```
## Load libraries
library(splines)
library(MASS)
library(FDRreg)

## Loading required package:  fda
## Warning:  package 'fda' was built under R version 3.3.3
## Loading required package:  Matrix
##
## Attaching package:  'fda'
## The following object is masked from 'package:graphics':
##
##    matplot
## Loading required package:  BayesLogit
## Warning:  package 'BayesLogit' was built under R version 3.3.2
## Loading required package:  mvtnorm
## Warning:  package 'mvtnorm' was built under R version 3.3.2

library(curl)

library(doParallel) ##to make cluster (on Windows)

## Loading required package:  foreach
## Loading required package:  iterators
## Loading required package:  parallel

library(foreach) ##to use foreach function that does the parallel processing
library(doRNG) ##for reproducible seeds when doing parallel processing

## Loading required package:  rngtools
## Warning:  package 'rngtools' was built under R version 3.3.2
## Loading required package:  pkgmaker
## Warning:  package 'pkgmaker' was built under R version 3.3.2
## Loading required package:  registry
## Warning:  package 'registry' was built under R version 3.3.2
##
## Attaching package:  'pkgmaker'
## The following object is masked from 'package:base':
##
##    isNamespaceLoaded

##Source functions
source("../functions.R")

options(warn=1)
```

Define nulltype for Scott method:

```
nulltype <- "empirical"
```

Simulations are performed for a variety of alternative distributions:

```
alts <- c("alt_beta","alt_chisq_large_3_3","alt_chisq_large",
          "alt_chisq_small_3_3","alt_chisq_small",
          "alt_t_large","alt_t_small",
          "alt_z_large",
          "alt_z_small")
```

1 Probability of being a false positive is flat

Perform estimation and save estimates:

```
for(alt in alts)
{
  print(alt)

  load(paste(alt,"simResults_1.RData",sep="/"))
  ntest <- ncol(zValuesSims)

  pi0hatScottMat_emp <- estimate_Scott_sims(zValuesSims, tme, nulltype)

  pi0hatScottMean_emp <- colMeans(pi0hatScottMat_emp[,1:ntest])
  pi0hatScottVar_emp <- apply(pi0hatScottMat_emp[,1:ntest],2,var)

  pi0hat.ScottMat_emp <- pi0hatScottMat_emp[,1:ntest]
  FDR.ScottMat_emp <- pi0hatScottMat_emp[,(ntest+1):(2*ntest)]

  ##save full results
  save(file=paste(alt,"simResults_pi0x_Scott_emp_1_full.RData",sep="/"),
       list=c("pi0hat.ScottMat_emp", "FDR.ScottMat_emp"))

  ##save summary results
  save(file=paste(alt,"simResults_pi0x_Scott_emp_1.RData",sep="/"),
       list=c("tme", "pi0",
            "pi0hatScottMean_emp","pi0hatScottVar_emp"))
}
```

```
## [1] "alt_beta"
## [1] "alt_chisq_large_3_3"
```

```
## Warning in apply(as.matrix(pi0hatScottMat), 2, as.numeric):  NAs
introduced by coercion
```

```
## Error in apply(as.matrix(pi0hatScottMat), 2, as.numeric):  (list)
object cannot be coerced to type 'double'
```

