Polygenic Risk Score (PRS) Introduction 501

LD and Concluding Remarks

Drs. Lei Sun, Wei Deng, Yanyan Zhao

Department of Statistical Sciences, FAS

Division of Biostatistics, DLSPH

University of Toronto

11 October, 2021

At the end of this lecture, some basic understanding of our **limited understanding of LD**

- No heterogeneity between ex.data and my.data.
- Only SNPs in perfect LD, $r^2 = 1$, and
- But vary the number tagging SNPs
- No allelic heterogeneity, and
- No multiple causual SNPs within a locus.

First

Live Final Quiz 8

Adding two perfectly tagging SNPs to each of the 10 causal SNP, the AUC of PRS.gw will

A: decrease
B: increase
C: ~same
D: identical

Recall the baseline model without any heterogeneity

10 out 5000 indep. SNPs with **varying 'moderate-large' effects** are truly associated with Y (**all** $\beta = 0.3$ **but MAF vary**).

$$Y_i = \sum_{j=1}^{10} \beta_j G_{ij} + e, \text{ where } \beta_j = 0.3$$

$$\text{MAF} \sim \text{ Unif}(0.05,0.5),\ e \sim N(0,1).$$

```
# external data
ex.nsnp.true=10; ex.beta.true=0.3
ex.nsnp=5000; ex.nsample=1000; ex.sigma=1; ex.seed=101
ex.sumstat=generate.ex.sumstat(ex.seed,ex.nsample,ex.nsnp,ex.nsnp.true,ex.beta.true,ex.sigma)

# my data
my.nsnp.true=10; my.beta.true=0.3; my.maf=ex.sumstat[,"MAF"]
my.nsnp=5000; my.nsample=1000; my.sigma=1; my.seed=102
my.data=generate.my.data(my.seed,my.nsample,my.nsnp,my.nsnp.true,my.beta.true,my.sigma,my.maf)
```

Total and SNP h^2 of the external model

```
## [1] 0.243
```

```
##  [1] 0.023 0.009 0.032 0.031 0.019 0.021 0.029 0.022 0.030 0.028
```

The index of the six 'genome-wide' significant ones

```
## [1]  1  3  4  6  7 10
```

Total and SNP h^2 of my model

```
## [1] 0.243
```

```
##  [1] 0.023 0.009 0.032 0.031 0.019 0.021 0.029 0.022 0.030 0.028
```

Recall the different $PRS_i = \sum_{j=1}^{J} \hat{\beta}_j G_{ij}$

Using the GW threshold on the external data

$$my.PRS_{GW} = \sum_{j=1}^{6} \hat{\beta}_j^{external} \times G_{ij}^{my.data}$$

Using $\alpha = 0.01$ (and also add $\alpha = 0.1$) on the external data

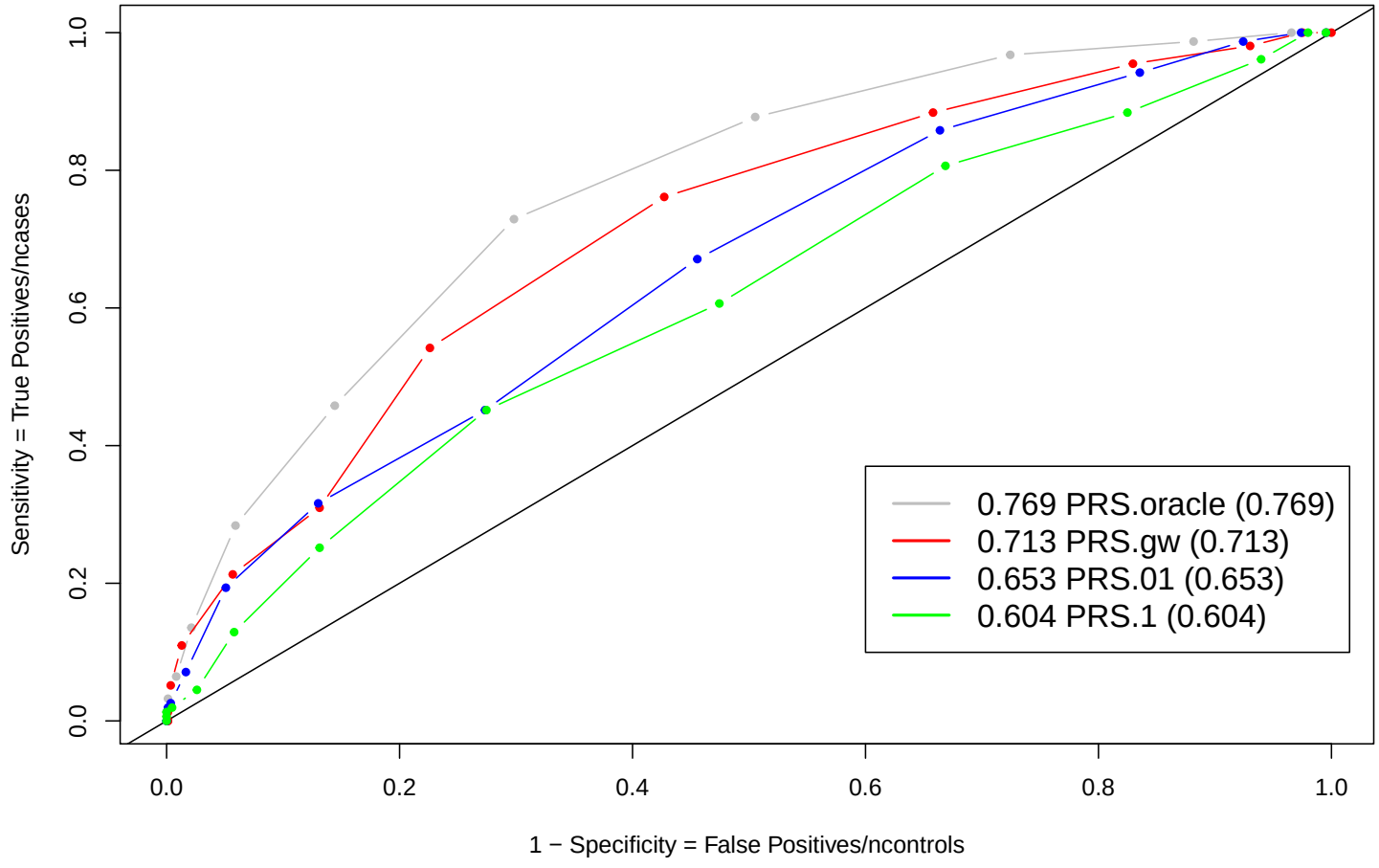
$$my.PRS_{.01} \text{ (or } my.PRS_{.1}) = \sum_{j=1}^{66 \text{ (or } 492)} \hat{\beta}_j^{external} \times G_{ij}^{my.data}$$

The oracle one (benchmarking the upper bound)

