$lambda[d3$lambda=="fitted.final.smooth"] <- "final smoothed pi0(x)"
d3$lambda[d3$lambda=="fitted.ScottT"] <- "Scott theoretical"
d3$lambda[d3$lambda=="fitted.ScottE"] <- "Scott empirical"

d3$Freq_MAF_Int_Hapmap <- paste("MAF in", d3$Freq_MAF_Int_Hapmap)

```

```
head(d3)
```

```
##          SNP A1 A2 Freq_MAF_Hapmap      b     se      p      N
## 1 rs10510371 T  C       0.0250  0.0147 0.0152 0.3335 212965
## 2 rs918232   A  G       0.3417 -0.0034 0.0037 0.3581 236084
## 3 rs4816764   A  C       0.0083  0.0163 0.0131 0.2134 221771
## 4 rs17630047   A  G       0.1667  0.0004 0.0048 0.9336 236177
## 5 rs4609437   C  G       0.2500  0.0011 0.0042 0.7934 236028
## 6 rs11130746   G  A       0.2333 -0.0006 0.0042 0.8864 235634
##          Freq_MAF_Int_Hapmap lower.CI upper.CI lambda      pi0
## 1 MAF in [0.000,0.127) 1.0000000 1.0000000 lambda=0.8 1.0000000
## 2 MAF in [0.302,0.500] 0.8662131 0.8766112 lambda=0.8 0.8838673
## 3 MAF in [0.000,0.127) 1.0000000 1.0000000 lambda=0.8 1.0000000
## 4 MAF in [0.127,0.302) 0.8976483 0.9073273 lambda=0.8 0.9101864
## 5 MAF in [0.127,0.302) 0.8975774 0.9070475 lambda=0.8 0.9101274
## 6 MAF in [0.127,0.302) 0.8999862 0.9096028 lambda=0.8 0.9118310
```

Function to change the way labels are displayed:

```
display_thousands <- function(l) {
  ##divide by 1000
  l1 <- l/1000
  ##add k afterwards
  l11 <- paste(l1, "k", sep="")

  l11
}
```

Generate Figure 4:

```
ggplot(d3, aes(x=N, y=pi0))+
  geom_vline(xintercept=median(tot$N), colour="grey50")+
  geom_ribbon(aes(ymin=lower.CI, ymax=upper.CI, group=Freq_MAF_Int_Hapmap), fill="grey50")+
  geom_line(aes(linetype=lambda)) +
  facet_grid(. ~ Freq_MAF_Int_Hapmap) +
  scale_x_continuous(labels=comma) +
  ylab("Estimated proportion of nulls") +
  scale_linetype_discrete(labels =
    c(expression(paste("Final smoothed ",
                      hat(pi)[0](x))),
      expression(paste(hat(pi)[0], "^lambda",
                      (x),
                      " for ", lambda, "=",
                      0.8)),
      expression(paste(hat(pi)[0], "^lambda,
                      (x),
                      " for ", lambda, "=",
                      0.9)),
      "Scott empirical",
      "Scott theoretical")) +
  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(),
```

```

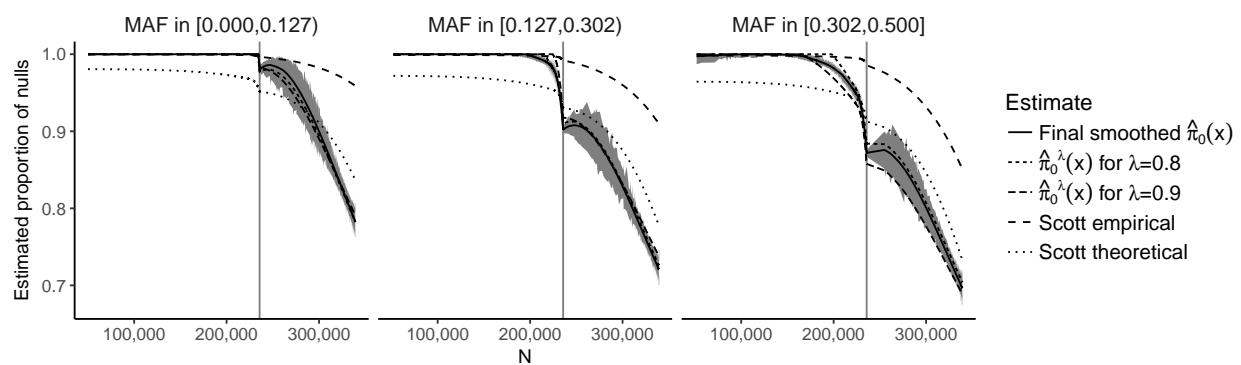
panel.background = element_blank(),
legend.key = element_blank(),
legend.text.align = 0,
legend.title=element_text(size=13),
legend.text=element_text(size=12),
strip.background = element_blank(),
strip.text.x = element_text(size = 12),
axis.text.x = element_text(size=10),
axis.text.y = element_text(size=10))+  

