Polygenic Risk Score (PRS) Introduction 301

basics-plus, some obvious or not so obvious follow-up Qs

Drs. Lei Sun, Wei Deng, Yanyan Zhao

Department of Statistical Sciences, FAS

Division of Biostatistics, DLSPH

University of Toronto

10 October, 2021

At the end of this lecture, a **deeper** understanding of

- 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?

Recall the illustrative 'polygenic' model simulation study

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

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

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

```
# now name it clearly as the summary statistics from the external data
ex.nsample=1000;ex.nsnp=5000;ex.nsnp.true=10;ex.beta.true=0.3;ex.sigma=1
ex.seed=101
ex.sumstat=generate.ex.sumstat(ex.seed,ex.nsample,ex.nsnp,ex.nsnp.true,ex.beta.
```

Total h^2

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

SNP h_j^2

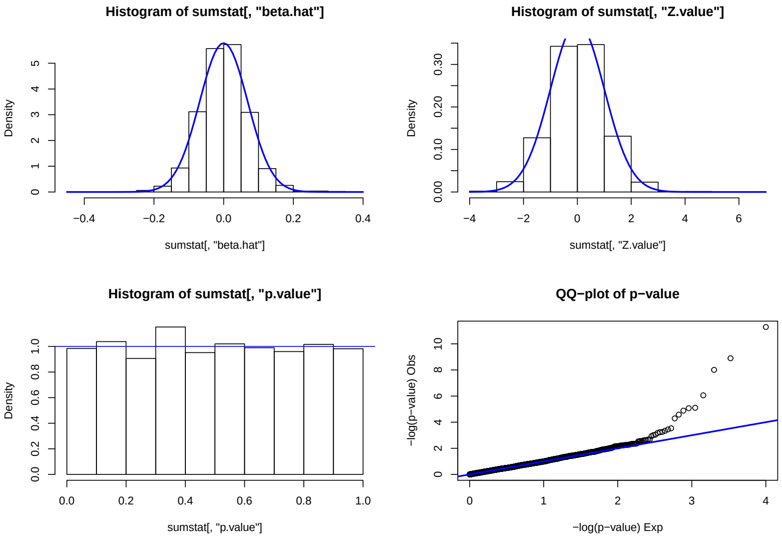
```
##  [1] 0.023 0.009 0.032 0.031 0.019 0.021 0.029 0.022 0.030 0.028
```

Recall the summary statistics

```
ex.sumstat[1:23,]
```

```
##                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
## [14,] 0.46923549  0.4740  0.0  0.03095466 0.05222091  0.5927637 5.534736e-01
## [15,] 0.25480427  0.2485  0.0  0.05966600 0.06226877  0.9582010 3.381935e-01
## [16,] 0.31564388  0.3205  0.0 -0.03353920 0.05695716 -0.5888496 5.560954e-01
## [17,] 0.41919624  0.4345  0.0 -0.08589125 0.05307934 -1.6181671 1.059426e-01
## [18,] 0.15085332  0.1410  0.0  0.03167344 0.07632138  0.4150009 6.782304e-01
## [19,] 0.23525007  0.2515  0.0 -0.05445552 0.05876396 -0.9266822 3.543156e-01
## [20,] 0.06737475  0.0740  0.0 -0.10570983 0.10173334 -1.0390873 2.990158e-01
## [21,] 0.36532020  0.3595  0.0  0.06726877 0.05527611  1.2169592 2.239074e-01
## [22,] 0.48057686  0.4785  0.0  0.00804454 0.05286361  0.1521754 8.790794e-01
## [23,] 0.14600840  0.1400  0.0 -0.06318882 0.07458090 -0.8472521 3.970578e-01
```

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



Recall effect size estimates in $PRS_i = \sum_{j=1}^{J} \hat{\beta}_j G_{ij}$: $\hat{\beta}_j$

Genome-wide significance level

```
J.index=which(ex.sumstat[,"p.value"]<=0.05/ex.nsnp); length(J.index)
```

```
## [1] 6
```

```
round(ex.sumstat[J.index,"beta.hat"],2)
```

```
## [1] 0.29 0.24 0.39 0.38 0.33 0.29
```

A less stringent significance level at 0.01

```
J.index=which(ex.sumstat[,"p.value"]<=0.01); length(J.index)
```

```
## [1] 66
```

```
round(ex.sumstat[J.index,"beta.hat"],2)
```

```
##  [1]  0.29  0.33  0.24  0.39  0.31  0.38  0.33  0.28  0.23  0.29 -0.20  0.15
## [13]  0.15 -0.17 -0.23 -0.25 -0.17 -0.17 -0.18 -0.31  0.17  0.18 -0.15 -0.16
## [25] -0.18  0.16  0.27 -0.19  0.19  0.19 -0.16 -0.16  0.33 -0.15 -0.15  0.17
## [37] -0.18  0.14 -0.14 -0.16  0.14 -0.24 -0.14  0.15  0.14 -0.14 -0.22 -0.17
## [49] -0.18  0.17 -0.20  0.15  0.14 -0.18  0.19 -0.21 -0.22 -0.43  0.33  0.15
## [61] -0.14 -0.15 -0.19 -0.22  0.26  0.18
```

Now also add 0.1

```
J.index=which(ex.sumstat[,"p.value"]<=0.1); length(J.index)
```

```
## [1] 492
```

Recall my.data WITHOUT any heterogeneity

```r
# no heterogeneity, i.e. same model with the same MAF but a new seed
my.nsnp.true=10; my.beta.true=0.3; my.maf=ex.sumstat[,"MAF"]
my.nsample=1000; my.nsnp=5000; 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)
my.data$my.sumstat[1:15,]
```

```
##              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
## [14,] 0.46923549  0.4700  0.0  0.07720190 0.05064487  1.5243776 1.277312e-01
## [15,] 0.25480427  0.2470  0.0  0.04074974 0.05878734  0.6931720 4.883629e-01
```