$$my.PRS_{oracle} = \sum_{j=1}^{10} 0.3 \times G_{ij}^{my.data}$$

The baseline model ROC and AUC

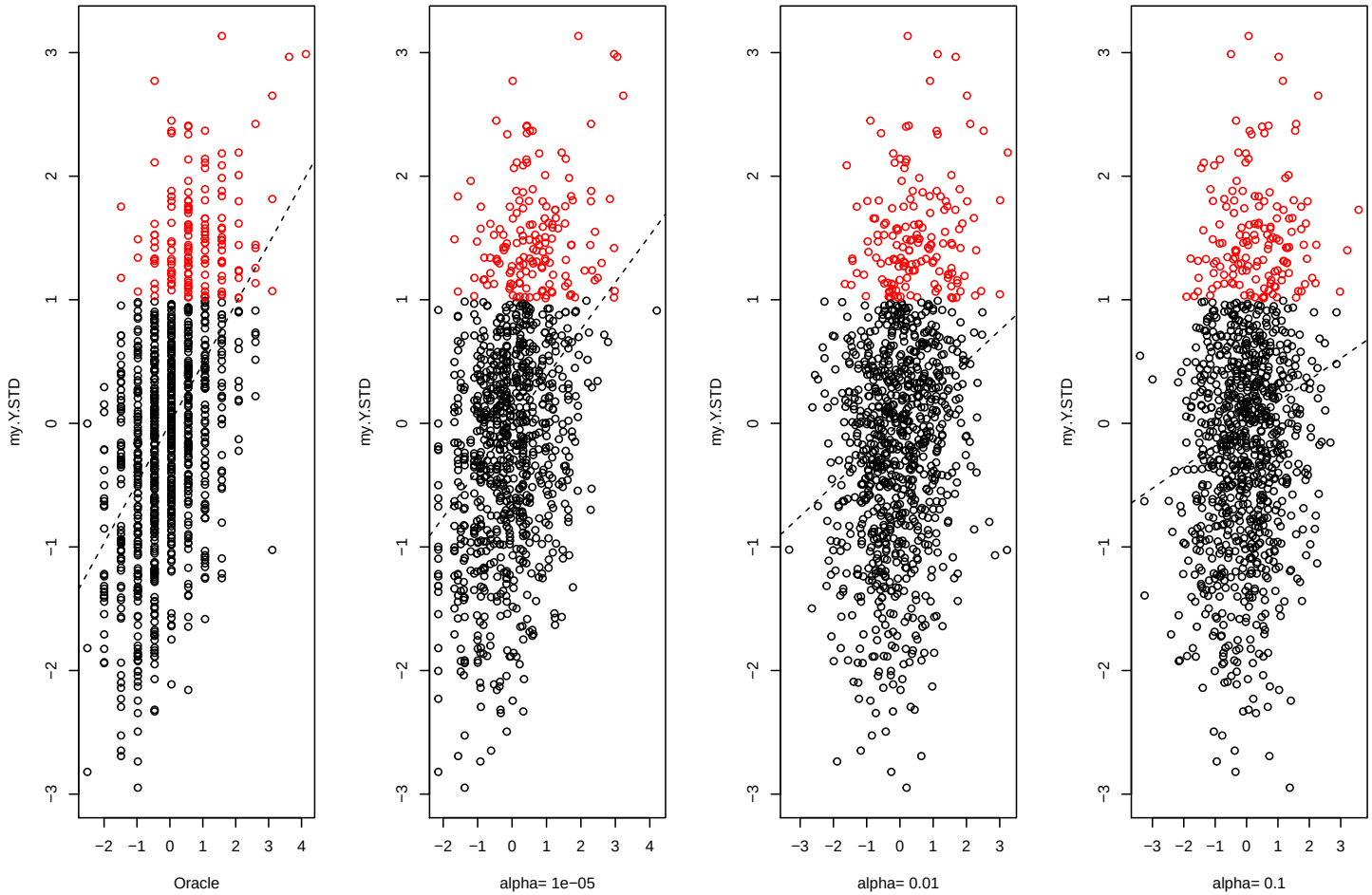
```
# generate the ROC plots
my.PRS.output=generate.my.PRS.output(ex.sumstat,my.data,alpha.level,l.threshold)
generate.ROC.plot(my.PRS.output)
```




```
##       alpha   J TP  FP
## [1,] 1e-05   6  6   0
## [2,] 1e-02  66 10  56
## [3,] 1e-01 492 10 482
```

the baseline model

```
generate.association.plot(my.PRS.output)
```




```
##      [,1]                [,2]          [,3]          [,4]          [,5]
## [1,] "slope.hat"         "0.484"       "0.38"        "0.251"       "0.175"
## [2,] "Z.value"           "17.467"      "12.983"      "8.185"       "5.628"
## [3,] "p.value"           "8.131e-60"   "1.001e-35"   "8.225e-16"   "2.364e-08"
## [4,] "n, case, control"  "1000"        "155"         "845"         " "
```

the baseline model

```
ex.sumstat[1:13,]
```

```
##              MAF MAF.hat beta    beta.hat         se    Z.value      p.value
##  [1,] 0.21748927  0.2215  0.3  0.29257288 0.06536792  4.4757871 8.489445e-06
##  [2,] 0.06972117  0.0610  0.3  0.33145758 0.10935747  3.0309551 2.500692e-03
##  [3,] 0.36935781  0.3780  0.3  0.23908858 0.05323916  4.4908404 7.922031e-06
##  [4,] 0.34596068  0.3480  0.3  0.38889542 0.05565755  6.9872894 5.116550e-12
##  [5,] 0.16243508  0.1695  0.3  0.30892955 0.07052329  4.3805323 1.308960e-05
##  [6,] 0.18502467  0.1995  0.3  0.37606430 0.06503910  5.7821265 9.859505e-09
##  [7,] 0.31318998  0.3375  0.3  0.33166110 0.05410586  6.1298559 1.264930e-09
##  [8,] 0.20006021  0.2020  0.3  0.28159164 0.06670313  4.2215657 2.647447e-05
##  [9,] 0.32990538  0.3360  0.3  0.23025579 0.05661344  4.0671574 5.134017e-05
## [10,] 0.29562285  0.2905  0.3  0.28906539 0.05841261  4.9486810 8.766086e-07
## [11,] 0.44590808  0.4445  0.0  0.09584075 0.05424572  1.7667892 7.756916e-02
## [12,] 0.36809363  0.3745  0.0 -0.02245388 0.05302784 -0.4234356 6.720687e-01
## [13,] 0.37938767  0.3750  0.0 -0.06366768 0.05424574 -1.1736899 2.407993e-01
```

```
my.data$my.sumstat[1:13,]
```

```
##              MAF MAF.hat beta    beta.hat         se    Z.value      p.value
##  [1,] 0.21748927  0.2270  0.3  0.17164714 0.06066349  2.8294965 4.755586e-03
##  [2,] 0.06972117  0.0755  0.3  0.33447059 0.09604226  3.4825358 5.182232e-04
##  [3,] 0.36935781  0.3555  0.3  0.32234988 0.05230483  6.1629085 1.034940e-09
##  [4,] 0.34596068  0.3545  0.3  0.25019642 0.05335395  4.6893703 3.121234e-06
##  [5,] 0.16243508  0.1530  0.3  0.32262395 0.06958963  4.6360925 4.021553e-06
##  [6,] 0.18502467  0.1800  0.3  0.28017270 0.06679596  4.1944557 2.978552e-05
##  [7,] 0.31318998  0.3045  0.3  0.36190034 0.05517189  6.5595060 8.652840e-11
##  [8,] 0.20006021  0.1780  0.3  0.35342514 0.06681630  5.2895046 1.507249e-07
##  [9,] 0.32990538  0.3300  0.3  0.31052039 0.05197947  5.9739043 3.218822e-09
## [10,] 0.29562285  0.2960  0.3  0.33840898 0.05429097  6.2332464 6.731015e-10
## [11,] 0.44590808  0.4465  0.0  0.04515026 0.05043254  0.8952605 3.708637e-01
## [12,] 0.36809363  0.3580  0.0 -0.02127391 0.05509213 -0.3861515 6.994668e-01
## [13,] 0.37938767  0.3870  0.0 -0.02908571 0.05264218 -0.5525171 5.807178e-01
```

Consider LD now

```
# First save the baseline noLD model results for later comparison
ex.nsnp.noLD=ex.nsnp
ex.sumstat.noLD=ex.sumstat
my.data.noLD=my.data
my.nsnp.noLD=my.nsnp
my.PRS.output.noLD=my.PRS.output

# Second, specify the ntag.T for each of the nsnp.true, start with 2 for each
# For now, no null SNPs are tagged, the extreme version of the assumption
# that `truly associated SNPs are more likely to be tagged than null SNPs'

# external data
ex.nsnp.true=10; ex.beta.true=0.3

# specify the ntag.T for each of the nsnp.true
ex.ntag.T=rep(2,ex.nsnp.true)
#ex.ntag.T=c(1,2,3,4,5,6,7,8,9,10)
#ex.ntag.T=c(10,9,8,7,6,5,4,3,2,1)
#ex.ntag.T=c(2,2,1,4,3,10,2,3,7,5)

ex.nsnp=(5000+sum(ex.ntag.T)); ex.nsample=1000; ex.sigma=1; ex.seed=101

# use a new LD-aware data and summary stat function
ex.sumstat=generate.ex.sumstat.LD(ex.seed,ex.nsample,ex.nsnp,ex.nsnp.true,ex.be
```

