Comparison between different local tests: Simes, Simes with Storey and Wilcoxon-Mann-Whitney

27-04-2023

The aim is to compare on real datasets the performance of three closed testing procedures, which respectively use Simes local test with and without Storey estimator for the proportion of true null hypotheses and Wilcoxon-Mann-Whitney local test.

R functions and libraries

```
library(nout)
library(R.matlab)
library(isotree)
library(farff)
library(tictoc)


sim_realdata = function(B, dataset, m1, m, n, l, in_index,
                        out_index=NULL, alpha=m/(l+1), lambda = 0.5){

  m0=m-m1
  if(m1!=0 & is.null(out_index)){
    stop("Error: arg out_index must be initialized.")
  }

  # if(m!=(m1+m0)){
  #   stop("Error: equation m=m1+m0 must be verified.")
  # }

  if(m1!=0){
    tr_ind = sample(in_index, size = n)
    tr = dataset[tr_ind,]
    iso.fo = isolation.forest(tr, ndim=ncol(dataset), ntrees=10, nthreads=1,
                              scoring_metric = "depth", output_score = TRUE)
    in_index2 = setdiff(in_index, tr_ind)

    crit=critWMW(m=m, n=n, alpha=alpha)

    d_WMW = rep(0,B)
    d_Simes = rep(0,B)
    d_StoSimes = rep(0,B)
    d_BH = rep(0,B)
    d_StoBH = rep(0,B)

    for(b in 1:B){
      cal_ind = sample(in_index2, size = l)
      in_index3 = setdiff(in_index2, cal_ind)
```

```
    tein_ind = sample(in_index3, size = m0)
    teout_ind = sample(out_index, size = m1)

    cal = dataset[cal_ind,]
    te = dataset[c(tein_ind, teout_ind),]

    S_cal = predict.isolation_forest(iso.fo$model, cal, type = "score")
    S_te = predict.isolation_forest(iso.fo$model, te, type = "score")

    d_WMW[b] = d_mannwhitney(S_X=S_cal, S_Y=S_te, crit=crit)
    d_Simes[b] = d_Simes(S_X=S_cal, S_Y=S_te, alpha=alpha)
    d_StoSimes[b] = d_StoreySimes(S_X=S_cal, S_Y=S_te, alpha=alpha)
    d_BH[b] = d_benjhoch(S_X=S_cal, S_Y=S_te, alpha=alpha)
    d_StoBH[b] = d_StoreyBH(S_X=S_cal, S_Y=S_te, alpha=alpha)
  }
}

else{
  tr_ind = sample(in_index, size = n)
  tr = dataset[tr_ind,]
  iso.fo = isolation.forest(tr, ndim=ncol(dataset), ntrees=10, nthreads=1,
                            scoring_metric = "depth", output_score = TRUE)
  in_index2 = setdiff(in_index, tr_ind)

  crit=critWMW(m=m, n=n, alpha=alpha)

  d_WMW = rep(0,B)
  d_Simes = rep(0,B)
  d_StoSimes = rep(0,B)
  d_BH = rep(0,B)
  d_StoBH = rep(0,B)

  for(b in 1:B){
    cal_ind = sample(in_index2, size = l)
    in_index3 = setdiff(in_index2, cal_ind)
    te_ind = sample(in_index3, size = m0)

    cal = dataset[cal_ind,]
    te = dataset[te_ind,]

    S_cal = predict.isolation_forest(iso.fo$model, cal, type = "score")
    S_te = predict.isolation_forest(iso.fo$model, te, type = "score")

    d_WMW[b] = d_mannwhitney(S_X=S_cal, S_Y=S_te, crit=crit)
    d_Simes[b] = d_Simes(S_X=S_cal, S_Y=S_te, alpha=alpha)
    d_StoSimes[b] = d_StoreySimes(S_X=S_cal, S_Y=S_te, alpha=alpha)
    d_BH[b] = d_benjhoch(S_X=S_cal, S_Y=S_te, alpha=alpha)
    d_StoBH[b] = d_StoreyBH(S_X=S_cal, S_Y=S_te, alpha=alpha)
  }
}

discov = as.data.frame(cbind("d_BH"=d_BH, "d_StoBH"=d_StoBH, "d_Simes"=d_Simes,
                             "d_StoSimes"=d_StoSimes, "d_WMW"=d_WMW))
```

```
  colnames(discov) = c("BH", "BHSto", "CTSim", "CTSimSto", "CTWMW")
  mean.discov = apply(discov, MARGIN = 2, FUN = mean)

  powerGlobalNull = as.data.frame(cbind("d_BH"=d_BH>0, "d_StoBH"=d_StoBH>0, "d_Simes"=d_Simes>0,
                                        "d_StoSimes"=d_StoSimes>0, "d_WMW"=d_WMW>0))
  colnames(powerGlobalNull) = c("BH", "BHSto", "CTSim", "CTSimSto", "CTWMW")
  mean.powerGlobalNull = apply(powerGlobalNull, MARGIN = 2, FUN = mean)

  return(list("discoveries"=discov, "mean.discoveries" = mean.discov,
              "powerGlobalNull"=powerGlobalNull, "mean.powerGlobalNull"=mean.powerGlobalNull,
              "m1"=m1, "alpha"=alpha))

}
```

In the following we set the calibration set and the test set size, respectively l and m, so that the nominal level α is proportional to $\frac{m}{l+1}$. The train set size is equal to n and the number of iterations is $B = 10^5$.

Statlog (Shuttle) dataset

The dataset is available at http://odds.cs.stonybrook.edu/shuttle-dataset

```
set.seed(321)

# Initializing parameters
B=10^5
n = 199
l = 199
m = 20
alpha = m/(l+1)
m1s = seq(from=0, to=m, by=1)

data = readMat("G:\\Il mio Drive\\PHD\\Progetto di ricerca\\Conformal Inference Project\\Simulazioni\\7
dataset = cbind(data$X, data$y); colnames(dataset)[ncol(dataset)] = "y"
in_ind = which(dataset[,ncol(dataset)]==0)
out_ind = which(dataset[,ncol(dataset)]==1)

tic()
res = lapply(m1s,
             function(m1) sim_realdata(B=B, in_index=in_ind, out_index=out_ind,
                                       dataset=dataset,
                                       alpha=alpha,l=l, n=n, m=m, m1=m1))
toc()
```

```
## 19278.39 sec elapsed
```

```
# Storing results
store_res = list("mean.discov" = matrix(nrow=length(m1s), ncol = 5),
                 "mean.powerGlobalNull" = matrix(nrow=length(m1s), ncol = 5))
row.names = rep(NA, times=length(m1s))
for(i in 1:length(m1s)){
  row.names[i] = paste("m1 =",m1s[i])
}
rownames(store_res$mean.discov) = row.names
colnames(store_res$mean.discov) = c("BH", "StoBH", "Simes", "StoSimes", "WMW")
rownames(store_res$mean.powerGlobalNull) = row.names
```

```r
colnames(store_res$mean.powerGlobalNull) = c("BH", "StoBH", "Simes", "StoSimes", "WMW")


for(i in 1:length(res)){
  store_res$mean.discov[i,] = res[[i]]$mean.discov
  store_res$mean.powerGlobalNull[i,] = res[[i]]$mean.powerGlobalNull
}

store_res$mean.discov
```