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)

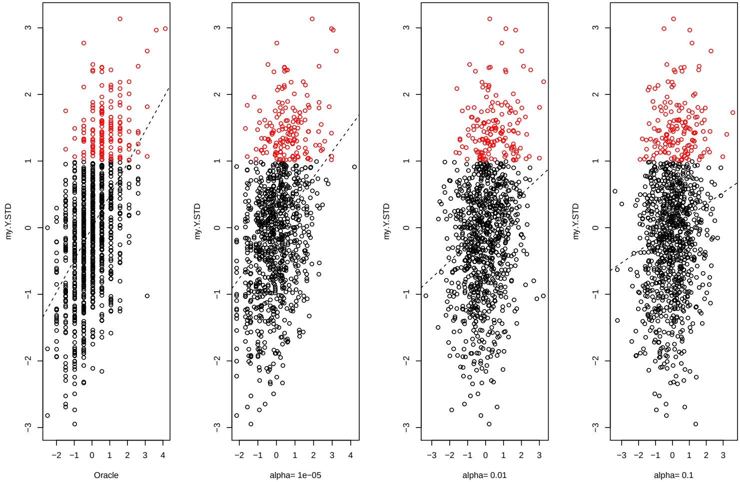
$$my.PRS_{oracle} = \sum_{j=1}^{10} 0.3 \times G_{ij}^{my.data}$$

(*my.PRS*.01.*null* omitted now; its expected AUC is 50%, the lower bound)

Recall alpha level, liability threshold and PRS calculation

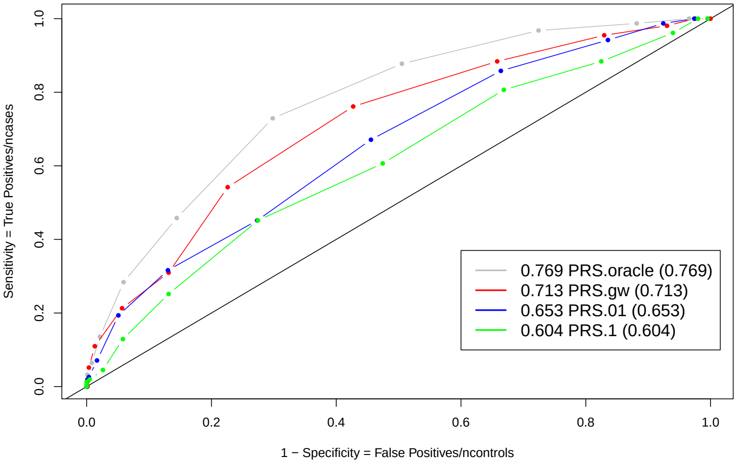
```
# the alphal level used on the external data
alpha.level=c((0.05/ex.nsnp),0.01,0.1)

# the liability threshold on the my.Y.STD scale
l.threshold=1

my.PRS.output=generate.my.PRS.output(ex.sumstat,my.data,alpha.level,l.threshold
```

Recall the association performance of the different PRSs (and adding $\alpha = 0.1$)




```
##       [,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"       " "
```

Recall the prediction performance of the PRSs




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

To make the lecture notes self-sufficient, first

- Exam the expected effects of n.external and beta on AUC:

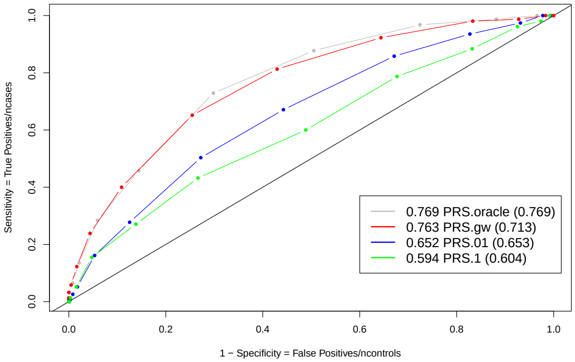
  AUC $\uparrow$ as $n_{ex} \uparrow$ (Quiz: effect of n_{my}?)

  AUC $\uparrow$ as $\beta \uparrow$ (assume $\beta_{ex} = \beta_{my}$)

- Also ask some less obvious questions.

Increase n_{ex} from 1000 to 2000

```
# external data
ex.nsample=1000*2 # HERE IS THE CHANGE
ex.nsnp=5000;ex.nsnp.true=10;ex.beta.true=0.3;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.nsample=1000; my.nsnp=5000; 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)
```




```
##       alpha   J TP  FP
## [1,] 1e-05  10 10   0
## [2,] 1e-02  58 10  48
## [3,] 1e-01 508 10 498
```

Some interesting Qs

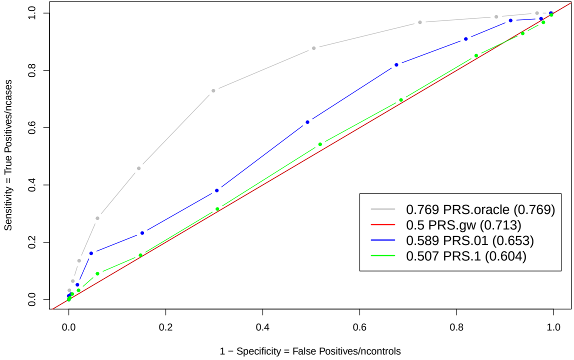
- Why PRS.oracle stayed the same?
- Why J of PRS.01 dropped from 66 to 58, when n increased from 1000 to 2000? Did we make a mistake?
  (Hint: $E(J) = 10 + 5000 * 0.01 = 60$ when $n \to \infty$)
- Why AUC of PRS.01 dropped from 0.653 to 0.652, when n increased from 1000 to 2000? Did we make a mistake?
- Why AUC of PRS.1 dropped from 0.604 to 0.594, when n increased from 1000 to 2000? Did we make a mistake?
- How large the n has to be before AUC of PRS.gw say $> 80\%$? Is this even possible?!