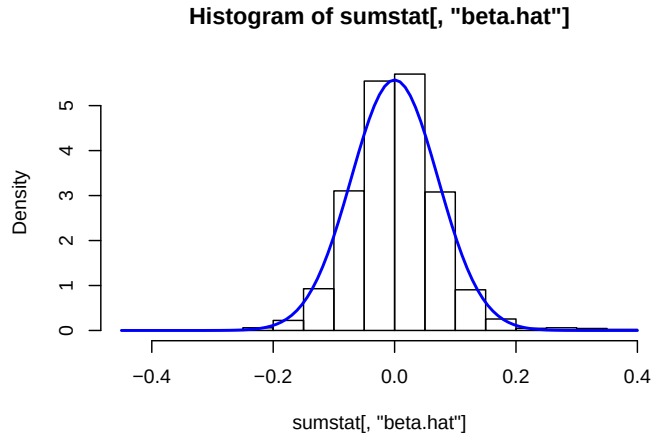
the LD

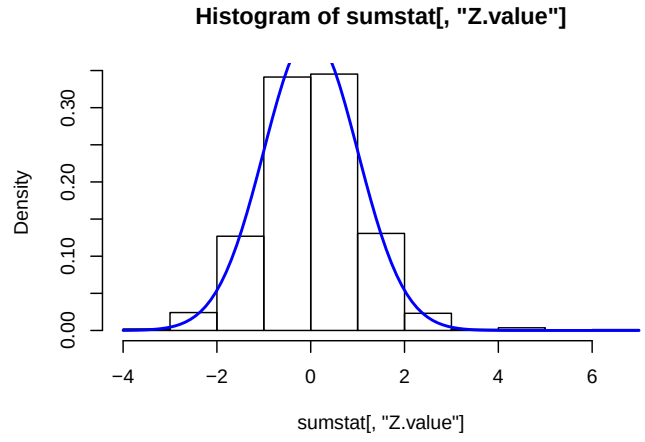
```
ex.sumstat[c(1:12,4999:ex.nsnp),]
```

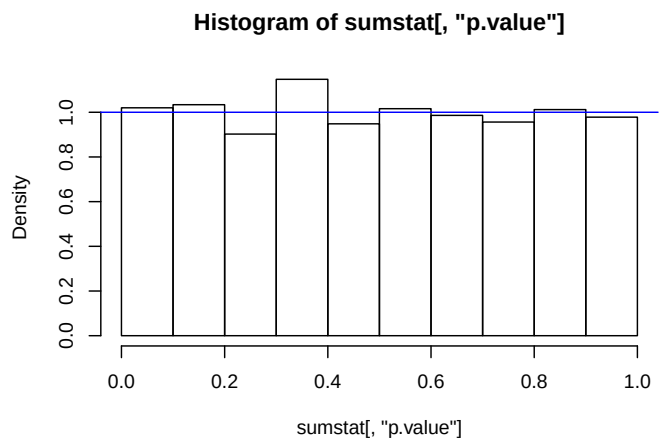
```
##              MAF MAF.hat beta    beta.hat         se    Z.value      p.value
##  [1,] 0.21748927  0.2215  0.3  0.29257288 0.06536792  4.4757871 8.489445e-06
##  [2,] 0.06972117  0.0610  0.3  0.33145758 0.10935747  3.0309551 2.500692e-03
##  [3,] 0.36935781  0.3780  0.3  0.23908858 0.05323916  4.4908404 7.922031e-06
##  [4,] 0.34596068  0.3480  0.3  0.38889542 0.05565755  6.9872894 5.116550e-12
##  [5,] 0.16243508  0.1695  0.3  0.30892955 0.07052329  4.3805323 1.308960e-05
##  [6,] 0.18502467  0.1995  0.3  0.37606430 0.06503910  5.7821265 9.859505e-09
##  [7,] 0.31318998  0.3375  0.3  0.33166110 0.05410586  6.1298559 1.264930e-09
##  [8,] 0.20006021  0.2020  0.3  0.28159164 0.06670313  4.2215657 2.647447e-05
##  [9,] 0.32990538  0.3360  0.3  0.23025579 0.05661344  4.0671574 5.134017e-05
## [10,] 0.29562285  0.2905  0.3  0.28906539 0.05841261  4.9486810 8.766086e-07
## [11,] 0.44590808  0.4445  0.0  0.09584075 0.05424572  1.7667892 7.756916e-02
## [12,] 0.36809363  0.3745  0.0 -0.02245388 0.05302784 -0.4234356 6.720687e-01
## [13,] 0.37662575  0.3720  0.0  0.03703145 0.05527881  0.6699033 5.030744e-01
## [14,] 0.37629499  0.3815  0.0 -0.07125913 0.05474273 -1.3017095 1.933161e-01
## [15,] 0.21748927  0.2215  0.0  0.29257288 0.06536792  4.4757871 8.489445e-06
## [16,] 0.21748927  0.2215  0.0  0.29257288 0.06536792  4.4757871 8.489445e-06
## [17,] 0.06972117  0.0610  0.0  0.33145758 0.10935747  3.0309551 2.500692e-03
## [18,] 0.06972117  0.0610  0.0  0.33145758 0.10935747  3.0309551 2.500692e-03
## [19,] 0.36935781  0.3780  0.0  0.23908858 0.05323916  4.4908404 7.922031e-06
## [20,] 0.36935781  0.3780  0.0  0.23908858 0.05323916  4.4908404 7.922031e-06
## [21,] 0.34596068  0.3480  0.0  0.38889542 0.05565755  6.9872894 5.116550e-12
## [22,] 0.34596068  0.3480  0.0  0.38889542 0.05565755  6.9872894 5.116550e-12
## [23,] 0.16243508  0.1695  0.0  0.30892955 0.07052329  4.3805323 1.308960e-05
## [24,] 0.16243508  0.1695  0.0  0.30892955 0.07052329  4.3805323 1.308960e-05
## [25,] 0.18502467  0.1995  0.0  0.37606430 0.06503910  5.7821265 9.859505e-09
## [26,] 0.18502467  0.1995  0.0  0.37606430 0.06503910  5.7821265 9.859505e-09
## [27,] 0.31318998  0.3375  0.0  0.33166110 0.05410586  6.1298559 1.264930e-09
## [28,] 0.31318998  0.3375  0.0  0.33166110 0.05410586  6.1298559 1.264930e-09
## [29,] 0.20006021  0.2020  0.0  0.28159164 0.06670313  4.2215657 2.647447e-05
## [30,] 0.20006021  0.2020  0.0  0.28159164 0.06670313  4.2215657 2.647447e-05
## [31,] 0.32990538  0.3360  0.0  0.23025579 0.05661344  4.0671574 5.134017e-05
## [32,] 0.32990538  0.3360  0.0  0.23025579 0.05661344  4.0671574 5.134017e-05
## [33,] 0.29562285  0.2905  0.0  0.28906539 0.05841261  4.9486810 8.766086e-07
```

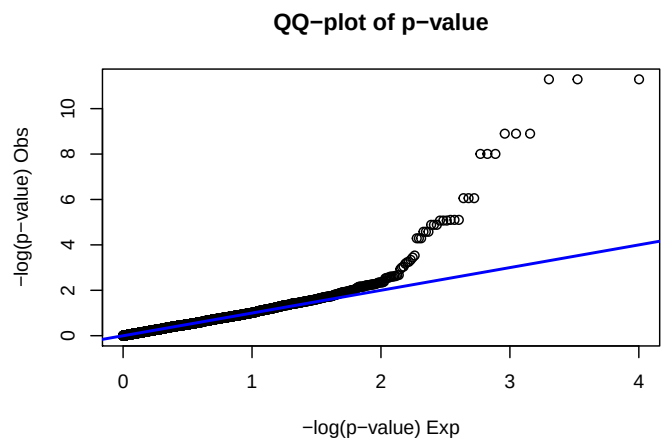
the LD ex.model, ntag.T=2

```
generate.sumstat.plot(ex.sumstat,ex.nsnp)
```