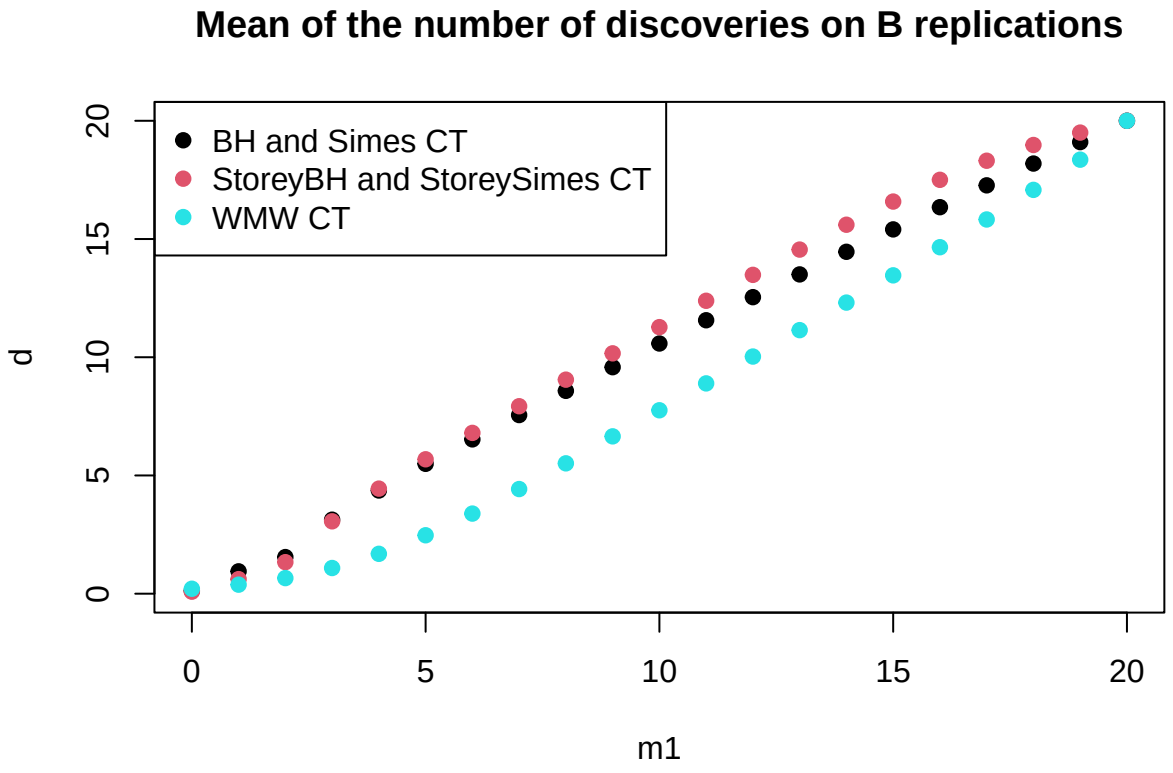
```
##               BH    StoBH    Simes StoSimes      WMW
## m1 = 0   0.12279  0.08486  0.10161  0.05985  0.20878
## m1 = 1   0.94485  0.61588  0.81613  0.47356  0.38188
## m1 = 2   1.54834  1.33319  1.02342  0.79384  0.65917
## m1 = 3   3.12925  3.06055  2.16253  1.91560  1.08548
## m1 = 4   4.36379  4.44283  3.11108  2.95512  1.68555
## m1 = 5   5.49174  5.68398  4.47186  4.37744  2.47015
## m1 = 6   6.52677  6.80009  5.04057  5.09199  3.38766
## m1 = 7   7.55537  7.92794  6.12720  6.22005  4.42489
## m1 = 8   8.57880  9.05095  7.06136  7.17058  5.51207
## m1 = 9   9.57938 10.16423  8.58504  8.63514  6.65332
## m1 = 10 10.58090 11.27342  9.50165  9.56517  7.75699
## m1 = 11 11.56448 12.38813 10.68115 10.51653  8.89366
## m1 = 12 12.54102 13.48422 11.76778 11.63767 10.02891
## m1 = 13 13.50335 14.55353 12.66060 12.63076 11.14614
## m1 = 14 14.45985 15.60399 13.62159 13.65300 12.31098
## m1 = 15 15.40672 16.58427 14.91890 14.81112 13.46371
## m1 = 16 16.34577 17.50387 16.06593 15.93064 14.65287
## m1 = 17 17.27025 18.31087 17.01430 16.81155 15.82672
## m1 = 18 18.19114 18.97680 18.10663 18.03269 17.07858
## m1 = 19 19.09917 19.50049 19.09703 19.02492 18.35577
## m1 = 20 20.00000 20.00000 20.00000 19.99913 20.00000
```

```r
store_res$mean.powerGlobalNull
```

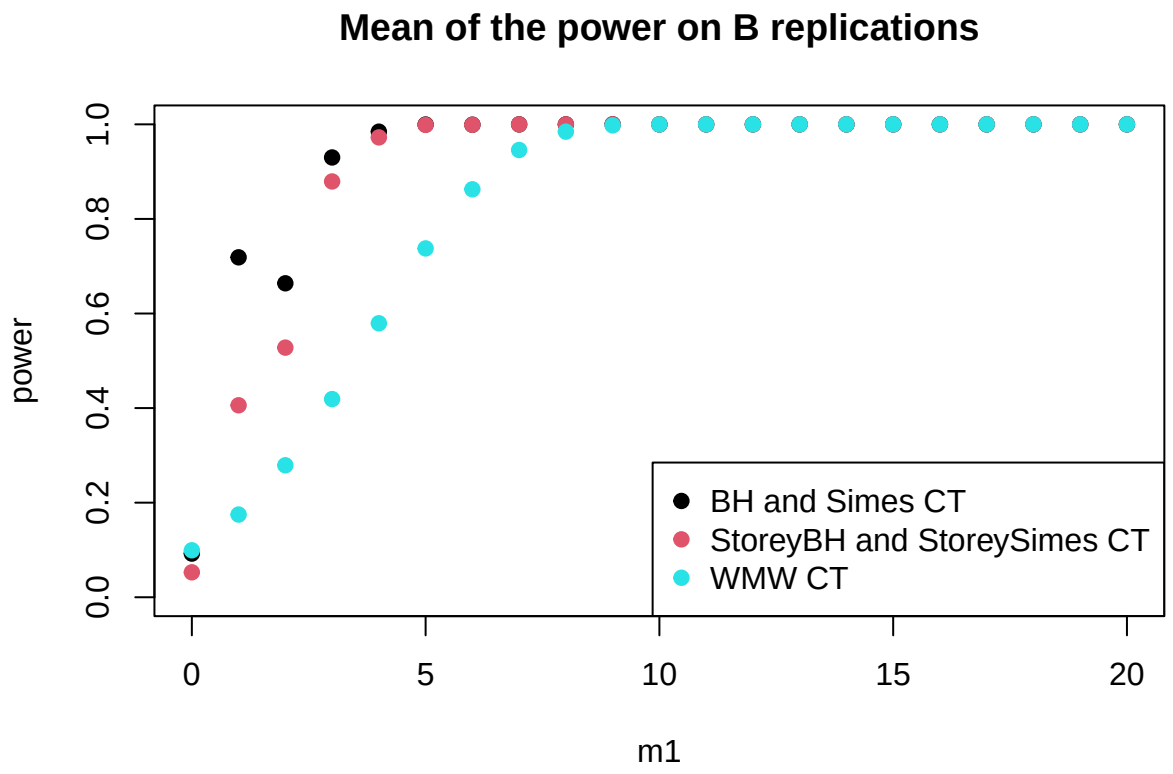
```
##              BH   StoBH   Simes StoSimes     WMW
## m1 = 0  0.09232 0.05273 0.09232  0.05273 0.09926
## m1 = 1  0.71878 0.40593 0.71878  0.40593 0.17486
## m1 = 2  0.66386 0.52784 0.66386  0.52784 0.27904
## m1 = 3  0.92997 0.87922 0.92997  0.87922 0.41901
## m1 = 4  0.98451 0.97243 0.98451  0.97243 0.57936
## m1 = 5  0.99963 0.99883 0.99963  0.99883 0.73764
## m1 = 6  0.99924 0.99903 0.99924  0.99903 0.86265
## m1 = 7  0.99998 0.99996 0.99998  0.99996 0.94575
## m1 = 8  0.99996 0.99998 0.99996  0.99998 0.98461
## m1 = 9  1.00000 1.00000 1.00000  1.00000 0.99788
## m1 = 10 1.00000 1.00000 1.00000  1.00000 0.99987
## m1 = 11 1.00000 1.00000 1.00000  1.00000 1.00000
## m1 = 12 1.00000 1.00000 1.00000  1.00000 1.00000
## m1 = 13 1.00000 1.00000 1.00000  1.00000 1.00000
## m1 = 14 1.00000 1.00000 1.00000  1.00000 1.00000
## m1 = 15 1.00000 1.00000 1.00000  1.00000 1.00000
## m1 = 16 1.00000 1.00000 1.00000  1.00000 1.00000
## m1 = 17 1.00000 1.00000 1.00000  1.00000 1.00000
```

```
## m1 = 18 1.00000 1.00000 1.00000  1.00000 1.00000
## m1 = 19 1.00000 1.00000 1.00000  1.00000 1.00000
## m1 = 20 1.00000 1.00000 1.00000  1.00000 1.00000
```

```
plot(x = m1s, y = store_res$mean.discov[,1], col = 1, ylab = "d",
     xlab = "m1", ylim=c(0,m), pch=19,
     main = "Mean of the number of discoveries on B replications")
points(x = m1s, y = store_res$mean.discov[,2], col = 2, pch=19)
points(x = m1s, y = store_res$mean.discov[,5], col = 5, pch=19)
legend("topleft", pch = 19, col = c(1,2,5),
       legend =c("BH and Simes CT", "StoreyBH and StoreySimes CT", "WMW CT"))
```




```
plot(x = m1s, y = store_res$mean.powerGlobalNull[,1], col = 1, ylab = "power",
     xlab = "m1", ylim=c(0,1), pch = 19,
     main = "Mean of the power on B replications")
points(x = m1s, y = store_res$mean.powerGlobalNull[,2], col = 2, pch=19)
points(x = m1s, y = store_res$mean.powerGlobalNull[,5], col = 5, pch=19)
legend("bottomright", pch = 19, col = c(1,2,5),
       legend = c("BH and Simes CT", "StoreyBH and StoreySimes CT", "WMW CT"))
```