##axis.line.x = element_line(color="black", size = 0.5), ##this is to show axes - bug in this v  

##axis.line.y = element_line(color="black", size = 0.5)) +  

guides(linetype=guide_legend(title="Estimate"))

```



Get FDR for our method:

```

##first get BH q-values
qValsBH <- p.adjust(tot$p, method="BH")
##also get Storey q-values
qValsStorey <- qvalue(tot$p)$qvalues
##what is their difference
head(qValsStorey/qValsBH)

## [1] 0.9486617 0.9486617 0.9486617 0.9486617 0.9486617 0.9486617

qValsReg <- qValsBH*tot$fitted.smooth

tot$qValsBH <- qValsBH
tot$qValsStorey <- qValsStorey
tot$qValsReg <- qValsReg

sum(tot$qValsBH < 0.05)

## [1] 12500
sum(tot$qValsStorey < 0.05)

## [1] 12771
sum(tot$qValsReg < 0.05)

## [1] 13384
sum(FDRScott_emp < 0.05)

## [1] 7636

```

```

sum(FDRScott_theo < 0.05)

## [1] 16697

mean(tot$qValsBH < 0.05)

## [1] 0.004998854

mean(tot$qValsStorey < 0.05)

## [1] 0.005107229

mean(tot$qValsReg < 0.05)

## [1] 0.005352373

mean(FDRScott_emp < 0.05)

## [1] 0.0030537

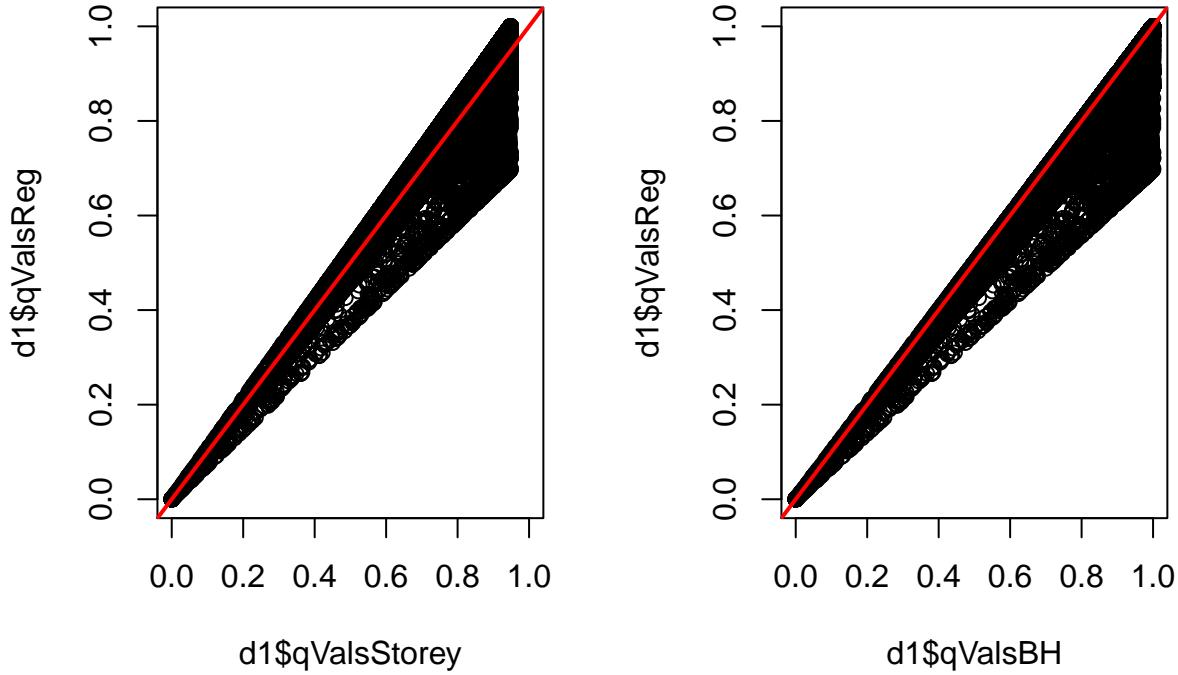
mean(FDRScott_theo < 0.05)

## [1] 0.00667727

set.seed(310841)
d1 = tot %>% sample_n(5e4)

par(mfrow=c(1,2))
plot(d1$qValsReg ~ d1$qValsStorey, xlim=c(0,1), ylim=c(0,1))
abline(a=0,b=1, col="red", lwd=2)
plot(d1$qValsReg ~ d1$qValsBH, xlim=c(0,1), ylim=c(0,1))
abline(a=0,b=1, col="red", lwd=2)

```



Make nice table for summary results:

```
xtable(data.frame(BL = sum(tot$qValsReg <= 0.05),
                  Scott_T = sum(FDRScott_theo <= 0.05),
                  Scott_E = sum(FDRScott_emp <= 0.05),
                  Storey = sum(tot$qValsStorey <= 0.05),
                  BH = sum(tot$qValsBH <= 0.05)))

## % latex table generated in R 3.4.0 by xtable 1.8-2 package
## % Tue Jun 13 16:00:01 2017
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrrr}
##   \hline
##   & BL & Scott\_T & Scott\_E & Storey & BH \\
##   \hline
##   1 & 13384 & 16697 & 7636 & 12771 & 12500 \\
##   \hline
## \end{tabular}
## \end{table}
```

How many are in common between all the methods?