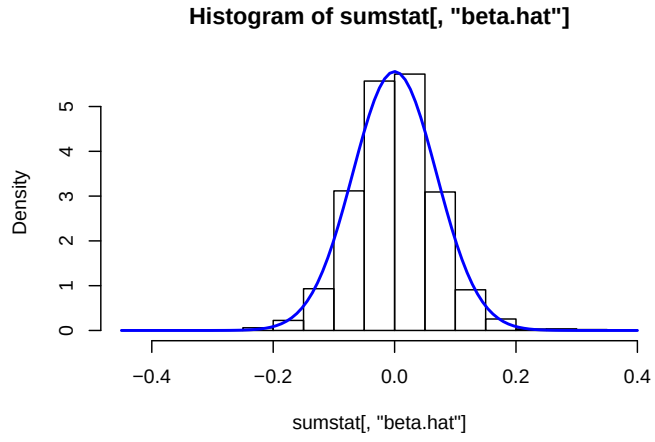


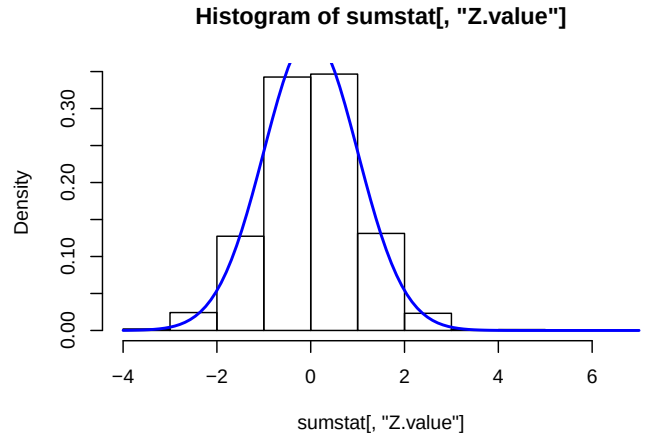


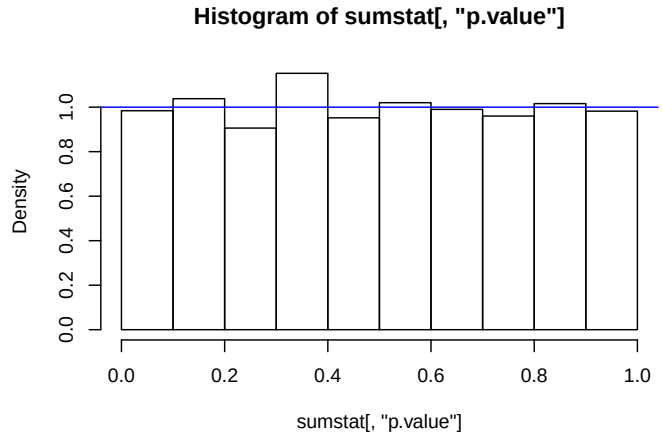


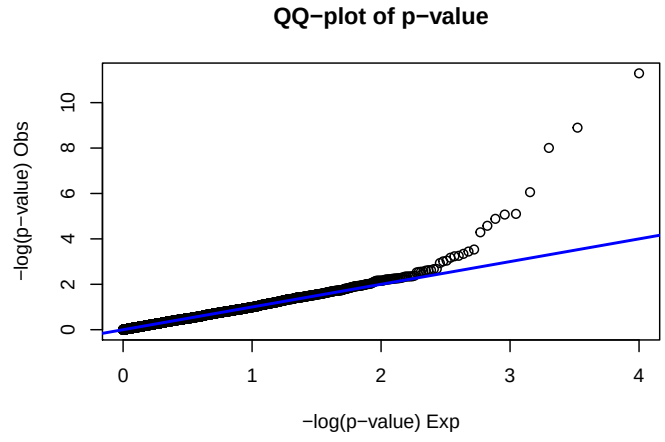
compare with no tagging ex.model

```
generate.sumstat.plot(ex.sumstat.noLD,ex.nsnp.noLD)
```









The true h^2 of our LD model should stay the same, because $\beta = 0$ for all the tagging SNPs.

GWAS-based estimates may be a different story!

Total and SNP h^2 of the ex.model

```
# the trait h2
V.G=sum(ex.sumstat[1:ex.nsnp.true,"beta"]^2*(2*ex.sumstat[1:ex.nsnp.true,"MAF"]*(1-ex.sumstat[1:ex.nsnp.tr
V.e=ex.sigma^2
round(V.G/(V.G+V.e),3)
```

```
## [1] 0.243
```

```
# the causal ones
V.G.loci=ex.sumstat[1:ex.nsnp.true,"beta"]^2*(2*ex.sumstat[1:ex.nsnp.true,"MAF"]*(1-ex.sumstat[1:ex.nsnp.t
round(V.G.loci/(V.G+V.e),3)
```

```
##  [1] 0.023 0.009 0.032 0.031 0.019 0.021 0.029 0.022 0.030 0.028
```

```
# the tagging ones
V.G.loci=ex.sumstat[(ex.nsnp-sum(ex.ntag.T)+1):ex.nsnp,"beta"]^2*(2*ex.sumstat[(ex.nsnp-sum(ex.ntag.T)+1)
round(V.G.loci/(V.G+V.e),3)
```

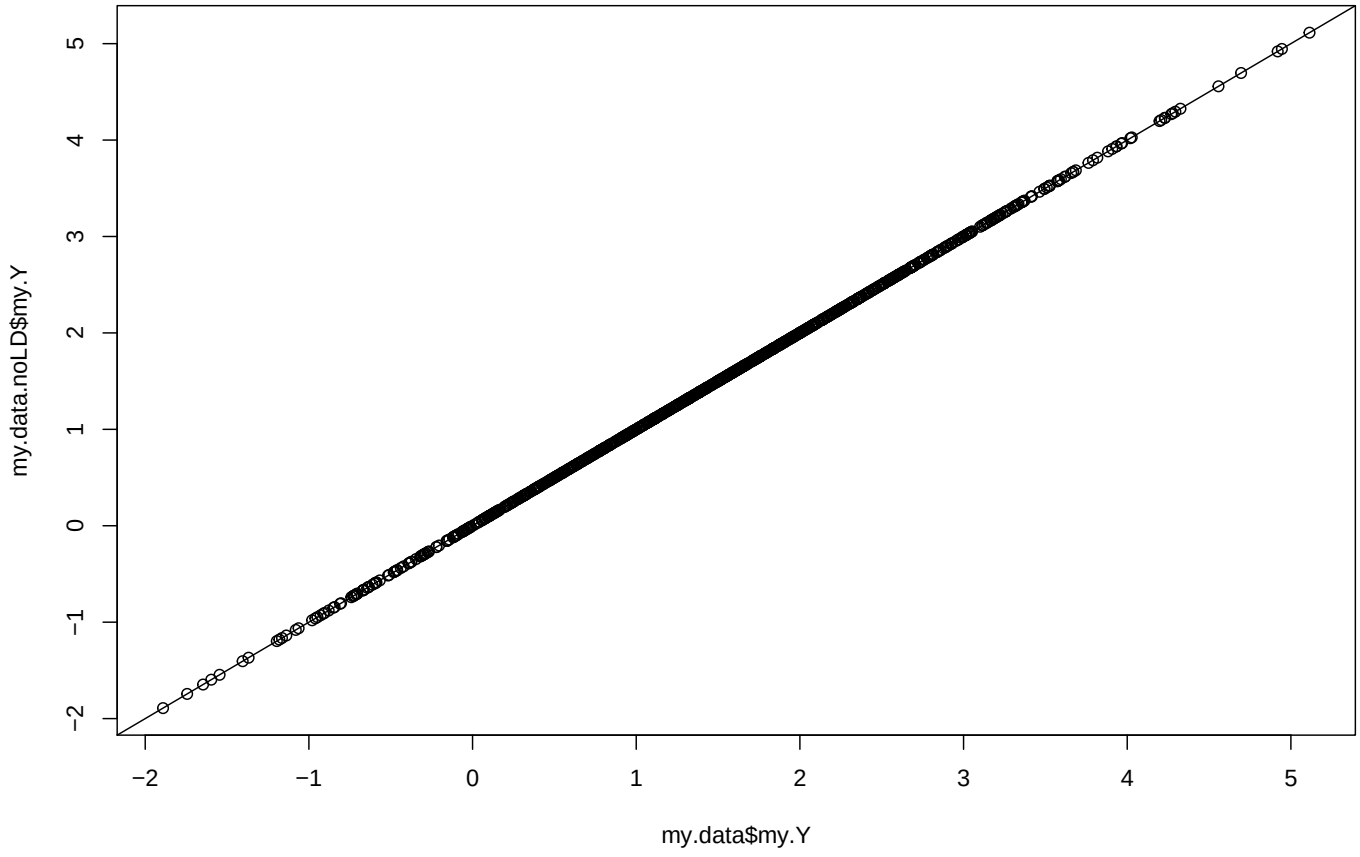
```
##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Move to my.data now

```
# my data

my.ntag.T=ex.ntag.T # SAME tagging; no heterogeneity between my. and ex.

my.nsnp.true=10; my.beta.true=0.3; my.maf=ex.sumstat[,"MAF"]
my.nsnp=(5000+sum(my.ntag.T)); my.nsample=1000; my.sigma=1; my.seed=102

my.data=generate.my.data.LD(my.seed,my.nsample,my.nsnp,my.nsnp.true,my.beta.true,my.sigma,my.maf,my.ntag.T
```

Checking our data: my.Y = my.Y.noLD (given the same seed etc.)