Credit Card Fraud Detection dataset

The dataset is available at https://www.kaggle.com/mlg-ulb/creditcardfraud.

```
set.seed(321)

# Initializing parameters
B=10^5
n = 199
l = 199
m = 20
alpha = m/(l+1)
m1s = seq(from=0, to=m, by=1)

dataset = read.csv("G:\\Il mio Drive\\PHD\\Progetto di ricerca\\Conformal Inference Project\\Simulazioni
in_ind = which(dataset[,ncol(dataset)]==0)
out_ind = which(dataset[,ncol(dataset)]==1)

tic()
res = lapply(m1s,
             function(m1) sim_realdata(B=B, in_index=in_ind, out_index=out_ind,
                                       dataset=dataset,
                                       alpha=alpha,l=l, n=n, m=m, m1=m1))
toc()
```

```
## 41313.31 sec elapsed
```

```r
# Storing results
store_res = list("mean.discov" = matrix(nrow=length(m1s), ncol = 5),
                 "mean.powerGlobalNull" = matrix(nrow=length(m1s), ncol = 5))
row.names = rep(NA, times=length(m1s))
for(i in 1:length(m1s)){
  row.names[i] = paste("m1 =",m1s[i])
}
rownames(store_res$mean.discov) = row.names
colnames(store_res$mean.discov) = c("BH", "StoBH", "Simes", "StoSimes", "WMW")
rownames(store_res$mean.powerGlobalNull) = row.names
colnames(store_res$mean.powerGlobalNull) = c("BH", "StoBH", "Simes", "StoSimes", "WMW")


for(i in 1:length(res)){
  store_res$mean.discov[i,] = res[[i]]$mean.discov
  store_res$mean.powerGlobalNull[i,] = res[[i]]$mean.powerGlobalNull
}

store_res$mean.discov
```

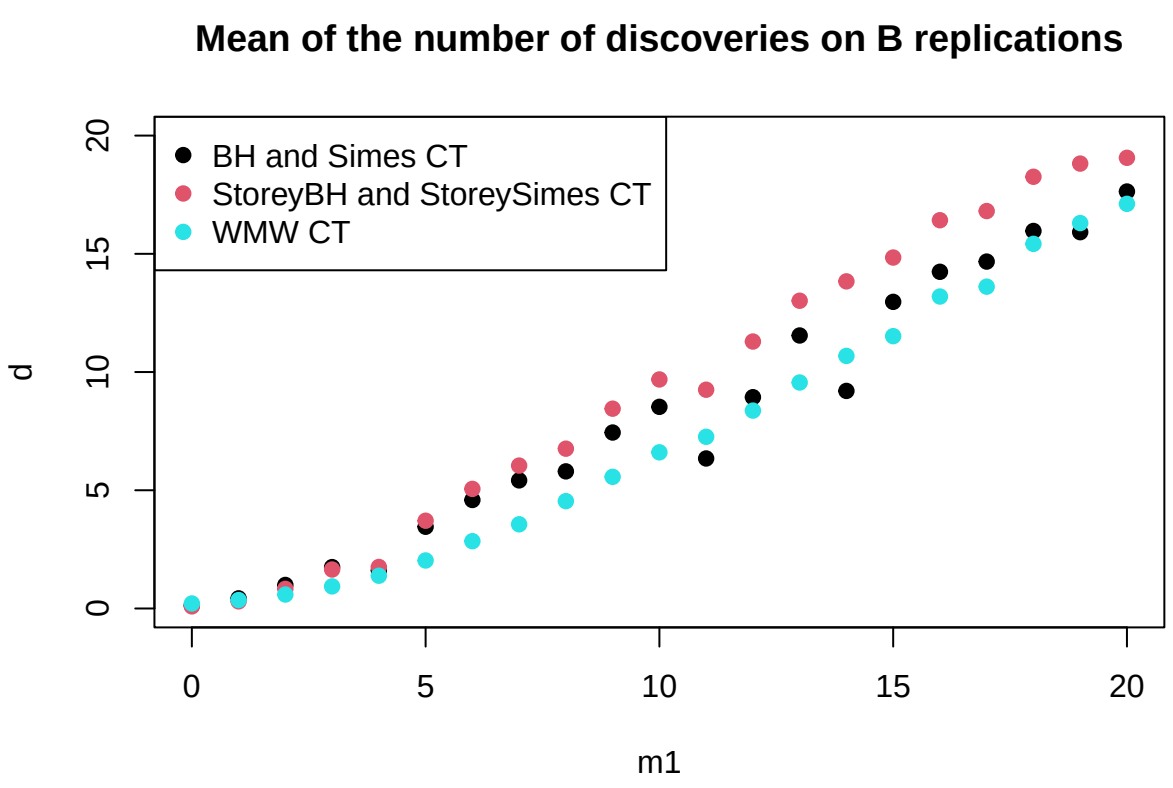
```
##               BH    StoBH    Simes StoSimes      WMW
## m1 = 0   0.13137  0.08952  0.10904  0.06395  0.21146
## m1 = 1   0.42787  0.30172  0.35137  0.21576  0.35768
## m1 = 2   0.99511  0.83130  0.73374  0.53382  0.59018
## m1 = 3   1.75049  1.64888  1.16729  0.97846  0.93422
## m1 = 4   1.61212  1.75529  0.91534  0.88946  1.38613
## m1 = 5   3.45583  3.71121  1.91781  1.95948  2.03052
## m1 = 6   4.58853  5.05795  2.63968  2.79055  2.84618
## m1 = 7   5.41810  6.04331  3.11707  3.37635  3.56102
## m1 = 8   5.79854  6.75889  3.51839  3.89316  4.54033
## m1 = 9   7.44188  8.45010  4.39301  4.95420  5.56787
## m1 = 10  8.52800  9.68687  5.39010  5.98538  6.60442
## m1 = 11  6.34480  9.25238  3.13631  4.46593  7.26202
## m1 = 12  8.93617 11.29071  5.11605  6.44375  8.36911
## m1 = 13 11.54573 13.01599  7.88831  8.78652  9.55791
## m1 = 14  9.20111 13.83500  4.51015  7.34167 10.68234
## m1 = 15 12.96802 14.84309  9.29787 10.43127 11.51752
## m1 = 16 14.23968 16.42124 10.90280 12.07921 13.19391
## m1 = 17 14.67192 16.80816 11.18350 12.39818 13.61119
## m1 = 18 15.96653 18.25932 12.30315 14.16140 15.42136
## m1 = 19 15.91561 18.81649 11.92176 14.53114 16.30054
## m1 = 20 17.63575 19.06049 15.61396 16.37830 17.11404
```

```r
store_res$mean.powerGlobalNull
```

```
##              BH   StoBH   Simes StoSimes     WMW
## m1 = 0  0.09893 0.05645 0.09893  0.05645 0.10139
## m1 = 1  0.30781 0.18246 0.30781  0.18246 0.16315
## m1 = 2  0.54399 0.39149 0.54399  0.39149 0.25175
## m1 = 3  0.70538 0.58952 0.70538  0.58952 0.36866
## m1 = 4  0.56711 0.52049 0.56711  0.52049 0.49456
## m1 = 5  0.82977 0.80813 0.82977  0.80813 0.64634
## m1 = 6  0.92950 0.92346 0.92950  0.92346 0.78112
## m1 = 7  0.95677 0.95685 0.95677  0.95685 0.86461
```

