Get bootstrap CIs for our method

Simina Boca, Jeff Leek

2017-06-20

Load the relevant libraries:

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.3
```

```
library(reshape2)
library(dplyr)
##library(swfdr)
library(devtools)
```

```
## Warning: package 'devtools' was built under R version 3.3.3
```

```
library(doParallel) ##to make cluster (on Windows)
library(foreach) ##to use foreach function that does the parallel processing
library(doRNG) ##for reproducible seeds when doing parallel processing
```

```
## Warning: package 'rngtools' was built under R version 3.3.2
```

```
## Warning: package 'pkgmaker' was built under R version 3.3.2
```

```
## Warning: package 'registry' was built under R version 3.3.2
```

```
source("../functions.R")
```

Load the data and the results:

```
load("BMI_GIANT_GWAS.RData")
load("BMI_GIANT_GWAS_results_logistic.RData")
load("BMI_GIANT_GWAS_results_Scott.RData")

tot <- BMI_GIANT_GWAS
```

Add the results to the data object:

```
##add the estimates from Scott
tot$fitted.Scott <- pi0EstScott

##add the estimates from our approach
##add the estimates for lambda=0.8, lambda=0.9, final smoothed value
tot$fitted0.8 <- fitted0.8
tot$fitted0.9 <- fitted0.9
tot$fitted.final.smooth <- fitted.final.smooth
```

Get range of estimates:

```
range(tot$fitted.final.smooth)
```

```
## [1] 0.6874214 1.0000000
```

```
range(tot$fitted0.8)
```

```
## [1] 0.6979177 1.0000000
```

```
range(tot$fitted0.9)
```

```
## [1] 0.6782152 1.0000000
```

Do bootstrapping:

```
nrSNPs <- nrow(tot)
X <- model.matrix(~ splines::ns(N,5) + Freq_MAF_Int_Hapmap, data = tot)[,-1]
dim(X)
```

```
## [1] 2500573       7
```

```
head(X)
```

```
##   splines::ns(N, 5)1 splines::ns(N, 5)2 splines::ns(N, 5)3
## 1       4.414107e-01       5.538398e-01      -0.0017421409
## 2       0.000000e+00       3.954615e-10      -0.1655612193
## 3       3.884106e-05       9.880678e-01       0.0099138318
## 4       3.209714e-04       9.891039e-01       0.0088153851
## 5       9.327150e-02       9.061998e-01       0.0002901038
## 6       6.724476e-04       9.894590e-01       0.0082264435
##   splines::ns(N, 5)4 splines::ns(N, 5)5 Freq_MAF_Int_Hapmap[0.127,0.302)
## 1       0.0034871548      -0.0017450139                                0
## 2       0.3336072837       0.8319539352                                0
## 3       0.0039611703      -0.0019816905                                1
## 4       0.0035221381      -0.0017623939                                1
## 5       0.0002264943      -0.0001133405                                1
## 6       0.0032867973      -0.0016447160                                0
##   Freq_MAF_Int_Hapmap[0.302,0.500]
## 1                                1
## 2                                1
## 3                                0
## 4                                0
## 5                                0
## 6                                1
```

```
##get duplicated rows in X
##duplX <- which(duplicated(X)) -> don't think I care about this actually

##the actual bootstrapping takes about 1.2 hrs for 10 iterations on desktop
cl<-makeCluster(8) ##specify number of cores less than or equal to number of cores on your computer
registerDoParallel(cl)

set.seed(310840)
nBoot <- 100
##save results
date()
```

```
## [1] "Tue Jun 20 15:30:16 2017"
```

```
fittedBoot <- foreach(boot=1:nBoot, .combine="rbind") %dorng% {
  indBoot <- sample.int(nrSNPs, size=nrSNPs, replace=TRUE);
  totboot <- tot[indBoot,];
  Xboot <- X[indBoot,];
  pi0boot <- lm_pi0(pValues=totboot$p, X=Xboot, smooth.df=3)$pi0;
  ##save only the results for the unique indices
  ##uniqueXboot <- !duplicated(Xboot)
  uniqueBoot <- unique(indBoot);
  cbind(indBoot[uniqueBoot],pi0boot[uniqueBoot])
}

date()
```

```
## [1] "Wed Jun 21 12:43:51 2017"
```

```
dim(fittedBoot)
```

```
## [1] 158057618         2
```

```
##close the cluster
stopCluster(cl)

save(list="fittedBoot", file="BMI_GIANT_GWAS_bootstrap_all_logistic.RData")

##get 0.05, 0.95 quantiles for each index ##took about 8 minutes for full dataset being bootstrapped 8 t
date()
```

```
## [1] "Wed Jun 21 12:47:28 2017"
```

```
perc.05.95 <- tapply(fittedBoot[,2], fittedBoot[,1], FUN=quantile, c(0.05, 0.95))
date()
```