Live Quiz 2: If $n = 500$, the AUC of PRS.gw will drop from 0.713 to

A: $<$0.6
B: ~0.5
C: $<$0.5

When $n_{ex} = 500$, no signficant SNPs to construct PRS.gw!

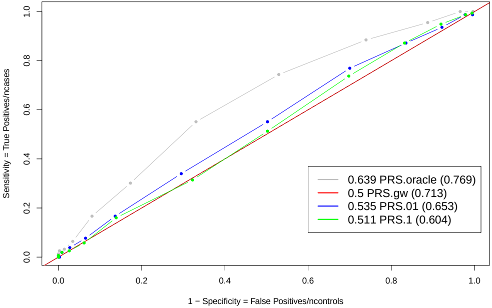
```
# external data
ex.nsample=1000/2 # HERE IS THE CHANGE
ex.nsnp=5000;ex.nsnp.true=10;ex.beta.true=0.3;ex.sigma=1;ex.seed=101
ex.sumstat=generate.ex.sumstat(ex.seed,ex.nsample,ex.nsnp,ex.nsnp.true,ex.beta.true,ex.sigma)
# my data
my.nsnp.true=10; my.beta.true=0.3; my.maf=ex.sumstat[,"MAF"]
my.nsample=1000; my.nsnp=5000; 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)
```




```
##       alpha   J TP  FP
## [1,] 1e-05   0  0   0
## [2,] 1e-02  70  7  63
## [3,] 1e-01 561  8 553
```

This can be ‘achieved’ by reducing β while keep $n_{ex} = 1000$ (But PRS.Oracle will drop as model changed)

```
# external data
ex.beta.true=0.3/2 # HERE IS THE CHANGE
ex.nsample=1000; ex.nsnp=5000;ex.nsnp.true=10;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.beta.true=0.3/2 # FOR THE MOMENT, NO HETEROGENEITY
my.nsnp.true=10; my.maf=ex.sumstat[,"MAF"];my.nsample=1000; my.nsnp=5000; 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)
```




```
##       alpha   J TP  FP
## [1,] 1e-05   0  0   0
## [2,] 1e-02  60  4  56
## [3,] 1e-01 511 10 501
```

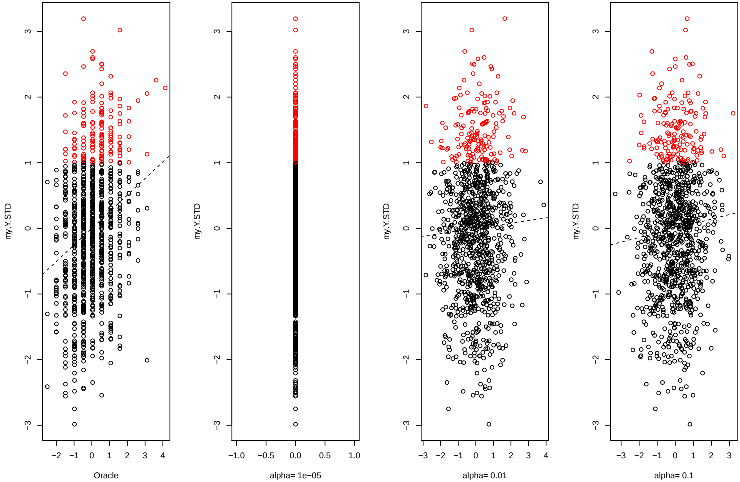
```r
ex.sumstat[1:13,]
```

```
##              MAF MAF.hat beta    beta.hat         se    Z.value      p.value
##  [1,] 0.21748927  0.2215 0.15  0.14600440 0.05973651  2.4441403 1.469200e-02
##  [2,] 0.06972117  0.0610 0.15  0.16677239 0.09955980  1.6750977 9.422833e-02
##  [3,] 0.36935781  0.3780 0.15  0.12319186 0.04864521  2.5324560 1.147900e-02
##  [4,] 0.34596068  0.3480 0.15  0.22596322 0.05123411  4.4104063 1.143756e-05
##  [5,] 0.16243508  0.1695 0.15  0.17073564 0.06438713  2.6517044 8.135624e-03
##  [6,] 0.18502467  0.1995 0.15  0.22913794 0.05956471  3.8468743 1.272255e-04
##  [7,] 0.31318998  0.3375 0.15  0.17349493 0.04971547  3.4897576 5.045849e-04
##  [8,] 0.20006021  0.2020 0.15  0.15356084 0.06087867  2.5224078 1.181001e-02
##  [9,] 0.32990538  0.3360 0.15  0.09991265 0.05170515  1.9323538 5.359859e-02
## [10,] 0.29562285  0.2905 0.15  0.13226661 0.05349339  2.4725786 1.357996e-02
## [11,] 0.44590808  0.4445 0.00  0.08383990 0.04923445  1.7028706 8.890366e-02
## [12,] 0.36809363  0.3745 0.00 -0.03212100 0.04811733 -0.6675557 5.045715e-01
## [13,] 0.37938767  0.3750 0.00 -0.07217775 0.04920997 -1.4667304 1.427644e-01
```