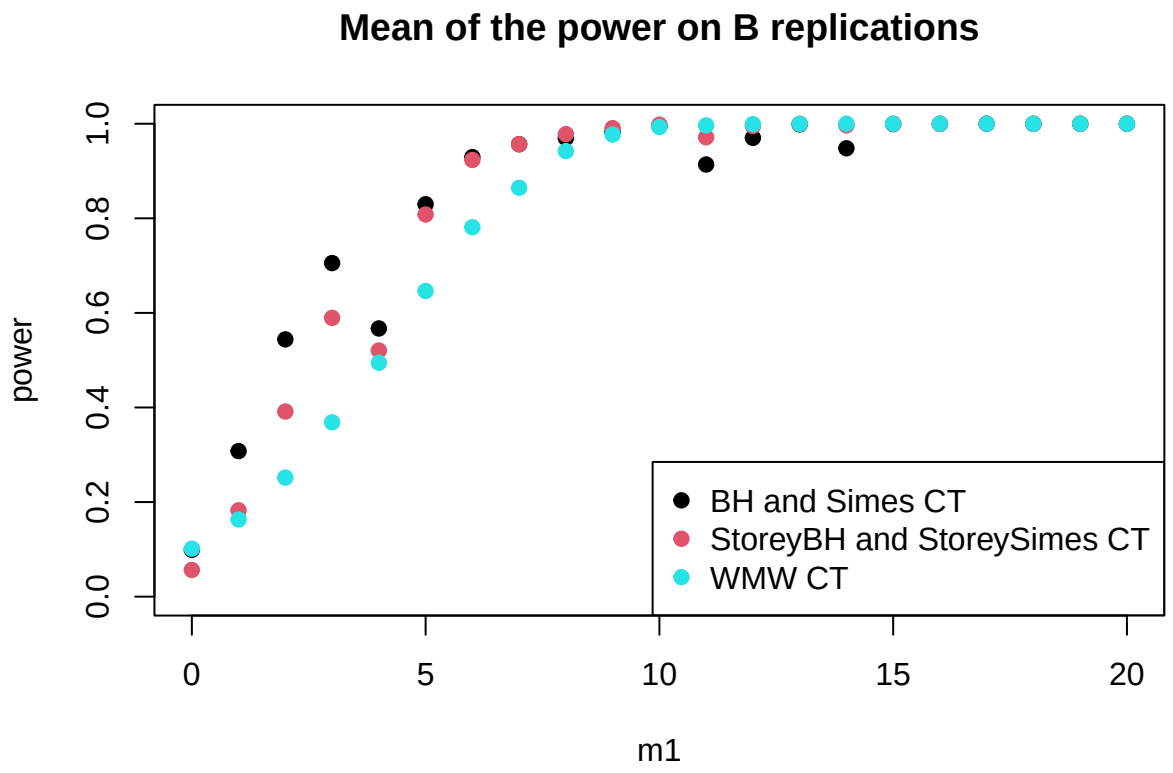
```
## m1 = 8  0.96971 0.97796 0.96971  0.97796 0.94210
## m1 = 9  0.98338 0.99068 0.98338  0.99068 0.97719
## m1 = 10 0.99531 0.99803 0.99531  0.99803 0.99313
## m1 = 11 0.91379 0.97138 0.91379  0.97138 0.99659
## m1 = 12 0.97022 0.99525 0.97022  0.99525 0.99906
## m1 = 13 0.99876 0.99988 0.99876  0.99988 0.99987
## m1 = 14 0.94814 0.99664 0.94814  0.99664 1.00000
## m1 = 15 0.99933 0.99999 0.99933  0.99999 1.00000
## m1 = 16 0.99980 0.99999 0.99980  0.99999 1.00000
## m1 = 17 0.99999 1.00000 0.99999  1.00000 1.00000
## m1 = 18 0.99964 1.00000 0.99964  1.00000 1.00000
## m1 = 19 0.99973 1.00000 0.99973  1.00000 1.00000
## m1 = 20 0.99997 1.00000 0.99997  1.00000 1.00000
```

```
plot(x = m1s, y = store_res$mean.discov[,1], col = 1, ylab = "d",
     xlab = "m1", ylim=c(0,m), pch=19,
     main = "Mean of the number of discoveries on B replications")
points(x = m1s, y = store_res$mean.discov[,2], col = 2, pch=19)
points(x = m1s, y = store_res$mean.discov[,5], col = 5, pch=19)
legend("topleft", pch = 19, col = c(1,2,5),
       legend =c("BH and Simes CT", "StoreyBH and StoreySimes CT", "WMW CT"))
```




```
plot(x = m1s, y = store_res$mean.powerGlobalNull[,1], col = 1, ylab = "power",
     xlab = "m1", ylim=c(0,1), pch = 19,
     main = "Mean of the power on B replications")
points(x = m1s, y = store_res$mean.powerGlobalNull[,2], col = 2, pch=19)
points(x = m1s, y = store_res$mean.powerGlobalNull[,5], col = 5, pch=19)
```

```
legend("bottomright", pch = 19, col = c(1,2,5),
       legend = c("BH and Simes CT", "StoreyBH and StoreySimes CT", "WMW CT"))
```



Covertype dataset

The dataset is available at http://odds.cs.stonybrook.edu/forestcovercovertype-dataset.

```
set.seed(321)

# Initializing parameters
B=10^5
n = 199
l = 199
m = 20
alpha = m/(l+1)
m1s = seq(from=0, to=m, by=1)

data = readMat("G:\\Il mio Drive\\PHD\\Progetto di ricerca\\Conformal Inference Project\\Simulazioni\\7
dataset = cbind(data$X, data$y); colnames(dataset)[ncol(dataset)] = "y"
in_ind = which(dataset[,ncol(dataset)]==0)
out_ind = which(dataset[,ncol(dataset)]==1)

tic()
res = lapply(m1s,
             function(m1) sim_realdata(B=B, in_index=in_ind, out_index=out_ind,
                                       dataset=dataset,
```

```
                                        alpha=alpha,l=l, n=n, m=m, m1=m1))
toc()
```

```
## 19122.75 sec elapsed
```

```
# Storing results
store_res = list("mean.discov" = matrix(nrow=length(m1s), ncol = 5),
                 "mean.powerGlobalNull" = matrix(nrow=length(m1s), ncol = 5))
row.names = rep(NA, times=length(m1s))
for(i in 1:length(m1s)){
  row.names[i] = paste("theta =",m1s[i])
}
rownames(store_res$mean.discov) = row.names
colnames(store_res$mean.discov) = c("BH", "StoBH", "Simes", "StoSimes", "WMW")
rownames(store_res$mean.powerGlobalNull) = row.names
colnames(store_res$mean.powerGlobalNull) = c("BH", "StoBH", "Simes", "StoSimes", "WMW")


for(i in 1:length(res)){
  store_res$mean.discov[i,] = res[[i]]$mean.discov
  store_res$mean.powerGlobalNull[i,] = res[[i]]$mean.powerGlobalNull
}

store_res$mean.discov
```