```
plot(my.data$my.Y,my.data.noLD$my.Y); abline(0,1)
```



Tagging SNPs have large GWAS Z and small p-values, but their true $\beta = 0$. Thus, they do not affect how we generate Y based on the true model.

(Multiple causal SNPs in LD is a different story!)

the LD

```
my.data$my.sumstat[c(1:12,4999:my.nsnp),]
```

```
##              MAF MAF.hat beta    beta.hat         se    Z.value      p.value
##  [1,] 0.21748927  0.2270  0.3  0.17164714 0.06066349  2.8294965 4.755586e-03
##  [2,] 0.06972117  0.0755  0.3  0.33447059 0.09604226  3.4825358 5.182232e-04
##  [3,] 0.36935781  0.3555  0.3  0.32234988 0.05230483  6.1629085 1.034940e-09
##  [4,] 0.34596068  0.3545  0.3  0.25019642 0.05335395  4.6893703 3.121234e-06
##  [5,] 0.16243508  0.1530  0.3  0.32262395 0.06958963  4.6360925 4.021553e-06
##  [6,] 0.18502467  0.1800  0.3  0.28017270 0.06679596  4.1944557 2.978552e-05
##  [7,] 0.31318998  0.3045  0.3  0.36190034 0.05517189  6.5595060 8.652840e-11
##  [8,] 0.20006021  0.1780  0.3  0.35342514 0.06681630  5.2895046 1.507249e-07
##  [9,] 0.32990538  0.3300  0.3  0.31052039 0.05197947  5.9739043 3.218822e-09
## [10,] 0.29562285  0.2960  0.3  0.33840898 0.05429097  6.2332464 6.731015e-10
## [11,] 0.44590808  0.4465  0.0  0.04515026 0.05043254  0.8952605 3.708637e-01
## [12,] 0.36809363  0.3580  0.0 -0.02127391 0.05509213 -0.3861515 6.994668e-01
## [13,] 0.37662575  0.3820  0.0 -0.02435158 0.05358273 -0.4544670 6.495915e-01
## [14,] 0.37629499  0.3780  0.0  0.01024406 0.05300796  0.1932551 8.467985e-01
## [15,] 0.21748927  0.3780  0.0  0.17164714 0.06066349  2.8294965 4.755586e-03
## [16,] 0.21748927  0.3780  0.0  0.17164714 0.06066349  2.8294965 4.755586e-03
## [17,] 0.06972117  0.3780  0.0  0.33447059 0.09604226  3.4825358 5.182232e-04
## [18,] 0.06972117  0.3780  0.0  0.33447059 0.09604226  3.4825358 5.182232e-04
## [19,] 0.36935781  0.3780  0.0  0.32234988 0.05230483  6.1629085 1.034940e-09
## [20,] 0.36935781  0.3780  0.0  0.32234988 0.05230483  6.1629085 1.034940e-09
## [21,] 0.34596068  0.3780  0.0  0.25019642 0.05335395  4.6893703 3.121234e-06
## [22,] 0.34596068  0.3780  0.0  0.25019642 0.05335395  4.6893703 3.121234e-06
## [23,] 0.16243508  0.3780  0.0  0.32262395 0.06958963  4.6360925 4.021553e-06
## [24,] 0.16243508  0.3780  0.0  0.32262395 0.06958963  4.6360925 4.021553e-06
## [25,] 0.18502467  0.3780  0.0  0.28017270 0.06679596  4.1944557 2.978552e-05
## [26,] 0.18502467  0.3780  0.0  0.28017270 0.06679596  4.1944557 2.978552e-05
## [27,] 0.31318998  0.3780  0.0  0.36190034 0.05517189  6.5595060 8.652840e-11
## [28,] 0.31318998  0.3780  0.0  0.36190034 0.05517189  6.5595060 8.652840e-11
## [29,] 0.20006021  0.3780  0.0  0.35342514 0.06681630  5.2895046 1.507249e-07
## [30,] 0.20006021  0.3780  0.0  0.35342514 0.06681630  5.2895046 1.507249e-07
## [31,] 0.32990538  0.3780  0.0  0.31052039 0.05197947  5.9739043 3.218822e-09
## [32,] 0.32990538  0.3780  0.0  0.31052039 0.05197947  5.9739043 3.218822e-09
## [33,] 0.29562285  0.3780  0.0  0.33840898 0.05429097  6.2332464 6.731015e-10
```

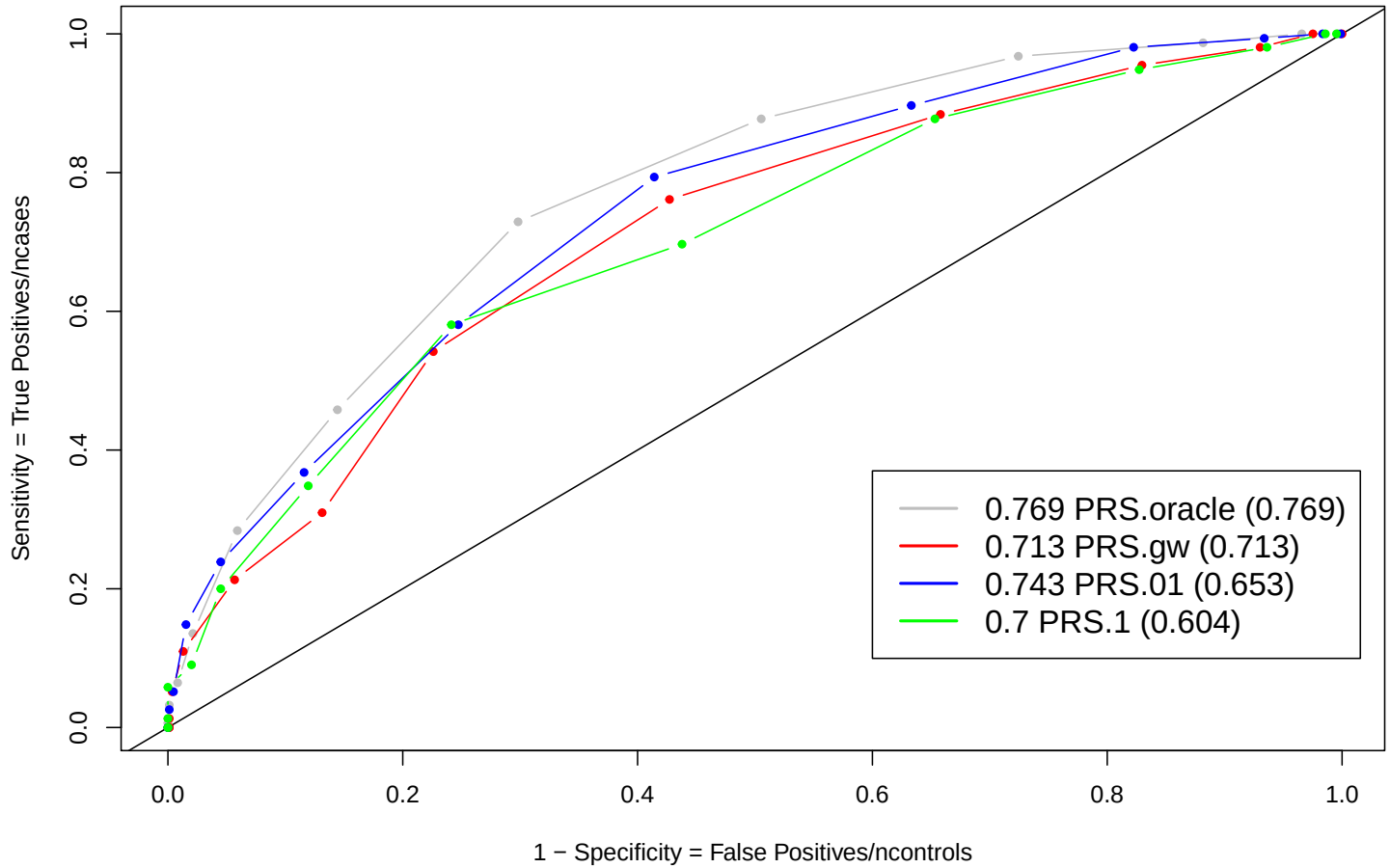
Move to PRS now

We can use the same PRS functions, as PRS construction only depend on ex.sumstat that, which were already generated and checked.