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

```
##              MAF MAF.hat beta    beta.hat         se    Z.value      p.value
##  [1,] 0.21748927  0.2270 0.15  0.01247489 0.05509259  0.2264350 8.209094e-01
##  [2,] 0.06972117  0.0755 0.15  0.17002640 0.08723730  1.9490102 5.157409e-02
##  [3,] 0.36935781  0.3555 0.15  0.18360183 0.04785363  3.8367381 1.325103e-04
##  [4,] 0.34596068  0.3545 0.15  0.11008786 0.04866656  2.2620842 2.390659e-02
##  [5,] 0.16243508  0.1530 0.15  0.17889081 0.06337038  2.8229406 4.853125e-03
##  [6,] 0.18502467  0.1800 0.15  0.10261456 0.06086522  1.6859310 9.212166e-02
##  [7,] 0.31318998  0.3045 0.15  0.20009426 0.05057621  3.9562921 8.150728e-05
##  [8,] 0.20006021  0.1780 0.15  0.18709681 0.06099449  3.0674380 2.217225e-03
##  [9,] 0.32990538  0.3300 0.15  0.13891836 0.04764986  2.9153995 3.631759e-03
## [10,] 0.29562285  0.2960 0.15  0.15937056 0.04980192  3.2000889 1.417193e-03
## [11,] 0.44590808  0.4465 0.00  0.03590208 0.04562390  0.7869139 4.315191e-01
## [12,] 0.36809363  0.3580 0.00 -0.04215175 0.04982051 -0.8460723 3.977152e-01
## [13,] 0.37938767  0.3870 0.00 -0.03979972 0.04760912 -0.8359685 4.033727e-01
```

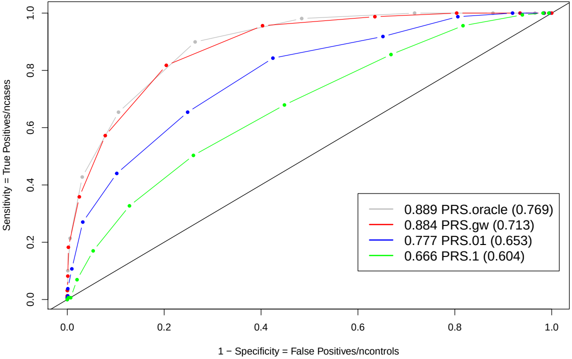
The association perspective: reduced as expected




```
##       [,1]                [,2]        [,3]  [,4]     [,5]
## [1,] "slope.hat"         "0.253"     "-9"  "0.039"  "0.07"
## [2,] "Z.value"           "8.261"     "-9"  "1.248"  "2.211"
## [3,] "p.value"           "4.55e-16"  "-9"  "0.2123" "0.02727"
## [4,] "n, case, control"  "1000"      "156" "844"    " "
```

The other way around: increase β while keep $n_{ex} = 1000$

```
# external data
ex.beta.true=0.3*2 # HERE IS THE CHANGE
ex.nsample=1000; ex.nsnp=5000;ex.nsnp.true=10;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.beta.true=0.3*2 # FOR THE MOMENT, NO HETEROGENEITY
my.nsnp.true=10; my.maf=ex.sumstat[,"MAF"]; my.nsample=1000; my.nsnp=5000; 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)
```




```
##      alpha   J TP  FP
## [1,] 1e-05  10 10   0
## [2,] 1e-02  62 10  52
## [3,] 1e-01 522 10 512
```

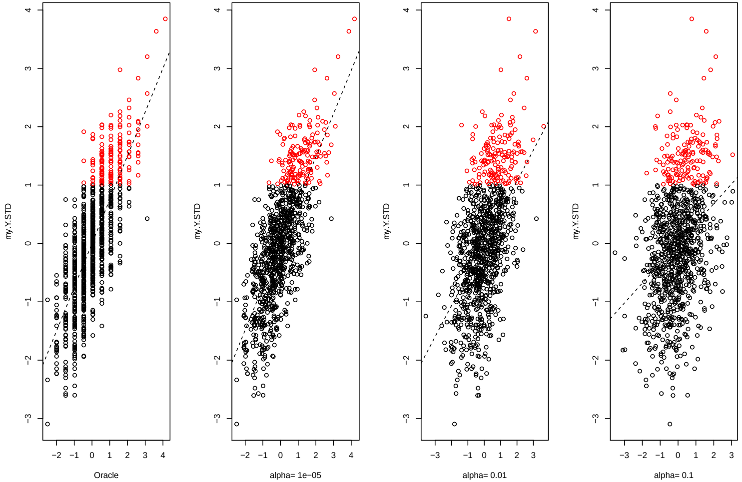
```
ex.sumstat[1:13,]
```

```
##              MAF MAF.hat beta     beta.hat         se     Z.value      p.value
##  [1,] 0.21748927  0.2215  0.6  0.585709843 0.08355777  7.00963925 4.394954e-12
##  [2,] 0.06972117  0.0610  0.6  0.660827952 0.14087871  4.69075813 3.100590e-06
##  [3,] 0.36935781  0.3780  0.6  0.470882014 0.06810161  6.91440369 8.375501e-12
##  [4,] 0.34596068  0.3480  0.6  0.714759811 0.07035014 10.16003382 3.818905e-23
##  [5,] 0.16243508  0.1695  0.6  0.585317380 0.09042326  6.47308458 1.503246e-10
##  [6,] 0.18502467  0.1995  0.6  0.669917015 0.08305361  8.06607984 2.069820e-15
##  [7,] 0.31318998  0.3375  0.6  0.647993428 0.06844419  9.46747192 1.999837e-20
##  [8,] 0.20006021  0.2020  0.6  0.537653234 0.08556724  6.28340042 4.939934e-10
##  [9,] 0.32990538  0.3360  0.6  0.490942068 0.07234984  6.78566924 1.978191e-11
## [10,] 0.29562285  0.2905  0.6  0.602662945 0.07423832  8.11795019 1.387824e-15
## [11,] 0.44590808  0.4445  0.0  0.119842457 0.07033241  1.70394363 8.870291e-02
## [12,] 0.36809363  0.3745  0.0 -0.003119627 0.06875198 -0.04537508 9.638174e-01
## [13,] 0.37938767  0.3750  0.0 -0.046647532 0.07035780 -0.66300444 5.074808e-01
```