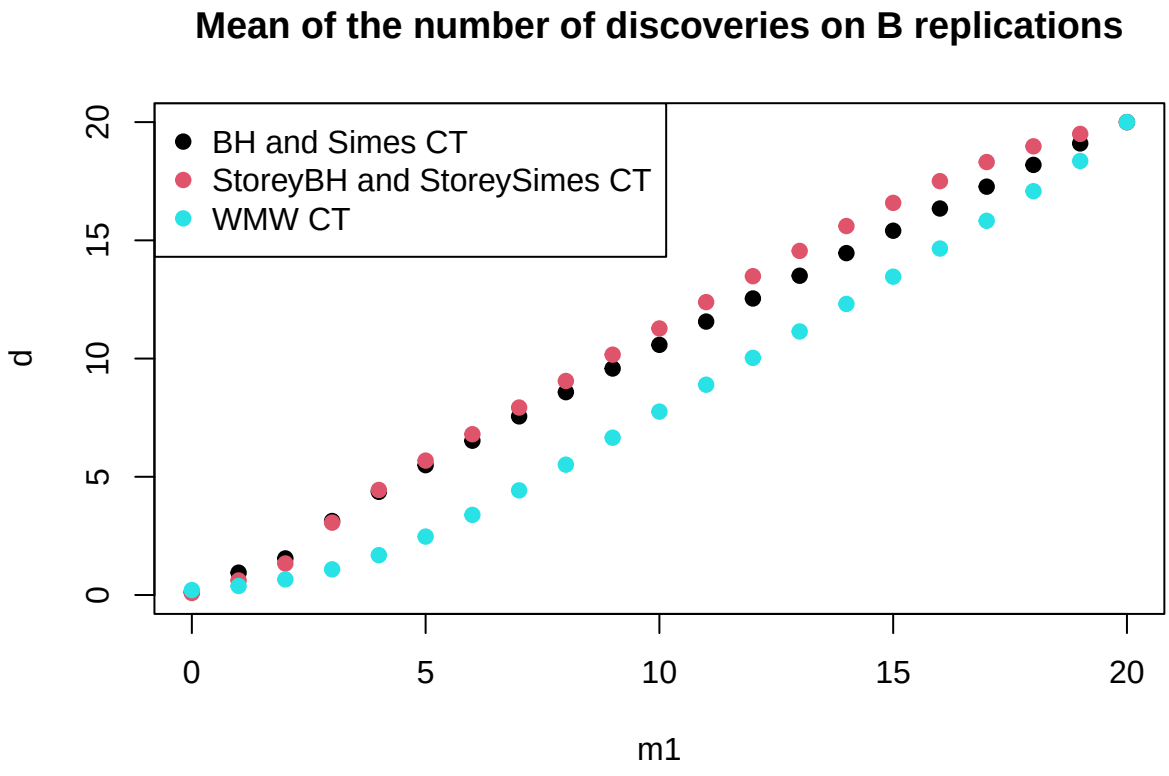
```
##                   BH    StoBH    Simes StoSimes      WMW
## theta = 0    0.12279  0.08486  0.10161  0.05985  0.20878
## theta = 1    0.94485  0.61588  0.81613  0.47356  0.38188
## theta = 2    1.54834  1.33319  1.02342  0.79384  0.65917
## theta = 3    3.12925  3.06055  2.16253  1.91560  1.08548
## theta = 4    4.36379  4.44283  3.11108  2.95512  1.68555
## theta = 5    5.49174  5.68398  4.47186  4.37744  2.47015
## theta = 6    6.52677  6.80009  5.04057  5.09199  3.38766
## theta = 7    7.55537  7.92794  6.12720  6.22005  4.42489
## theta = 8    8.57880  9.05095  7.06136  7.17058  5.51207
## theta = 9    9.57938 10.16423  8.58504  8.63514  6.65332
## theta = 10 10.58090 11.27342  9.50165  9.56517  7.75699
## theta = 11 11.56448 12.38813 10.68115 10.51653  8.89366
## theta = 12 12.54102 13.48422 11.76778 11.63767 10.02891
## theta = 13 13.50335 14.55353 12.66060 12.63076 11.14614
## theta = 14 14.45985 15.60399 13.62159 13.65300 12.31098
## theta = 15 15.40672 16.58427 14.91890 14.81112 13.46371
## theta = 16 16.34577 17.50387 16.06593 15.93064 14.65287
## theta = 17 17.27025 18.31087 17.01430 16.81155 15.82672
## theta = 18 18.19114 18.97680 18.10663 18.03269 17.07858
## theta = 19 19.09917 19.50049 19.09703 19.02492 18.35577
## theta = 20 20.00000 20.00000 20.00000 19.99913 20.00000
```

```
store_res$mean.powerGlobalNull
```

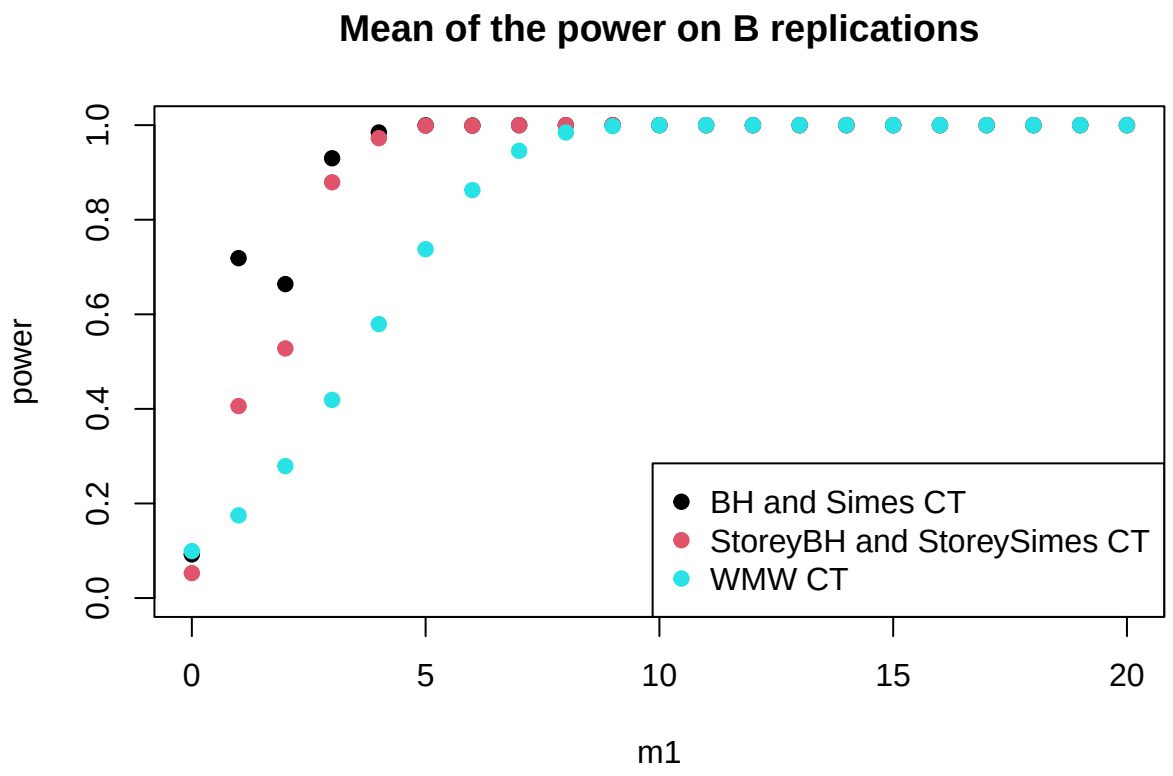
```
##                  BH   StoBH   Simes StoSimes     WMW
## theta = 0  0.09232 0.05273 0.09232  0.05273 0.09926
## theta = 1  0.71878 0.40593 0.71878  0.40593 0.17486
## theta = 2  0.66386 0.52784 0.66386  0.52784 0.27904
## theta = 3  0.92997 0.87922 0.92997  0.87922 0.41901
```

```
## theta = 4  0.98451 0.97243 0.98451  0.97243 0.57936
## theta = 5  0.99963 0.99883 0.99963  0.99883 0.73764
## theta = 6  0.99924 0.99903 0.99924  0.99903 0.86265
## theta = 7  0.99998 0.99996 0.99998  0.99996 0.94575
## theta = 8  0.99996 0.99998 0.99996  0.99998 0.98461
## theta = 9  1.00000 1.00000 1.00000  1.00000 0.99788
## theta = 10 1.00000 1.00000 1.00000  1.00000 0.99987
## theta = 11 1.00000 1.00000 1.00000  1.00000 1.00000
## theta = 12 1.00000 1.00000 1.00000  1.00000 1.00000
## theta = 13 1.00000 1.00000 1.00000  1.00000 1.00000
## theta = 14 1.00000 1.00000 1.00000  1.00000 1.00000
## theta = 15 1.00000 1.00000 1.00000  1.00000 1.00000
## theta = 16 1.00000 1.00000 1.00000  1.00000 1.00000
## theta = 17 1.00000 1.00000 1.00000  1.00000 1.00000
## theta = 18 1.00000 1.00000 1.00000  1.00000 1.00000
## theta = 19 1.00000 1.00000 1.00000  1.00000 1.00000
## theta = 20 1.00000 1.00000 1.00000  1.00000 1.00000
```

```r
plot(x = m1s, y = store_res$mean.discov[,1], col = 1, ylab = "d",
     xlab = "m1", ylim=c(0,m), pch=19,
     main = "Mean of the number of discoveries on B replications")
points(x = m1s, y = store_res$mean.discov[,2], col = 2, pch=19)
points(x = m1s, y = store_res$mean.discov[,5], col = 5, pch=19)
legend("topleft", pch = 19, col = c(1,2,5),
       legend =c("BH and Simes CT", "StoreyBH and StoreySimes CT", "WMW CT"))
```



```r
plot(x = m1s, y = store_res$mean.powerGlobalNull[,1], col = 1, ylab = "power",
     xlab = "m1", ylim=c(0,1), pch = 19,
     main = "Mean of the power on B replications")
points(x = m1s, y = store_res$mean.powerGlobalNull[,2], col = 2, pch=19)
points(x = m1s, y = store_res$mean.powerGlobalNull[,5], col = 5, pch=19)
legend("bottomright", pch = 19, col = c(1,2,5),
       legend = c("BH and Simes CT", "StoreyBH and StoreySimes CT", "WMW CT"))
```