```
# Even the ex.nsnp is slightly bigger now, use the same alpha due to LD
alpha.level=c(0.00001,0.01,0.1)
l.threshold=1
```

the LD my.model, ntag.T=2

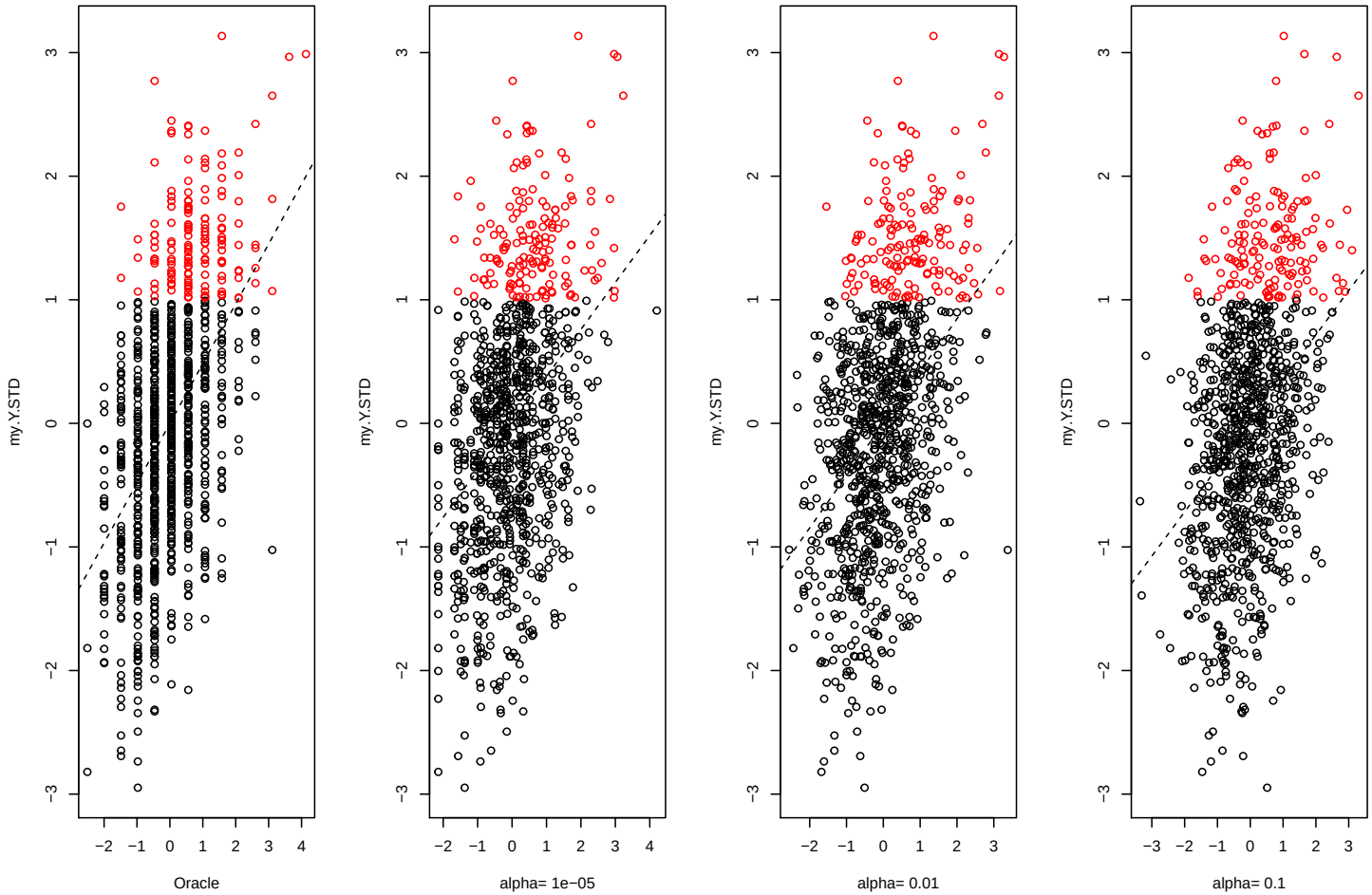
```
# generate the ROC plots
my.PRS.output=generate.my.PRS.output(ex.sumstat,my.data,alpha.level,l.threshold)
generate.ROC.plot(my.PRS.output)
```




```
##       alpha   J TP  FP
## [1,] 1e-05  18  6  12
## [2,] 1e-02  86 10  76
## [3,] 1e-01 512 10 502
```

the LD my.model, ntag.T=2

```
generate.association.plot(my.PRS.output)
```




```
##      [,1]                [,2]        [,3]         [,4]         [,5]
## [1,] "slope.hat"         "0.484"     "0.38"       "0.423"      "0.358"
## [2,] "Z.value"           "17.467"    "12.983"     "14.748"     "12.116"
## [3,] "p.value"           "8.131e-60" "1.001e-35" "1.111e-44" "1.27e-31"
## [4,] "n, case, control" "1000"      "155"        "845"        " "
```

Did we make a mistake?! Recall

$$PRS_i^{my.data} = \sum_{j=1}^{J} \hat{\beta}_j^{external} G_{ij}^{my.data}$$

When $r^2 = 1$ and $ntag.T = 2$ for all the causal SNPS:

$$PRS_i^{my.LD} = \sum_{j=1}^{J_{LD}} \hat{\beta}_j^{ex.LD} G_{ij}^{my.LD}$$

$$= 3 \times \sum_{j=1;TP}^{J_{noLD}} \hat{\beta}_j^{ex.noLD} G_{ij}^{my.noLD} + \sum_{j=1;FP}^{J_{noLD}} \hat{\beta}_j^{ex.noLD} G_{ij}^{my.noLD}$$

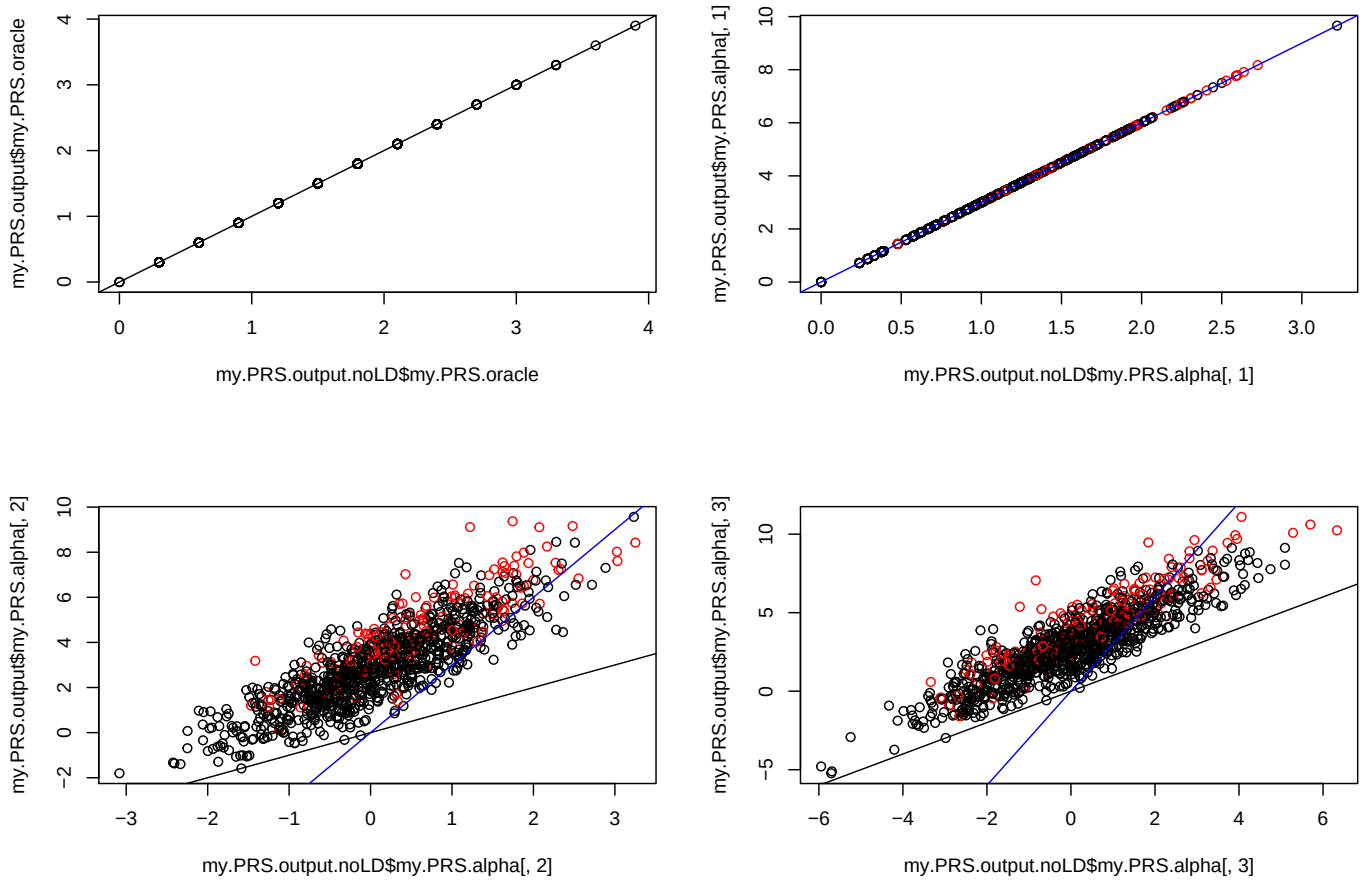
When FP=0

$$PRS_i^{my.LD} = 3 \times PRS_i^{my.noLD}$$

PRS.oracle should stay the same

$$PRS_{i,oracle}^{my.LD} = \sum_{j=1}^{J=10} \beta_j \cdot G_{ij}^{my.LD} = \sum_{j=1}^{J=10} \beta_j \cdot G_{ij}^{my.noLD} = PRS_{i,oracle}^{my.noLD}, \text{ where } \beta_j = 0.3$$

Check the NON-STD PRS data



In this simplest setting ($r^2 = 1$ and only causal SNPs are tagged), NOT adjusting LD '=' leveraging LD to improve performance!

Could consider different types of ntag.T, e.g.