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

```
##              MAF MAF.hat beta     beta.hat         se    Z.value      p.value
##  [1,] 0.21748927  0.2270  0.6  0.489991629 0.07915134  6.1905668 8.743198e-10
##  [2,] 0.06972117  0.0755  0.6  0.663358979 0.12622211  5.2554896 1.805160e-07
##  [3,] 0.36935781  0.3555  0.6  0.599845985 0.06796933  8.8252445 4.808576e-18
##  [4,] 0.34596068  0.3545  0.6  0.530413550 0.06942814  7.6397485 5.090469e-14
##  [5,] 0.16243508  0.1530  0.6  0.610090229 0.09111881  6.6955468 3.580370e-11
##  [6,] 0.18502467  0.1800  0.6  0.635288983 0.08693697  7.3074663 5.568375e-13
##  [7,] 0.31318998  0.3045  0.6  0.685512522 0.07139637  9.6015033 6.121637e-21
##  [8,] 0.20006021  0.1780  0.6  0.686081816 0.08704659  7.8817768 8.415737e-15
##  [9,] 0.32990538  0.3300  0.6  0.653724435 0.06692904  9.7674261 1.387754e-21
## [10,] 0.29562285  0.2960  0.6  0.696485818 0.06988728  9.9658454 2.290153e-22
## [11,] 0.44590808  0.4465  0.0  0.063646633 0.06678298  0.9530368 3.408022e-01
## [12,] 0.36809363  0.3580  0.0  0.020481781 0.07295969  0.2807274 7.789777e-01
## [13,] 0.37938767  0.3870  0.0 -0.007657671 0.06972295 -0.1098300 9.125663e-01
```

The association perspective: improved as expected



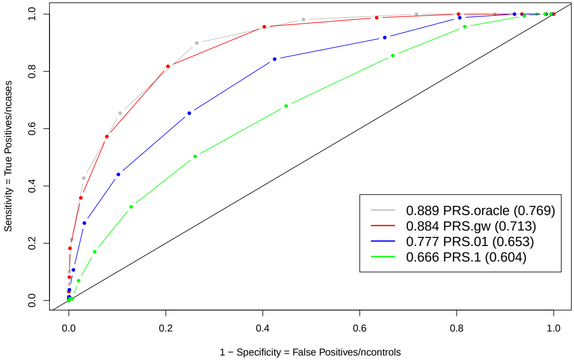

```
##        [,1]               [,2]         [,3]         [,4]        [,5]
## [1,] "slope.hat"         "0.751"      "0.741"      "0.533"     "0.34"
## [2,] "Z.value"           "35.878"     "34.891"     "19.892"    "11.422"
## [3,] "p.value"           "9.691e-182" "5.273e-175" "1.994e-74" "1.749e-28"
## [4,] "n, case, control" "1000"       "159"        "841"       " "
```

Keep $\beta = 0.3$ and $n_{ex} = 1000$, but DECREASE σ

Live Quiz 3: compared with beta=0.6, sigma =1,
AUC of beta=0.3, sigma =0.5 will be

A: smaller
B: larger
C: ~same
D: identical

```r
# external data
ex.sigma=0.5 # HERE IS THE CHANGE
ex.beta.true=0.3; ex.nsample=1000; ex.nsnp=5000;ex.nsnp.true=10;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.sigma=0.5 # no heterogeneity
my.beta.true=0.3;my.nsnp.true=10; my.maf=ex.sumstat[,"MAF"];my.nsample=1000; my.nsnp=5000;my.seed=102
my.data=generate.my.data(my.seed,my.nsample,my.nsnp,my.nsnp.true,my.beta.true,my.sigma,my.maf)
```




```
##        alpha   J  TP  FP
## [1,] 1e-05  10 10   0
## [2,] 1e-02  62 10  52
## [3,] 1e-01 522 10 512
```

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

```
##             MAF MAF.hat beta     beta.hat         se     Z.value      p.value
##  [1,] 0.21748927  0.2215  0.3  0.292854922 0.04177889  7.00963925 4.394954e-12
##  [2,] 0.06972117  0.0610  0.3  0.330413976 0.07043935  4.69075813 3.100590e-06
##  [3,] 0.36935781  0.3780  0.3  0.235441007 0.03405080  6.91440369 8.375501e-12
##  [4,] 0.34596068  0.3480  0.3  0.357379906 0.03517507 10.16003382 3.818905e-23
##  [5,] 0.16243508  0.1695  0.3  0.292658690 0.04521163  6.47308458 1.503246e-10
##  [6,] 0.18502467  0.1995  0.3  0.334958508 0.04152680  8.06607984 2.069820e-15
##  [7,] 0.31318998  0.3375  0.3  0.323996714 0.03422209  9.46747192 1.999837e-20
##  [8,] 0.20006021  0.2020  0.3  0.268826617 0.04278362  6.28340042 4.939934e-10
##  [9,] 0.32990538  0.3360  0.3  0.245471034 0.03617492  6.78566924 1.978191e-11
## [10,] 0.29562285  0.2905  0.3  0.301331472 0.03711916  8.11795019 1.387824e-15
## [11,] 0.44590808  0.4445  0.0  0.059921229 0.03516620  1.70394363 8.870291e-02
## [12,] 0.36809363  0.3745  0.0 -0.001559813 0.03437599 -0.04537508 9.638174e-01
## [13,] 0.37938767  0.3750  0.0 -0.023323766 0.03517890 -0.66300444 5.074808e-01
```

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

```
##             MAF MAF.hat beta     beta.hat         se    Z.value      p.value
##  [1,] 0.21748927  0.2270  0.3  0.244995815 0.03957567  6.1905668 8.743198e-10
##  [2,] 0.06972117  0.0755  0.3  0.331679490 0.06311105  5.2554896 1.805160e-07
##  [3,] 0.36935781  0.3555  0.3  0.299922993 0.03398467  8.8252445 4.808576e-18
##  [4,] 0.34596068  0.3545  0.3  0.265206775 0.03471407  7.6397485 5.090469e-14
##  [5,] 0.16243508  0.1530  0.3  0.305045114 0.04555940  6.6955468 3.580370e-11
##  [6,] 0.18502467  0.1800  0.3  0.317644491 0.04346849  7.3074663 5.568375e-13
##  [7,] 0.31318998  0.3045  0.3  0.342756261 0.03569819  9.6015033 6.121637e-21
##  [8,] 0.20006021  0.1780  0.3  0.343040908 0.04352330  7.8817768 8.415737e-15
##  [9,] 0.32990538  0.3300  0.3  0.326862217 0.03346452  9.7674261 1.387754e-21
## [10,] 0.29562285  0.2960  0.3  0.348242909 0.03494364  9.9658454 2.290153e-22
## [11,] 0.44590808  0.4465  0.0  0.031823317 0.03339149  0.9530368 3.408022e-01
## [12,] 0.36809363  0.3580  0.0  0.010240891 0.03647984  0.2807274 7.789777e-01
## [13,] 0.37938767  0.3870  0.0 -0.003828836 0.03486148 -0.1098300 9.125663e-01
```

Why identical results: puzzling?!