Mammography dataset

The dataset is available at http://odds.cs.stonybrook.edu/mammography-dataset/.

```r
set.seed(321)

# Initializing parameters
B=10^5
n = 199
l = 199
m = 20
alpha = m/(l+1)
m1s = seq(from=0, to=m, by=1)
data = readMat("G:\\Il mio Drive\\PHD\\Progetto di ricerca\\Conformal Inference Project\\Simulazioni\\7
dataset = cbind(data$X, data$y); colnames(dataset)[ncol(dataset)] = "y"
in_ind = which(dataset[,ncol(dataset)]==0)
out_ind = which(dataset[,ncol(dataset)]==1)
```

```
tic()
res = lapply(m1s,
             function(m1) sim_realdata(B=B, in_index=in_ind, out_index=out_ind,
                                       dataset=dataset,
                                       alpha=alpha,l=l, n=n, m=m, m1=m1))
toc()
```

```
## 7466.89 sec elapsed
```

```
# Storing results
store_res = list("mean.discov" = matrix(nrow=length(m1s), ncol = 5),
                 "mean.powerGlobalNull" = matrix(nrow=length(m1s), ncol = 5))
row.names = rep(NA, times=length(m1s))
for(i in 1:length(m1s)){
  row.names[i] = paste("theta =",m1s[i])
}
rownames(store_res$mean.discov) = row.names
colnames(store_res$mean.discov) = c("BH", "StoBH", "Simes", "StoSimes", "WMW")
rownames(store_res$mean.powerGlobalNull) = row.names
colnames(store_res$mean.powerGlobalNull) = c("BH", "StoBH", "Simes", "StoSimes", "WMW")


for(i in 1:length(res)){
  store_res$mean.discov[i,] = res[[i]]$mean.discov
  store_res$mean.powerGlobalNull[i,] = res[[i]]$mean.powerGlobalNull
}

store_res$mean.discov
```

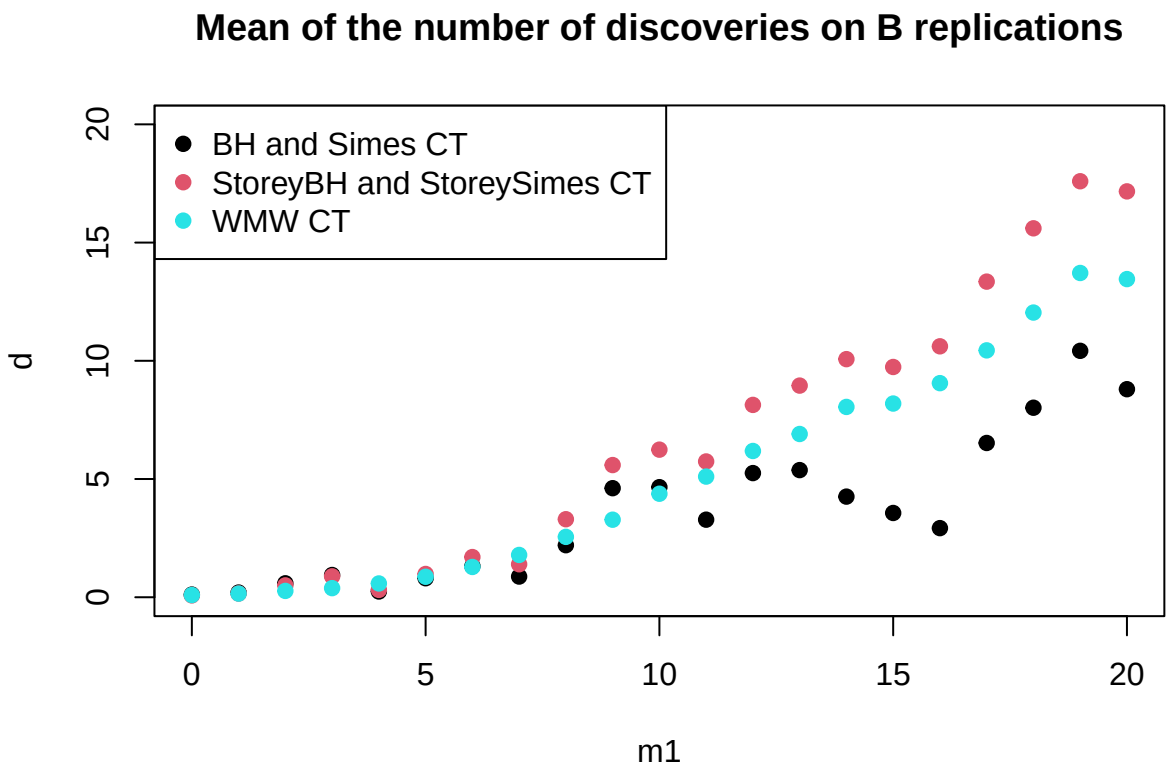
```
##                  BH    StoBH   Simes StoSimes      WMW
## theta = 0   0.11285  0.07884 0.09265  0.05599  0.09353
## theta = 1   0.19804  0.15595 0.15331  0.10284  0.15397
## theta = 2   0.58977  0.50105 0.41360  0.30749  0.26778
## theta = 3   0.94054  0.88381 0.62183  0.51348  0.38503
## theta = 4   0.24368  0.31113 0.16966  0.17345  0.58230
## theta = 5   0.80428  0.98381 0.48323  0.51743  0.86545
## theta = 6   1.31698  1.69870 0.72944  0.84633  1.28137
## theta = 7   0.87502  1.39469 0.54648  0.68652  1.78883
## theta = 8   2.20159  3.30004 1.09866  1.51428  2.55344
## theta = 9   4.61290  5.59056 2.62053  3.13552  3.27906
## theta = 10  4.65558  6.24261 2.67047  3.32808  4.37781
## theta = 11  3.28094  5.74222 2.16686  2.80753  5.10262
## theta = 12  5.25129  8.13432 2.59824  3.97184  6.18526
## theta = 13  5.37658  8.95153 2.58743  4.35017  6.90386
## theta = 14  4.25548 10.07048 1.86803  4.31074  8.04929
## theta = 15  3.56350  9.73787 1.55712  4.07104  8.19164
## theta = 16  2.92335 10.61254 1.26903  4.31121  9.05459
## theta = 17  6.52435 13.34853 3.30233  6.45656 10.44109
## theta = 18  8.01458 15.60255 4.00583  8.21239 12.04023
## theta = 19 10.42146 17.59337 5.61992 10.71793 13.71350
## theta = 20  8.79906 17.16498 5.68871  9.62927 13.45527
```

```
store_res$mean.powerGlobalNull
```