```
# only one tagging SNP
# ex.ntag.T=rep(1,ex.nsnp.true)

# heterogeneity in the number of tagging SNPs
# ex.ntag.T=c(1,2,3,4,5,6,7,8,9,10)
# ex.ntag.T=c(10,9,8,7,6,5,4,3,2,1)
# ex.ntag.T=c(2,2,1,4,3,10,2,3,7,5)

# my.ntag.T=ex.ntag.T # SAME tagging; no heterogeneity between my. and ex.
```

Results are not drastically different, as expected, e.g.

```
# external data
ex.nsnp.true=10; ex.beta.true=0.3
# specify the ntag.T for each of the nsnp.true
ex.ntag.T=c(2,2,1,4,3,10,2,3,7,5)
ex.nsnp=(5000+sum(ex.ntag.T)); ex.nsample=1000; ex.sigma=1; ex.seed=101
ex.sumstat=generate.ex.sumstat.LD(ex.seed,ex.nsample,ex.nsnp,ex.nsnp.true,ex.beta.true,ex.sigma,ex.ntag.T)

# my data
my.ntag.T=ex.ntag.T # SAME tagging; no heterogeneity between my. and ex.
my.nsnp.true=10; my.beta.true=0.3; my.maf=ex.sumstat[,"MAF"]
my.nsnp=(5000+sum(my.ntag.T)); my.nsample=1000; my.sigma=1; my.seed=102
my.data=generate.my.data.LD(my.seed,my.nsample,my.nsnp,my.nsnp.true,my.beta.true,my.sigma,my.maf,my.ntag.T
```

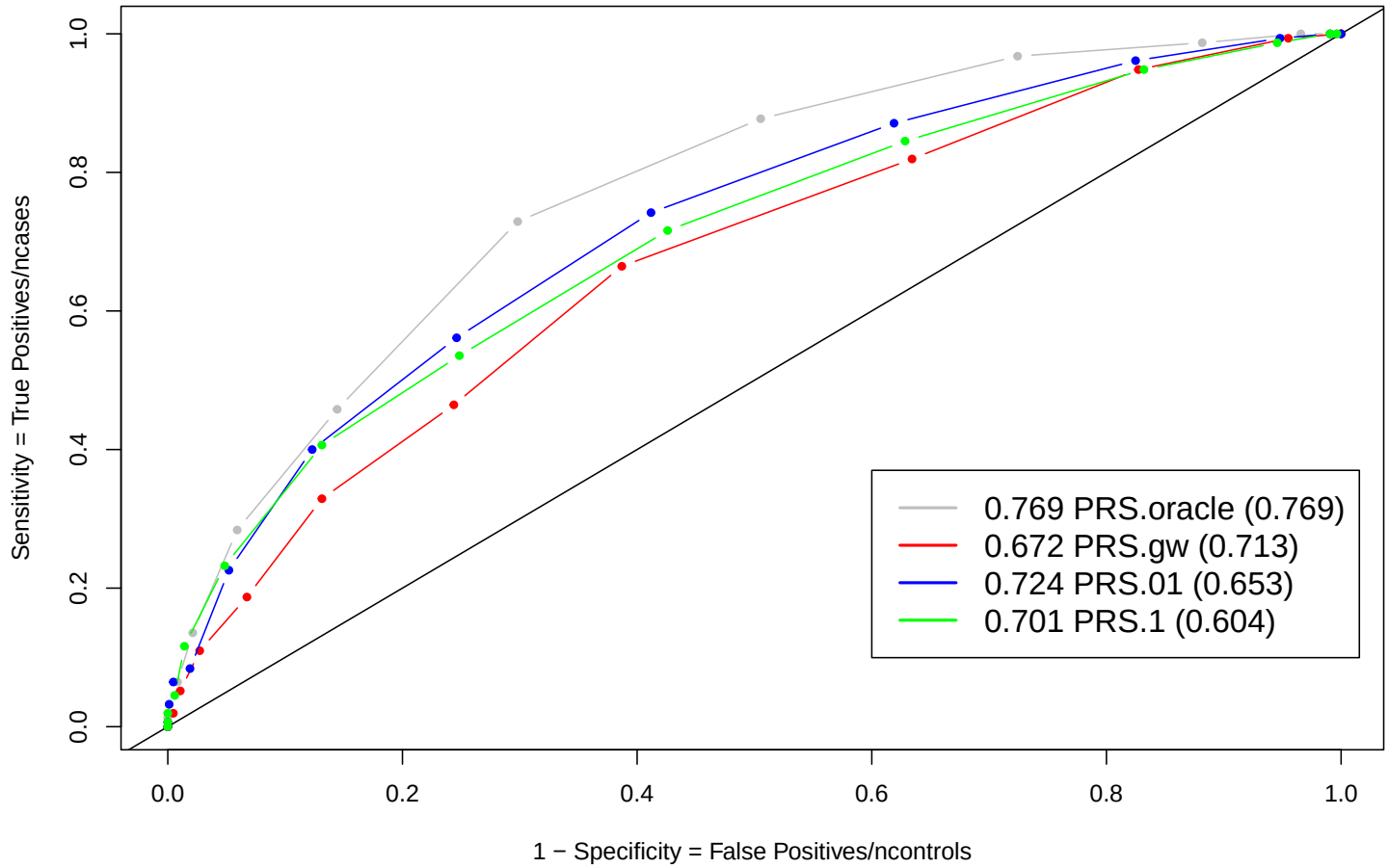
the LD