2 Probability of being a false positive is smooth in one variable

Perform estimation and save estimates:

```
for(alt in alts)
{
  print(alt)

  load(paste(alt,"simResults_2.RData",sep="/"))
  ntest <- ncol(zValuesSims)

  ##--------linear fit-------##
  print("linear")
  pi0hatScottMat_empFitLin <- estimate_Scott_sims(zValuesSims, tme, nulltype)

  pi0hatLin.ScottMean_emp <- colMeans(pi0hatScottMat_empFitLin[,1:ntest])
  pi0hatLin.ScottVar_emp <- apply(pi0hatScottMat_empFitLin[,1:ntest],2,var)

  pi0hat.Lin.ScottMat_emp <- pi0hatScottMat_empFitLin[,1:ntest]
  FDR.Lin.ScottMat_emp <- pi0hatScottMat_empFitLin[,(ntest+1):(2*ntest)]

  ##---------spline fit---------#
  print("spline")
  splineMat <- ns(tme,df=3)

  pi0hatScottMat_empFitSpl <- estimate_Scott_sims(zValuesSims, splineMat, nulltype)

  pi0hatSpl.ScottMean_emp <- colMeans(pi0hatScottMat_empFitSpl[,1:ntest])
  pi0hatSpl.ScottVar_emp <- apply(pi0hatScottMat_empFitSpl[,1:ntest],2,var)

  pi0hat.Spl.ScottMat_emp <- pi0hatScottMat_empFitSpl[,1:ntest]
  FDR.Spl.ScottMat_emp <- pi0hatScottMat_empFitSpl[,(ntest+1):(2*ntest)]

  ##save full results
  save(file=paste(alt,"simResults_pi0x_Scott_emp_2_full.RData",sep="/"),
       list=c("pi0hat.Lin.ScottMat_emp", "FDR.Lin.ScottMat_emp",
              "pi0hat.Spl.ScottMat_emp", "FDR.Spl.ScottMat_emp"))

  ##save summary results
```

```
  save(file=paste(alt,"simResults_pi0x_Scott_emp_2.RData",sep="/"),
       list=c("tme", "pi0",
              "pi0hatLin.ScottMean_emp", "pi0hatLin.ScottVar_emp",
              "pi0hatSpl.ScottMean_emp", "pi0hatSpl.ScottVar_emp"))
}
```

```
## [1] "alt_beta"
## [1] "linear"
## [1] "spline"
## [1] "alt_chisq_large_3_3"
## [1] "linear"
## [1] "spline"
## [1] "alt_chisq_large"
## [1] "linear"
```

```
## Warning in apply(as.matrix(pi0hatScottMat), 2, as.numeric): NAs
introduced by coercion
## Warning in apply(as.matrix(pi0hatScottMat), 2, as.numeric): NAs
introduced by coercion
## Warning in apply(as.matrix(pi0hatScottMat), 2, as.numeric): NAs
introduced by coercion
## Warning in apply(as.matrix(pi0hatScottMat), 2, as.numeric): NAs
introduced by coercion
## Warning in apply(as.matrix(pi0hatScottMat), 2, as.numeric): NAs
introduced by coercion
## Error in apply(as.matrix(pi0hatScottMat), 2, as.numeric): (list)
object cannot be coerced to type 'double'
```