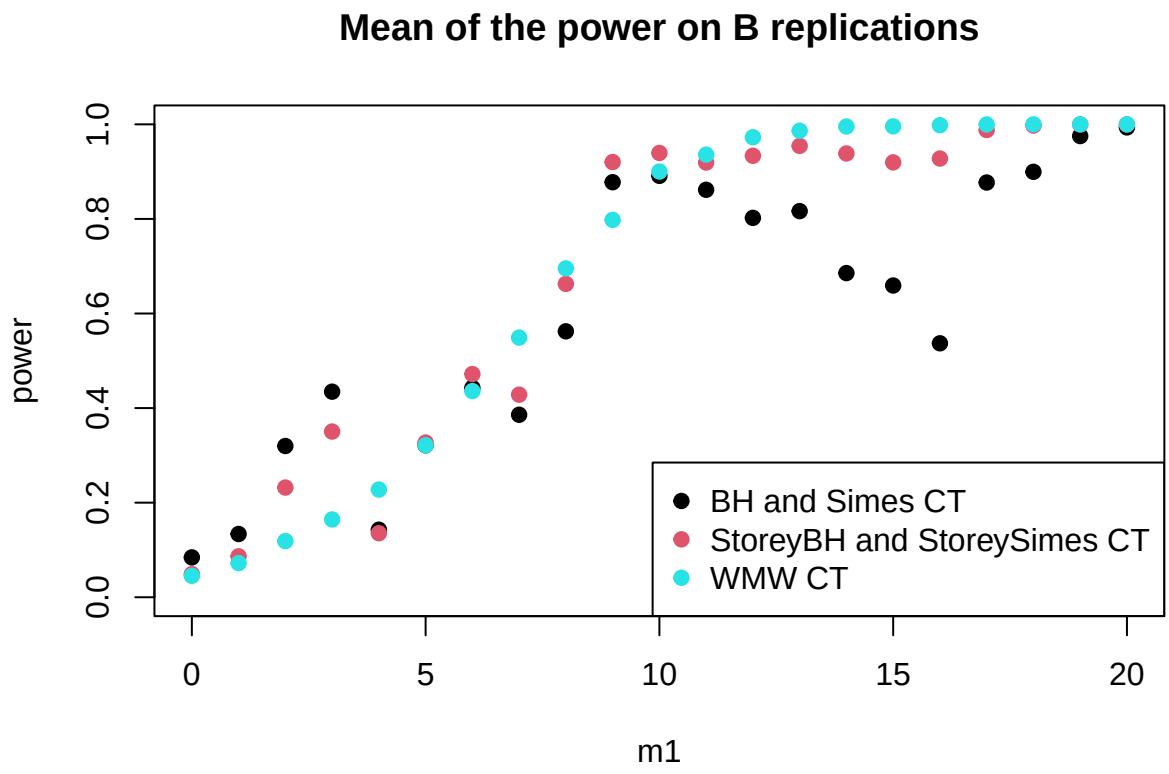
```
##                  BH    StoBH   Simes StoSimes     WMW
```

```
## theta = 0  0.08429 0.04902 0.08429  0.04902 0.04533
## theta = 1  0.13375 0.08658 0.13375  0.08658 0.07215
## theta = 2  0.31981 0.23200 0.31981  0.23200 0.11876
## theta = 3  0.43469 0.35034 0.43469  0.35034 0.16448
## theta = 4  0.14279 0.13537 0.14279  0.13537 0.22770
## theta = 5  0.32149 0.32708 0.32149  0.32708 0.32129
## theta = 6  0.44287 0.47193 0.44287  0.47193 0.43598
## theta = 7  0.38586 0.42835 0.38586  0.42835 0.54906
## theta = 8  0.56231 0.66261 0.56231  0.66261 0.69544
## theta = 9  0.87765 0.92021 0.87765  0.92021 0.79795
## theta = 10 0.89108 0.93950 0.89108  0.93950 0.90019
## theta = 11 0.86162 0.91926 0.86162  0.91926 0.93592
## theta = 12 0.80228 0.93360 0.80228  0.93360 0.97272
## theta = 13 0.81654 0.95445 0.81654  0.95445 0.98652
## theta = 14 0.68547 0.93845 0.68547  0.93845 0.99543
## theta = 15 0.65920 0.91962 0.65920  0.91962 0.99575
## theta = 16 0.53698 0.92772 0.53698  0.92772 0.99823
## theta = 17 0.87697 0.98799 0.87697  0.98799 0.99969
## theta = 18 0.89965 0.99742 0.89965  0.99742 0.99998
## theta = 19 0.97524 0.99984 0.97524  0.99984 1.00000
## theta = 20 0.99354 0.99980 0.99354  0.99980 1.00000
```

```
plot(x = m1s, y = store_res$mean.discov[,1], col = 1, ylab = "d",
     xlab = "m1", ylim=c(0,m), pch=19,
     main = "Mean of the number of discoveries on B replications")
points(x = m1s, y = store_res$mean.discov[,2], col = 2, pch=19)
points(x = m1s, y = store_res$mean.discov[,5], col = 5, pch=19)
legend("topleft", pch = 19, col = c(1,2,5),
       legend =c("BH and Simes CT", "StoreyBH and StoreySimes CT", "WMW CT"))
```




```
plot(x = m1s, y = store_res$mean.powerGlobalNull[,1], col = 1, ylab = "power",
     xlab = "m1", ylim=c(0,1), pch = 19,
     main = "Mean of the power on B replications")
points(x = m1s, y = store_res$mean.powerGlobalNull[,2], col = 2, pch=19)
points(x = m1s, y = store_res$mean.powerGlobalNull[,5], col = 5, pch=19)
legend("bottomright", pch = 19, col = c(1,2,5),
       legend = c("BH and Simes CT", "StoreyBH and StoreySimes CT", "WMW CT"))
```



Pen-Based Recognition of Handwritten Digits dataset

The dataset is available at http://odds.cs.stonybrook.edu/pendigits-dataset.

```
set.seed(321)

# Initializing parameters
B=10^5
n = 199
l = 199
m = 20
alpha = m/(l+1)
m1s = seq(from=0, to=m, by=1)

data = readMat("G:\\Il mio Drive\\PHD\\Progetto di ricerca\\Conformal Inference Project\\Simulazioni\\7
dataset = cbind(data$X, data$y); colnames(dataset)[ncol(dataset)] = "y"
in_ind = which(dataset[,ncol(dataset)]==0)
out_ind = which(dataset[,ncol(dataset)]==1)

tic()
res = lapply(m1s,
             function(m1) sim_realdata(B=B, in_index=in_ind, out_index=out_ind,
                                       dataset=dataset,
                                       alpha=alpha,l=l, n=n, m=m, m1=m1))
toc()
```

```
## 8862.06 sec elapsed
```

```r
# Storing results
store_res = list("mean.discov" = matrix(nrow=length(m1s), ncol = 5),
                 "mean.powerGlobalNull" = matrix(nrow=length(m1s), ncol = 5))
row.names = rep(NA, times=length(m1s))
for(i in 1:length(m1s)){
  row.names[i] = paste("theta =",m1s[i])
}
rownames(store_res$mean.discov) = row.names
colnames(store_res$mean.discov) = c("BH", "StoBH", "Simes", "StoSimes", "WMW")
rownames(store_res$mean.powerGlobalNull) = row.names
colnames(store_res$mean.powerGlobalNull) = c("BH", "StoBH", "Simes", "StoSimes", "WMW")


for(i in 1:length(res)){
  store_res$mean.discov[i,] = res[[i]]$mean.discov
  store_res$mean.powerGlobalNull[i,] = res[[i]]$mean.powerGlobalNull
}

store_res$mean.discov
```