```
length(Reduce(intersect, list(which(tot$qValsReg <= 0.05),
                           which(FDRScott_theo <= 0.05),
                           which(FDRScott_emp <= 0.05),
                           which(tot$qValsStorey <= 0.05),
                           which(tot$qValsBH <= 0.05))))
```

```
[1] 6892
length(Reduce(intersect, list(which(tot$qValsReg <= 0.05),
                                which(tot$qValsStorey <= 0.05),
                                which(tot$qValsBH <= 0.05))))
```

```
[1] 12500
length(Reduce(intersect, list(which(tot$qValsStorey <= 0.05),
                                which(tot$qValsReg <= 0.05))))
```

```
[1] 12740
length(Reduce(intersect, list(which(FDRScott_theo <= 0.05),
                                which(tot$qValsReg <= 0.05))))
```

```
[1] 13119
length(Reduce(intersect, list(which(FDRScott_theo <= 0.05),
                                which(FDRScott_emp <= 0.05),
                                which(tot$qValsBH <= 0.05))))
```

```
[1] 6892
length(Reduce(intersect, list(which(FDRScott_emp <= 0.05),
                                which(tot$qValsBH <= 0.05))))
```

```
[1] 6892
length(Reduce(intersect, list(which(FDRScott_theo <= 0.05),
                                which(tot$qValsBH <= 0.05))))
```

```
[1] 12251
```

Session Information

```
devtools::session_info()

## Session info -----
## setting  value
## version  R version 3.4.0 (2017-04-21)
## system   x86_64, mingw32
## ui        RTerm
## language (EN)
## collate  English_United States.1252
## tz       America/New_York
## date     2017-06-13

## Packages -----
## package    * version date      source
## assertthat  0.2.0   2017-04-11 CRAN (R 3.4.0)
## backports   1.0.5   2017-01-18 CRAN (R 3.4.0)
## BiocStyle   * 2.4.0   2017-04-25 Bioconductor
## colorspace  1.3-2   2016-12-14 CRAN (R 3.4.0)
## DBI        0.6-1   2017-04-01 CRAN (R 3.4.0)
## devtools    1.12.0  2016-12-05 CRAN (R 3.4.0)
## digest     0.6.12  2017-01-27 CRAN (R 3.4.0)
```

```
## dplyr      * 0.5.0  2016-06-24 CRAN (R 3.4.0)
## evaluate    0.10   2016-10-11 CRAN (R 3.4.0)
## ggplot2     * 2.2.1  2016-12-30 CRAN (R 3.4.0)
## gridExtra    0.2.0  2016-02-26 CRAN (R 3.4.0)
## htmtools     0.3.5  2016-03-21 CRAN (R 3.4.0)
## knitr       1.15.1 2016-11-22 CRAN (R 3.4.0)
## lazyeval     0.2.0  2016-06-12 CRAN (R 3.4.0)
## magrittr     1.5    2014-11-22 CRAN (R 3.4.0)
## memoise      1.1.0  2017-04-21 CRAN (R 3.4.0)
## munsell      0.4.3  2016-02-13 CRAN (R 3.4.0)
## plyr        1.8.4  2016-06-08 CRAN (R 3.4.0)
## qvalue      * 2.8.0  2017-04-25 Bioconductor
## R6          2.2.0  2016-10-05 CRAN (R 3.4.0)
## Rcpp        0.12.10 2017-03-19 CRAN (R 3.4.0)
## reshape2     * 1.4.2  2016-10-22 CRAN (R 3.4.0)
## rmarkdown    1.5    2017-04-26 CRAN (R 3.4.0)
## rprojroot    1.2    2017-01-16 CRAN (R 3.4.0)
## scales      * 0.4.1  2016-11-09 CRAN (R 3.4.0)
## stringi      1.1.5  2017-04-07 CRAN (R 3.4.0)
## stringr      1.2.0  2017-02-18 CRAN (R 3.4.0)
## tibble       1.3.0  2017-04-01 CRAN (R 3.4.0)
## withr        1.0.2  2016-06-20 CRAN (R 3.4.0)
## xtable      * 1.8-2  2016-02-05 CRAN (R 3.4.0)
## yaml        2.1.14 2016-11-12 CRAN (R 3.4.0)
```