3 Probability of being a false positive is smooth in one variable within levels of second variable

Perform estimation and save estimates:

```
for(alt in alts)
{
  print(alt)

  load(paste(alt,"simResults_3.RData",sep="/"))
  ntest <- ncol(zValuesSims)

  m <- model.matrix(~as.character(tme2))[,-1]

  linearMat <- cbind(tme1, m)
```

```
  splineMat <- cbind(ns(tme1,df=3), m)

  ##--------linear fit-------##
  print("linear")
  pi0hatScottMat_empFitLin <- estimate_Scott_sims(zValuesSims, linearMat, nulltype)

  pi0hatLin.ScottMean_emp <- colMeans(pi0hatScottMat_empFitLin[,1:ntest])
  pi0hatLin.ScottVar_emp <- apply(pi0hatScottMat_empFitLin[,1:ntest],2,var)

  pi0hat.Lin.ScottMat_emp <- pi0hatScottMat_empFitLin[,1:ntest]
  FDR.Lin.ScottMat_emp <- pi0hatScottMat_empFitLin[,(ntest+1):(2*ntest)]

  ##---------spline fit---------#
  print("spline")
  pi0hatScottMat_empFitSpl <- estimate_Scott_sims(zValuesSims, splineMat, nulltype)

  pi0hatSpl.ScottMean_emp <- colMeans(pi0hatScottMat_empFitSpl[,1:ntest])
  pi0hatSpl.ScottVar_emp <- apply(pi0hatScottMat_empFitSpl[,1:ntest],2,var)

  pi0hat.Spl.ScottMat_emp <- pi0hatScottMat_empFitSpl[,1:ntest]
  FDR.Spl.ScottMat_emp <- pi0hatScottMat_empFitSpl[,(ntest+1):(2*ntest)]

  ##save full results
  save(file=paste(alt,"simResults_pi0x_Scott_emp_3_full.RData",sep="/"),
       list=c("pi0hat.Lin.ScottMat_emp", "FDR.Lin.ScottMat_emp",
              "pi0hat.Spl.ScottMat_emp", "FDR.Spl.ScottMat_emp"))

  ##save summary results
  save(file=paste(alt,"simResults_pi0x_Scott_emp_3.RData",sep="/"),
       list=c("tme", "pi0",
              "pi0hatLin.ScottMean_emp", "pi0hatLin.ScottVar_emp",
              "pi0hatSpl.ScottMean_emp", "pi0hatSpl.ScottVar_emp"))
}
```

```
## [1] "alt_beta"
## [1] "linear"
## [1] "spline"
## [1] "alt_chisq_large_3_3"
## [1] "linear"
## [1] "spline"
## [1] "alt_chisq_large"
## [1] "linear"
## [1] "spline"
## [1] "alt_chisq_small_3_3"
## [1] "linear"
## [1] "spline"
```

```
## [1] "alt_chisq_small"
## [1] "linear"
## [1] "spline"
## [1] "alt_t_large"
## [1] "linear"
## [1] "spline"
## [1] "alt_t_small"
## [1] "linear"
## [1] "spline"
## [1] "alt_z_large"
## [1] "linear"
## [1] "spline"
## [1] "alt_z_small"
## [1] "linear"
## [1] "spline"
```

4 Probability of being a false positive is smooth in one variable within levels of second variable - lower priors

Perform estimation and save estimates:

```
for(alt in alts)
{
  print(alt)

  load(paste(alt,"simResults_4.RData",sep="/"))
  ntest <- ncol(zValuesSims)

  m <- model.matrix(~as.character(tme2))[,-1]

  linearMat <- cbind(tme1, m)
  splineMat <- cbind(ns(tme1,df=3), m)

  ##--------linear fit-------##
  print("linear")
  piOhatScottMat_empFitLin <- estimate_Scott_sims(zValuesSims, linearMat, nulltype)

  ##if only have 2 columns, make everything NULL (this means there was an error in every si
  if(ncol(piOhatScottMat_empFitLin) > 2)
  {
    piOhatLin.ScottMean_emp <- colMeans(piOhatScottMat_empFitLin[,1:ntest])
    piOhatLin.ScottVar_emp <- apply(piOhatScottMat_empFitLin[,1:ntest],2,var)
```

```
    pi0hat.Lin.ScottMat_emp <- pi0hatScottMat_empFitLin[,1:ntest]
    FDR.Lin.ScottMat_emp <- pi0hatScottMat_empFitLin[,(ntest+1):(2*ntest)]
  } else {
    pi0hatLin.ScottMean_emp <- pi0hatLin.ScottVar_emp <-
      pi0hat.Lin.ScottMat_emp <- FDR.Lin.ScottMat_emp <- NULL
  }

  ##---------spline fit---------#
  print("spline")
  pi0hatScottMat_empFitSpl <- estimate_Scott_sims(zValuesSims, splineMat, nulltype)

  ##if only have 2 columns, make everything NULL (this means there was an error in every si
  if(ncol(pi0hatScottMat_empFitLin) > 2)
  {
    pi0hatSpl.ScottMean_emp <- colMeans(pi0hatScottMat_empFitSpl[,1:ntest])
    pi0hatSpl.ScottVar_emp <- apply(pi0hatScottMat_empFitSpl[,1:ntest],2,var)

    pi0hat.Spl.ScottMat_emp <- pi0hatScottMat_empFitSpl[,1:ntest]
    FDR.Spl.ScottMat_emp <- pi0hatScottMat_empFitSpl[,(ntest+1):(2*ntest)]
  } else {
    pi0hatSpl.ScottMean_emp <- pi0hatSpl.ScottVar_emp <-
      pi0hat.Spl.ScottMat_emp <- FDR.Spl.ScottMat_emp <- NULL
  }

  ##save full results
  save(file=paste(alt,"simResults_pi0x_Scott_emp_4_full.RData",sep="/"),
       list=c("pi0hat.Lin.ScottMat_emp", "FDR.Lin.ScottMat_emp",
              "pi0hat.Spl.ScottMat_emp", "FDR.Spl.ScottMat_emp"))

  ##save summary results
  save(file=paste(alt,"simResults_pi0x_Scott_emp_4.RData",sep="/"),
       list=c("tme", "pi0",
              "pi0hatLin.ScottMean_emp", "pi0hatLin.ScottVar_emp",
              "pi0hatSpl.ScottMean_emp", "pi0hatSpl.ScottVar_emp"))
}


## [1] "alt_beta"
## [1] "linear"
## [1] "spline"
## [1] "alt_chisq_large_3_3"
## [1] "linear"


## Warning in apply(as.matrix(pi0hatScottMat), 2, as.numeric):  NAs
introduced by coercion
```

```
## Error in apply(as.matrix(piOhatScottMat), 2, as.numeric):  (list)
object cannot be coerced to type 'double'
```