Hint/Solution in the heritability formula:

$$\begin{aligned}
\text{(narrow) } h^2 &= \frac{V_G}{V_G + V_e} \\
&= \frac{\sum_j^J \beta_j^2 2p_j(1-p_j)}{\sum_j^J \beta_j^2 2p_j(1-p_j) + \sigma^2} \\
&= \frac{\sum_j^J (\frac{\beta_j}{2})^2 2p_j(1-p_j)}{\sum_j^J (\frac{\beta_j}{2})^2 2p_j(1-p_j) + (\frac{\sigma}{2})^2}
\end{aligned}$$

Now consider a 'more polygenic' model

ex.beta.true from 0.3 to 0.1 but ex.nsnp.true from 10 to 90

- h^2 and SNP h^2? Answers in the h^2 expression below:

$$\text{(narrow) } h^2 = \frac{V_G}{V_G + V_e} = \frac{\sum_j \beta_j^2 Var(G_j)}{Var(Y)} = \frac{\sum_j \beta_j^2 2p_j(1-p_j)}{\sum_j \beta_j^2 2p_j(1-p_j) + \sigma^2}.$$

- AUC in general?
- AUC between PRS.gw and PRS.01?

Live Quiz 4: Compared with 10 SNPs with beta=0.3,
the trait h2 of 90 SNPs with beta=0.1 will

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

The ‘more polygenic’ model: 90 signals each with $\beta = 0.1$

```
# external data
ex.nsnp.true=90; ex.beta.true=0.1 # HERE IS THE CHANGE
ex.nsample=1000; ex.nsnp=5000;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=90; my.beta.true=0.1;my.maf=ex.sumstat[,"MAF"] #NO HETEROGENEITY
my.nsample=1000; my.nsnp=5000; 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 h^2 (=0.243 for the previous model)

```
## [1] 0.254
```

(would be identical if the MAFs of the 10 causal SNPs in the previous model were duplicated eight times for the additional 80 causal SNPs.)

Heritability of GWAS SNPs (In the previous model, the first 10 SNPs have h^2: 0.023 0.009 0.032 0.031 0.019 0.021 0.029 0.022 0.030 0.028)

```
##  [1] 0.003 0.001 0.003 0.003 0.002 0.002 0.003 0.002 0.003 0.003 0.004 0.003
## [13] 0.004 0.004 0.003 0.003 0.004 0.002 0.003 0.001 0.003 0.004 0.002 0.003
## [25] 0.004 0.004 0.001 0.003 0.003 0.003 0.003 0.002 0.002 0.002 0.003 0.004
## [37] 0.002 0.004 0.001 0.004 0.003 0.003 0.004 0.002 0.003 0.002 0.001 0.004
## [49] 0.004 0.001 0.001 0.003 0.003 0.004 0.003 0.004 0.003 0.003 0.003 0.003
## [61] 0.004 0.004 0.003 0.004 0.004 0.003 0.001 0.002 0.004 0.004 0.002 0.004
## [73] 0.003 0.004 0.004 0.002 0.001 0.002 0.003 0.004 0.003 0.004 0.004 0.004
## [85] 0.001 0.004 0.002 0.003 0.001 0.001
```

an exact factor of $(\frac{0.3}{0.1})^2 = 9$ **for the first 10 SNPs, up to some rounding errors**

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

```
##              MAF MAF.hat beta   beta.hat         se    Z.value      p.value
##  [1,] 0.21748927  0.2215  0.1 0.12081718 0.06627959 1.82284148 6.862648e-02
##  [2,] 0.06972117  0.0610  0.1 0.22222689 0.11024967 2.01566948 4.410218e-02
##  [3,] 0.36935781  0.3780  0.1 0.04687522 0.05405469 0.86718140 3.860511e-01
##  [4,] 0.34596068  0.3480  0.1 0.24837192 0.05677965 4.37431251 1.346113e-05
##  [5,] 0.16243508  0.1695  0.1 0.05953832 0.07157131 0.83187408 4.056790e-01
##  [6,] 0.18502467  0.1995  0.1 0.16549441 0.06628250 2.49680387 1.269210e-02
##  [7,] 0.31318998  0.3375  0.1 0.08997211 0.05534992 1.62551478 1.043686e-01
##  [8,] 0.20006021  0.2020  0.1 0.05790201 0.06764749 0.85593734 3.922379e-01
##  [9,] 0.32990538  0.3360  0.1 0.02768553 0.05739314 0.48238396 6.296390e-01
## [10,] 0.29562285  0.2905  0.1 0.08611827 0.05939298 1.44997375 1.473800e-01
## [11,] 0.44590808  0.4445  0.1 0.22361807 0.05417375 4.12779355 3.968460e-05
## [12,] 0.36809363  0.3745  0.1 0.04717400 0.05330820 0.88492951 3.764078e-01
## [13,] 0.37938767  0.3750  0.1 0.00457198 0.05458647 0.08375666 9.332667e-01
```

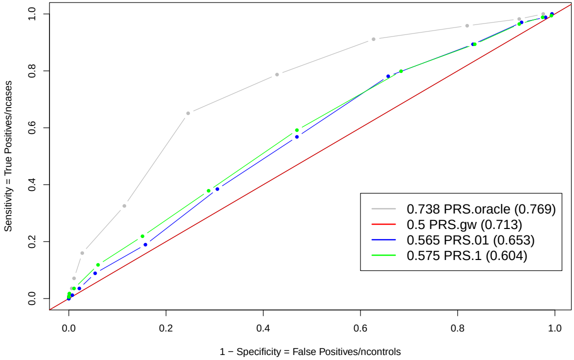
```r
my.data$my.sumstat[1:13,]
```