```
my.data$my.sumstat[c(1:12,4999:my.nsnp),]
```

```
##              MAF MAF.hat beta    beta.hat         se    Z.value      p.value
##  [1,] 0.21748927  0.2270  0.3  0.17164714 0.06066349  2.8294965 4.755586e-03
##  [2,] 0.06972117  0.0755  0.3  0.33447059 0.09604226  3.4825358 5.182232e-04
##  [3,] 0.36935781  0.3555  0.3  0.32234988 0.05230483  6.1629085 1.034940e-09
##  [4,] 0.34596068  0.3545  0.3  0.25019642 0.05335395  4.6893703 3.121234e-06
##  [5,] 0.16243508  0.1530  0.3  0.32262395 0.06958963  4.6360925 4.021553e-06
##  [6,] 0.18502467  0.1800  0.3  0.28017270 0.06679596  4.1944557 2.978552e-05
##  [7,] 0.31318998  0.3045  0.3  0.36190034 0.05517189  6.5595060 8.652840e-11
##  [8,] 0.20006021  0.1780  0.3  0.35342514 0.06681630  5.2895046 1.507249e-07
##  [9,] 0.32990538  0.3300  0.3  0.31052039 0.05197947  5.9739043 3.218822e-09
## [10,] 0.29562285  0.2960  0.3  0.33840898 0.05429097  6.2332464 6.731015e-10
## [11,] 0.44590808  0.4465  0.0  0.04515026 0.05043254  0.8952605 3.708637e-01
## [12,] 0.36809363  0.3580  0.0 -0.02127391 0.05509213 -0.3861515 6.994668e-01
## [13,] 0.37662575  0.3820  0.0 -0.02435158 0.05358273 -0.4544670 6.495915e-01
## [14,] 0.37629499  0.3780  0.0  0.01024406 0.05300796  0.1932551 8.467985e-01
## [15,] 0.21748927  0.3780  0.0  0.17164714 0.06066349  2.8294965 4.755586e-03
## [16,] 0.21748927  0.3780  0.0  0.17164714 0.06066349  2.8294965 4.755586e-03
## [17,] 0.06972117  0.3780  0.0  0.33447059 0.09604226  3.4825358 5.182232e-04
## [18,] 0.06972117  0.3780  0.0  0.33447059 0.09604226  3.4825358 5.182232e-04
## [19,] 0.36935781  0.3780  0.0  0.32234988 0.05230483  6.1629085 1.034940e-09
## [20,] 0.34596068  0.3780  0.0  0.25019642 0.05335395  4.6893703 3.121234e-06
## [21,] 0.34596068  0.3780  0.0  0.25019642 0.05335395  4.6893703 3.121234e-06
## [22,] 0.34596068  0.3780  0.0  0.25019642 0.05335395  4.6893703 3.121234e-06
## [23,] 0.34596068  0.3780  0.0  0.25019642 0.05335395  4.6893703 3.121234e-06
## [24,] 0.16243508  0.3780  0.0  0.32262395 0.06958963  4.6360925 4.021553e-06
## [25,] 0.16243508  0.3780  0.0  0.32262395 0.06958963  4.6360925 4.021553e-06
## [26,] 0.16243508  0.3780  0.0  0.32262395 0.06958963  4.6360925 4.021553e-06
## [27,] 0.18502467  0.3780  0.0  0.28017270 0.06679596  4.1944557 2.978552e-05
## [28,] 0.18502467  0.3780  0.0  0.28017270 0.06679596  4.1944557 2.978552e-05
## [29,] 0.18502467  0.3780  0.0  0.28017270 0.06679596  4.1944557 2.978552e-05
## [30,] 0.18502467  0.3780  0.0  0.28017270 0.06679596  4.1944557 2.978552e-05
## [31,] 0.18502467  0.3780  0.0  0.28017270 0.06679596  4.1944557 2.978552e-05
## [32,] 0.18502467  0.3780  0.0  0.28017270 0.06679596  4.1944557 2.978552e-05
## [33,] 0.18502467  0.3780  0.0  0.28017270 0.06679596  4.1944557 2.978552e-05
```

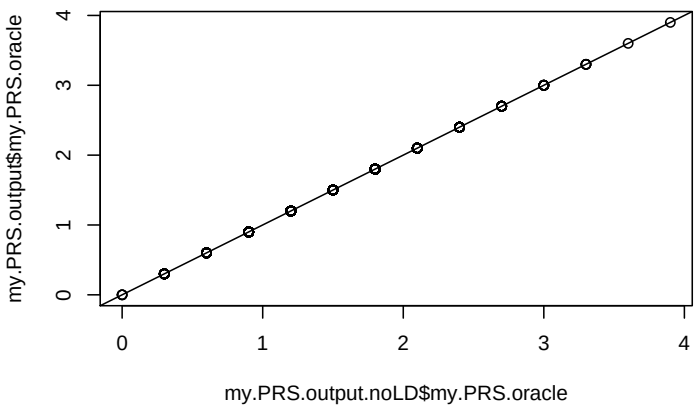
the LD my.model, ntag.T=c(2,2,1,4,3,10,2,3,7,5)

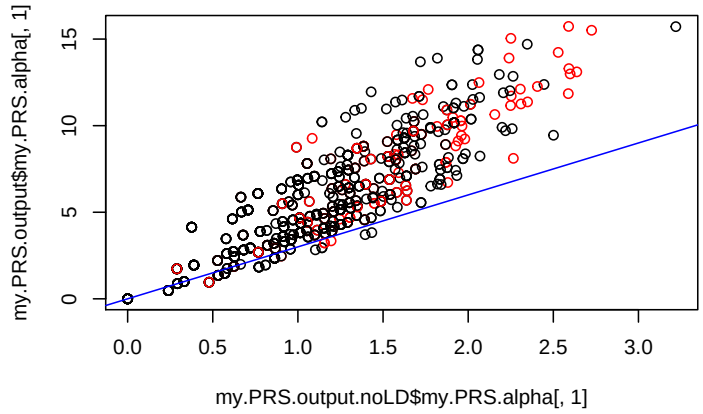
```
# generate the ROC plots
my.PRS.output=generate.my.PRS.output(ex.sumstat,my.data,alpha.level,l.threshold)
generate.ROC.plot(my.PRS.output)
```

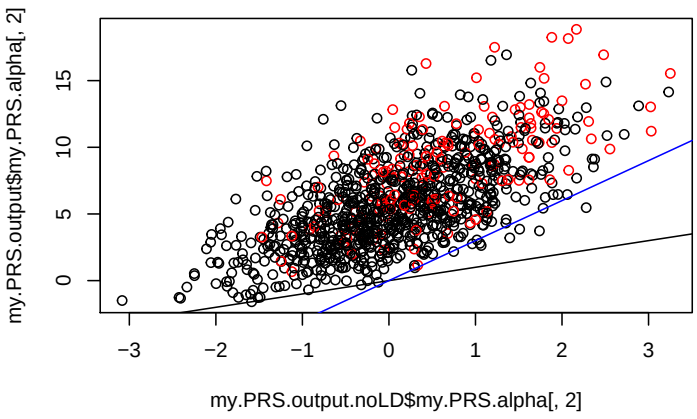


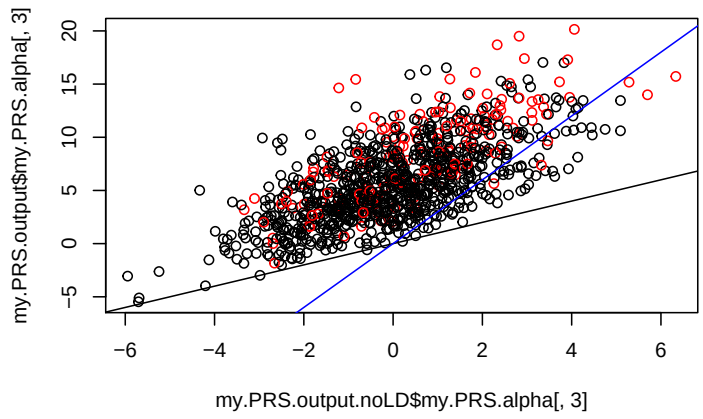

```
##      alpha   J TP  FP
## [1,] 1e-05  30  6  24
## [2,] 1e-02 105 10  95
## [3,] 1e-01 531 10 521
```

the LD my.model, ntag.T=c(2,2,1,4,3,10,2,3,7,5)









The effect of adding ntag.F for null SNPs?

To do... Educated guess:

- AUC stays the same if ntag.T $=$ ntag.F
- AUC drops if ntag.T $\neq$ ntag.F
- AUC drops more significantly if ntag.T $<$ ntag.F
- AUC drops more significantly for less stringent alpha

Some models/methods not discussed

- GxG and GxE interactions
- Bayesian methods
- Rare variants
- The X chromosome
- Pitfalls of standardization (how do we define a case in practice?)
- Many more...

Recap of the learning goal: a **deeper** understanding of

1. **PRS foundation: GWAS, h2 and prediction**
   - the multiple hypothesis testing issue inherent in GWAS
   - the (high) variability inherent in the h2 estimates
   - h2 as a function of both genetic effect beta and MAF
   - the 'genetic effect size' of a SNP as a function of beta and MAF
   - a conceptual PRS construction based on the ground truth, PRS.oracle
   - DIY ROC plotting and AUC calculation for a PRS-based prediction

2. **PRS basic: PRS calculation and performance evaluation**
   - the complexity of constructing a good PRS even under the simplest setting without LD or any heterogeneities; 10 out 5000 independent SNPs are truly associated with the same effect size of 0.3 but varying MAFs.
   - the trouble introduced by false positives, due to multiple hypothesis testing and low power.
   - 'the more is not always better' statement: PRS based on 6 'genome-wide' significant SNPs vs. 66 SNPs significant at 0.01.
   - the various over-fitting or selection biases, and winner's curse in beta estimates for both false positives and true positives.

Learning goal Cont'd, a **deeper** understanding of

3. **PRS basic-plus: some obvious or not so obvious follow-up Qs**
   - Effects of ex.nsample and ex.beta.true on AUC: easy to answer.
   - Answers to these Qs are less obvious: **If we decrease ex.beta.true from 0.3 to 0.1 but increase ex.nsnp.true from 10 to 90**,
     h^2 and SNP h^2?
     AUC in general?
     AUC between PRS.gw and PRS.01?

4. **PRS heterogeneity and transportability**
   - First, why reference allele (genome build) matching is so consequential
   - Then, population and locus heterogeneity including
     my.maf $\neq$ ex.maf
     my.beta.true $\neq$ ex.beta.true
     my.nsnp.true $\neq$ ex.nsnp.true

5. **PRS LD consideration**
   - Some basic understanding of our **limited understanding of LD**.

End of the (hopefully fun) ride!