Session info:

```
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     2017-06-13

## Packages -----------------------------------------------------

##  package    * version date
##  assertthat   0.1     2013-12-06
##  BayesLogit * 0.6     2016-10-20
##  codetools    0.2-14  2015-07-15
##  colorspace   1.2-6   2015-03-11
##  curl       * 0.9.7   2016-04-10
##  DBI          0.4-1   2016-05-08
##  devtools     1.12.0  2016-06-24
##  digest       0.6.9   2016-01-08
##  doParallel * 1.0.10  2015-10-14
##  doRNG      * 1.6     2014-03-07
##  dplyr        0.4.3   2015-09-01
##  evaluate     0.10    2016-10-11
##  fda        * 2.4.4   2014-12-16
##  FDRreg     * 0.2-1   2017-05-03
##  foreach    * 1.4.3   2015-10-13
##  ggdendro     0.1-20  2016-04-27
##  ggplot2      2.2.1   2016-12-30
##  gridExtra    2.2.1   2016-02-29
##  gtable       0.2.0   2016-02-26
##  highr        0.6     2016-05-09
##  iterators  * 1.0.8   2015-10-13
##  knitr      * 1.15.1  2016-11-22
##  lattice      0.20-33 2015-07-14
##  lazyeval     0.2.0   2016-06-12
##  magrittr     1.5     2014-11-22
##  MASS       * 7.3-45  2016-04-21
```

```
##  Matrix     * 1.2-6   2016-05-02
##  memoise      1.0.0   2016-01-29
##  mosaic       0.14.4  2016-07-29
##  mosaicData   0.14.0  2016-06-17
##  munsell      0.4.3   2016-02-13
##  mvtnorm    * 1.0-6   2017-03-02
##  pkgmaker   * 0.22    2014-05-14
##  plyr         1.8.4   2016-06-08
##  R6           2.1.2   2016-01-26
##  Rcpp         0.12.10 2017-03-19
##  registry   * 0.3     2015-07-08
##  rngtools   * 1.2.4   2014-03-06
##  scales       0.4.1   2016-11-09
##  stringi      1.1.1   2016-05-27
##  stringr      1.0.0   2015-04-30
##  tibble       1.2     2016-08-26
##  tidyr        0.5.1   2016-06-14
##  withr        1.0.2   2016-06-20
##  xtable       1.8-2   2016-02-05
##  source
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.2)
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.3)
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.3)
##  Github (jgscott/FDRreg@8025d1a)
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.3)
##  CRAN (R 3.3.3)
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.0)
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.1)
```

```
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.3)
##  CRAN (R 3.3.3)
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.2)
##  CRAN (R 3.3.2)
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.3)
##  CRAN (R 3.3.2)
##  CRAN (R 3.3.2)
##  CRAN (R 3.3.3)
##  CRAN (R 3.3.0)
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.2)
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.1)
##  CRAN (R 3.3.1)
```