```
## [1] "Wed Jun 21 12:56:45 2017"
```

```
head(perc.05.95)
```

```
## $`1`
##        5%       95%
## 0.8946947 0.9062979
##
## $`2`
##        5%       95%
## 0.6841069 0.7169933
##
## $`3`
##        5%       95%
## 0.8973993 0.9056517
##
## $`4`
##        5%       95%
## 0.8976166 0.9070543
##
## $`5`
##        5%       95%
## 0.9038571 0.9139008
##
## $`6`
##        5%       95%
## 0.8667469 0.8766078
```

```
##get the actual indices
indPerc.05.95 <- names(perc.05.95)
head(indPerc.05.95)
```

```
## [1] "1" "2" "3" "4" "5" "6"
```

```
class(indPerc.05.95)
```

```
## [1] "character"
```

```
indPerc.05.95 <- as.numeric(indPerc.05.95)

perc.05.95 <- matrix(unlist(perc.05.95), ncol=2, byrow=TRUE)

dim(perc.05.95)
```

```
## [1] 2500573       2
```

```
quant0.05 <- perc.05.95[,1]
quant0.95 <- perc.05.95[,2]

rownames(perc.05.95) <- as.character(indPerc.05.95)

##save these
save(list=c("perc.05.95"), file="BMI_GIANT_GWAS_bootstrap_CIs_logistic.RData")
```

Session Information

```
devtools::session_info()
```

```
## Session info -------------------------------------------------------------
```

```
##  setting  value
##  version  R version 3.3.1 (2016-06-21)
##  system   x86_64, mingw32
##  ui       RTerm
##  language (EN)
##  collate  English_United States.1252
##  tz       America/New_York
##  date     2017-06-21
```

```
## Packages -----------------------------------------------------------------
```

```
##  package    * version date       source
##  assertthat   0.1     2013-12-06 CRAN (R 3.3.1)
##  backports    1.0.4   2016-10-24 CRAN (R 3.3.1)
##  BiocStyle  * 2.0.3   2016-08-04 Bioconductor
##  codetools    0.2-14  2015-07-15 CRAN (R 3.3.1)
##  colorspace   1.2-6   2015-03-11 CRAN (R 3.3.1)
##  DBI          0.4-1   2016-05-08 CRAN (R 3.3.1)
##  devtools   * 1.12.0  2016-06-24 CRAN (R 3.3.3)
##  digest       0.6.9   2016-01-08 CRAN (R 3.3.1)
##  doParallel * 1.0.10  2015-10-14 CRAN (R 3.3.1)
##  doRNG      * 1.6     2014-03-07 CRAN (R 3.3.1)
##  dplyr      * 0.4.3   2015-09-01 CRAN (R 3.3.1)
##  evaluate     0.10    2016-10-11 CRAN (R 3.3.1)
##  foreach    * 1.4.3   2015-10-13 CRAN (R 3.3.1)
##  ggplot2    * 2.2.1   2016-12-30 CRAN (R 3.3.3)
##  gtable       0.2.0   2016-02-26 CRAN (R 3.3.1)
##  htmltools    0.3.5   2016-03-21 CRAN (R 3.3.1)
##  iterators  * 1.0.8   2015-10-13 CRAN (R 3.3.0)
##  knitr        1.15.1  2016-11-22 CRAN (R 3.3.1)
##  lazyeval     0.2.0   2016-06-12 CRAN (R 3.3.1)
##  magrittr     1.5     2014-11-22 CRAN (R 3.3.1)
##  memoise      1.0.0   2016-01-29 CRAN (R 3.3.1)
##  munsell      0.4.3   2016-02-13 CRAN (R 3.3.1)
##  pkgmaker   * 0.22    2014-05-14 CRAN (R 3.3.2)
##  plyr         1.8.4   2016-06-08 CRAN (R 3.3.1)
##  R6           2.1.2   2016-01-26 CRAN (R 3.3.1)
##  Rcpp         0.12.10 2017-03-19 CRAN (R 3.3.3)
##  registry   * 0.3     2015-07-08 CRAN (R 3.3.2)
##  reshape2   * 1.4.1   2014-12-06 CRAN (R 3.3.1)
##  rmarkdown    1.2     2016-11-21 CRAN (R 3.3.1)
##  rngtools   * 1.2.4   2014-03-06 CRAN (R 3.3.2)
##  rprojroot    1.1     2016-10-29 CRAN (R 3.3.1)
##  scales       0.4.1   2016-11-09 CRAN (R 3.3.3)
##  stringi      1.1.1   2016-05-27 CRAN (R 3.3.0)
##  stringr      1.0.0   2015-04-30 CRAN (R 3.3.1)
##  tibble       1.2     2016-08-26 CRAN (R 3.3.2)
##  withr        1.0.2   2016-06-20 CRAN (R 3.3.1)
##  xtable       1.8-2   2016-02-05 CRAN (R 3.3.1)
##  yaml         2.1.13  2014-06-12 CRAN (R 3.3.1)
```