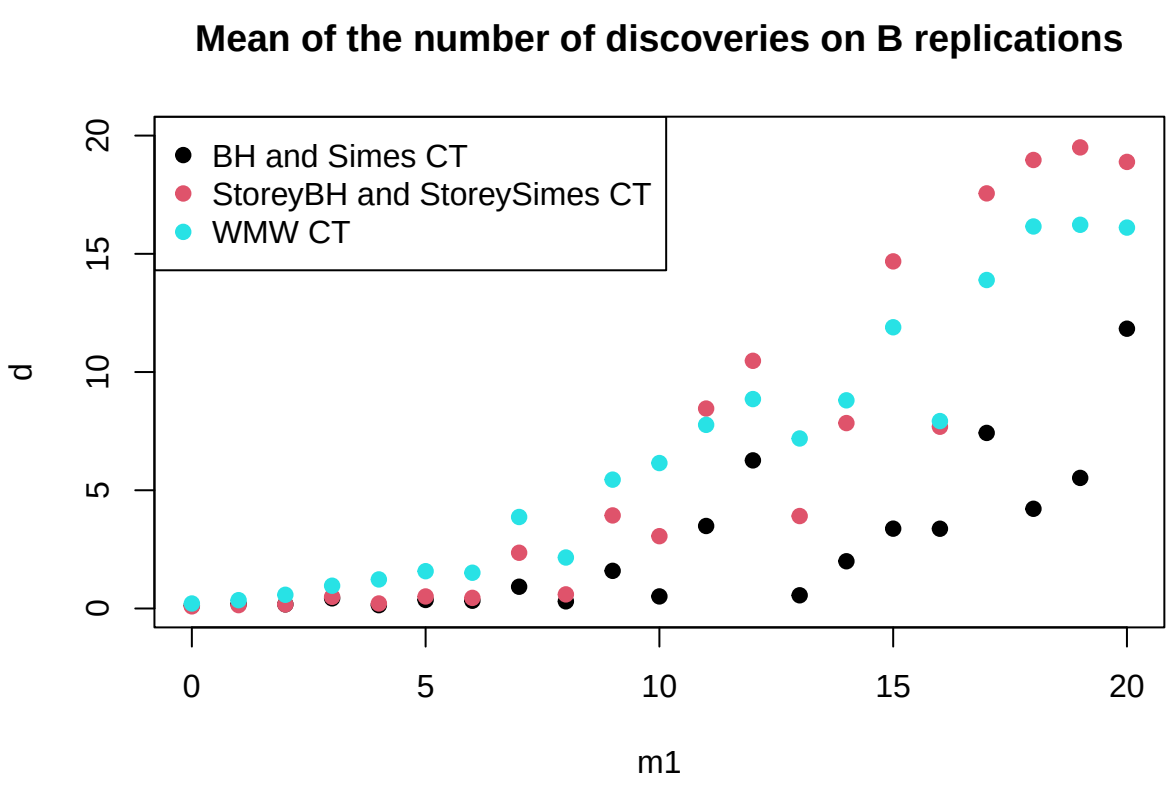
```
##                  BH    StoBH   Simes StoSimes      WMW
## theta = 0   0.12932  0.08958 0.10700  0.06334  0.21082
## theta = 1   0.18665  0.14187 0.15431  0.10147  0.35186
## theta = 2   0.17602  0.17208 0.13451  0.10727  0.57897
## theta = 3   0.42880  0.49287 0.26420  0.25266  0.96226
## theta = 4   0.14072  0.21132 0.11137  0.11909  1.22844
## theta = 5   0.36016  0.50831 0.23719  0.27351  1.57723
## theta = 6   0.32788  0.44811 0.24791  0.27908  1.51256
## theta = 7   0.91966  2.35718 0.45084  0.84753  3.86924
## theta = 8   0.30191  0.59591 0.22095  0.30165  2.15653
## theta = 9   1.59227  3.93623 0.68900  1.47945  5.44914
## theta = 10  0.51058  3.05821 0.27057  0.96599  6.15102
## theta = 11  3.48953  8.45653 1.32734  3.36600  7.76708
## theta = 12  6.26391 10.47452 3.35849  4.95841  8.85653
## theta = 13  0.55839  3.90944 0.37802  1.31868  7.19074
## theta = 14  1.99750  7.84144 1.05017  2.96043  8.80338
## theta = 15  3.37594 14.68081 1.20767  6.21905 11.89389
## theta = 16  3.37333  7.68632 1.48607  3.17675  7.92610
## theta = 17  7.42492 17.55590 3.10216  9.82323 13.89208
## theta = 18  4.21704 18.96820 1.42773 12.65120 16.15721
## theta = 19  5.52298 19.50066 2.96583 12.00037 16.22735
## theta = 20 11.83270 18.88689 7.23118 12.53404 16.10927
```

```r
store_res$mean.powerGlobalNull
```

```
##                  BH   StoBH   Simes StoSimes     WMW
## theta = 0  0.09630 0.05547 0.09630  0.05547 0.10039
## theta = 1  0.13716 0.08702 0.13716  0.08702 0.15786
## theta = 2  0.11939 0.09024 0.11939  0.09024 0.24205
## theta = 3  0.20677 0.18403 0.20677  0.18403 0.37173
## theta = 4  0.10093 0.09920 0.10093  0.09920 0.43789
## theta = 5  0.18624 0.19996 0.18624  0.19996 0.53050
## theta = 6  0.20750 0.21885 0.20750  0.21885 0.50951
## theta = 7  0.30990 0.43360 0.30990  0.43360 0.89335
```

```
## theta = 8  0.18760 0.22399 0.18760  0.22399 0.64818
## theta = 9  0.38469 0.60855 0.38469  0.60855 0.97713
## theta = 10 0.20270 0.43292 0.20270  0.43292 0.98752
## theta = 11 0.55343 0.86904 0.55343  0.86904 0.99956
## theta = 12 0.94923 0.98972 0.94923  0.98972 0.99997
## theta = 13 0.30435 0.57342 0.30435  0.57342 0.99565
## theta = 14 0.58730 0.88370 0.58730  0.88370 0.99987
## theta = 15 0.49185 0.97980 0.49185  0.97980 1.00000
## theta = 16 0.66087 0.86005 0.66087  0.86005 0.99539
## theta = 17 0.81604 0.99983 0.81604  0.99983 1.00000
## theta = 18 0.36641 1.00000 0.36641  1.00000 1.00000
## theta = 19 0.92154 1.00000 0.92154  1.00000 1.00000
## theta = 20 0.98698 1.00000 0.98698  1.00000 1.00000
```

```
plot(x = m1s, y = store_res$mean.discov[,1], col = 1, ylab = "d",
     xlab = "m1", ylim=c(0,m), pch=19,
     main = "Mean of the number of discoveries on B replications")
points(x = m1s, y = store_res$mean.discov[,2], col = 2, pch=19)
points(x = m1s, y = store_res$mean.discov[,5], col = 5, pch=19)
legend("topleft", pch = 19, col = c(1,2,5),
       legend =c("BH and Simes CT", "StoreyBH and StoreySimes CT", "WMW CT"))
```




```
plot(x = m1s, y = store_res$mean.powerGlobalNull[,1], col = 1, ylab = "power",
     xlab = "m1", ylim=c(0,1), pch = 19,
     main = "Mean of the power on B replications")
points(x = m1s, y = store_res$mean.powerGlobalNull[,2], col = 2, pch=19)
points(x = m1s, y = store_res$mean.powerGlobalNull[,5], col = 5, pch=19)
```

```
legend("bottomright", pch = 19, col = c(1,2,5),
       legend = c("BH and Simes CT", "StoreyBH and StoreySimes CT", "WMW CT"))
```