```
##              MAF MAF.hat beta     beta.hat         se    Z.value      p.value
##  [1,] 0.21748927  0.2270  0.1 -0.018359121 0.06138469 -0.2990831 0.7649389309
##  [2,] 0.06972117  0.0755  0.1  0.126935376 0.09730439  1.3045185 0.1923576011
##  [3,] 0.36935781  0.3555  0.1  0.159905582 0.05347275  2.9904125 0.0028543881
##  [4,] 0.34596068  0.3545  0.1  0.076838091 0.05431019  1.4148007 0.1574388573
##  [5,] 0.16243508  0.1530  0.1  0.100955620 0.07081851  1.4255541 0.1543097864
##  [6,] 0.18502467  0.1800  0.1 -0.006995559 0.06791406 -0.1030061 0.9179788583
##  [7,] 0.31318998  0.3045  0.1  0.191669621 0.05646877  3.3942590 0.0007152209
##  [8,] 0.20006021  0.1780  0.1  0.154289405 0.06810668  2.2654080 0.0237012821
##  [9,] 0.32990538  0.3300  0.1  0.062084469 0.05328232  1.1651983 0.2442171393
## [10,] 0.29562285  0.2960  0.1  0.120550255 0.05564412  2.1664511 0.0305128398
## [11,] 0.44590808  0.4465  0.1  0.180504428 0.05052930  3.5722726 0.0003707974
## [12,] 0.36809363  0.3580  0.1  0.057220824 0.05550191  1.0309704 0.3028044996
## [13,] 0.37938767  0.3870  0.1  0.038532121 0.05305210  0.7263071 0.4678208168
```

As expected

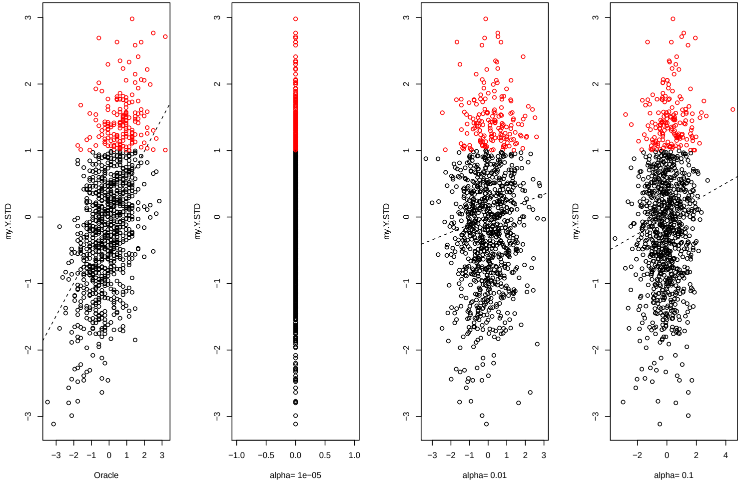
All PRS's performance dropped considerably
(with the exception of Oracle, which depends on the SUM of $\beta_j^2(2p_j(1-p_j))$)

Between PRS.01 and PRS.1, Which one is better?




```
##       alpha   J TP  FP
## [1,] 1e-05   0  0   0
## [2,] 1e-02  78 20  58
## [3,] 1e-01 526 40 486
```

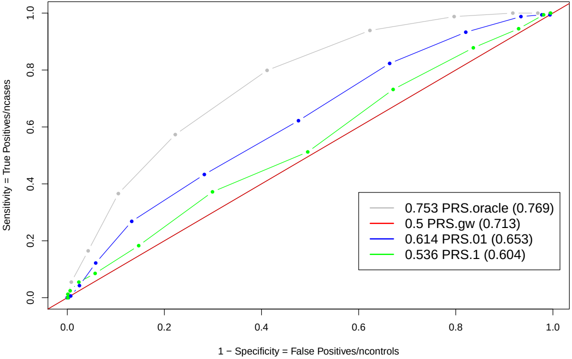
The association perspective: reduced as expected




```
##        [,1]             [,2]        [,3]  [,4]        [,5]
## [1,] "slope.hat"        "0.495"     "-9"  "0.114"     "0.127"
## [2,] "Z.value"          "18.019"    "-9"  "3.626"     "4.046"
## [3,] "p.value"          "4.621e-63" "-9"  "0.0003019" "5.612e-05"
## [4,] "n, case, control" "1000"      "169" "831"       " "
```

A demonstration of sampling variation: my.seed=104

```
# external data
ex.nsnp.true=90; ex.beta.true=0.1
ex.nsample=1000; ex.nsnp=5000;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=90; my.beta.true=0.1;my.maf=ex.sumstat[,"MAF"]
my.nsample=1000; my.nsnp=5000; my.sigma=1;my.seed=104 # HERE IS THE CHANGE
my.data=generate.my.data(my.seed,my.nsample,my.nsnp,my.nsnp.true,my.beta.true,my.sigma,my.maf)
```




```
##         alpha   J TP  FP
## [1,] 1e-05    0  0   0
## [2,] 1e-02   78 20  58
## [3,] 1e-01  526 40 486
```

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

```
##              MAF MAF.hat beta   beta.hat         se    Z.value      p.value
##  [1,] 0.21748927  0.2215  0.1 0.12081718 0.06627959 1.82284148 6.862648e-02
##  [2,] 0.06972117  0.0610  0.1 0.22222689 0.11024967 2.01566948 4.410218e-02
##  [3,] 0.36935781  0.3780  0.1 0.04687522 0.05405469 0.86718140 3.860511e-01
##  [4,] 0.34596068  0.3480  0.1 0.24837192 0.05677965 4.37431251 1.346113e-05
##  [5,] 0.16243508  0.1695  0.1 0.05953832 0.07157131 0.83187408 4.056790e-01
##  [6,] 0.18502467  0.1995  0.1 0.16549441 0.06628250 2.49680387 1.269210e-02
##  [7,] 0.31318998  0.3375  0.1 0.08997211 0.05534992 1.62551478 1.043686e-01
##  [8,] 0.20006021  0.2020  0.1 0.05790201 0.06764749 0.85593734 3.922379e-01
##  [9,] 0.32990538  0.3360  0.1 0.02768553 0.05739314 0.48238396 6.296390e-01
## [10,] 0.29562285  0.2905  0.1 0.08611827 0.05939298 1.44997375 1.473800e-01
## [11,] 0.44590808  0.4445  0.1 0.22361807 0.05417375 4.12779355 3.968460e-05
## [12,] 0.36809363  0.3745  0.1 0.04717400 0.05330820 0.88492951 3.764078e-01
## [13,] 0.37938767  0.3750  0.1 0.00457198 0.05458647 0.08375666 9.332667e-01
```

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

```
##              MAF MAF.hat beta    beta.hat         se   Z.value      p.value
##  [1,] 0.21748927  0.2160  0.1  0.09258794 0.06298784  1.469934 0.1418949014
##  [2,] 0.06972117  0.0670  0.1 -0.03844030 0.10497421 -0.366188 0.7143023853
##  [3,] 0.36935781  0.3630  0.1  0.20964101 0.05473719  3.829956 0.0001361615
##  [4,] 0.34596068  0.3430  0.1  0.10125006 0.05396060  1.876370 0.0608960987
##  [5,] 0.16243508  0.1675  0.1  0.19376458 0.07032348  2.755333 0.0059699770
##  [6,] 0.18502467  0.1910  0.1  0.13460914 0.06790534  1.982306 0.0477189503
##  [7,] 0.31318998  0.2915  0.1  0.08702704 0.05723256  1.520586 0.1286804598
##  [8,] 0.20006021  0.2065  0.1  0.11932271 0.06532614  1.826569 0.0680631629
##  [9,] 0.32990538  0.3150  0.1  0.11431448 0.05664997  2.017909 0.0438679399
## [10,] 0.29562285  0.3000  0.1  0.16448772 0.05633534  2.919796 0.0035813483
## [11,] 0.44590808  0.4420  0.1  0.08635832 0.05583359  1.546709 0.1222504558
## [12,] 0.36809363  0.3770  0.1  0.15122082 0.05209983  2.902520 0.0037831667
## [13,] 0.37938767  0.3835  0.1  0.14872792 0.05310641  2.800564 0.0051998402
```

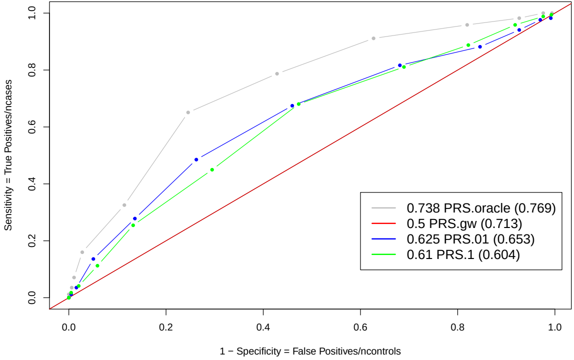
The message is clear

Don’t claim AUC.1=0.575 is better than AUC.01=0.565 from a single run!

Don’t claim AUC.01=0.614 is better than ACU.1=0.536 from a single run either!

The same 90 and $\beta = 0.1$ model, but $n_{ex} = 2000$

```
# external data
ex.nsample=2000 # HERE IS THE CHANGE
ex.nsnp.true=90; ex.beta.true=0.1; ex.nsnp=5000;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=90; my.beta.true=0.1;my.maf=ex.sumstat[,"MAF"]
my.nsample=1000; my.nsnp=5000; 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)
```




```
##       alpha   J TP  FP
## [1,] 1e-05   0  0   0
## [2,] 1e-02  88 40  48
## [3,] 1e-01 534 69 465
```

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

```
##              MAF MAF.hat beta   beta.hat         se  Z.value      p.value
##  [1,] 0.21748927 0.22250  0.1 0.06522651 0.04437269 1.469970 1.417274e-01
##  [2,] 0.06972117 0.07800  0.1 0.08940668 0.06868688 1.301656 1.931841e-01
##  [3,] 0.36935781 0.36475  0.1 0.05804817 0.03739792 1.552176 1.207784e-01
##  [4,] 0.34596068 0.34050  0.1 0.08792224 0.03897037 2.256130 2.417018e-02
##  [5,] 0.16243508 0.16675  0.1 0.07100976 0.05010728 1.417154 1.565937e-01
##  [6,] 0.18502467 0.18600  0.1 0.09813412 0.04679786 2.096979 3.612086e-02
##  [7,] 0.31318998 0.31050  0.1 0.16358684 0.03920681 4.172409 3.143107e-05
##  [8,] 0.20006021 0.20175  0.1 0.06320495 0.04628974 1.365420 1.722748e-01
##  [9,] 0.32990538 0.34025  0.1 0.14511701 0.03830620 3.788343 1.561412e-04
## [10,] 0.29562285 0.29000  0.1 0.11538534 0.04008788 2.878310 4.040619e-03
## [11,] 0.44590808 0.44275  0.1 0.06226218 0.03688941 1.687806 9.160452e-02
## [12,] 0.36809363 0.37625  0.1 0.11627357 0.03765381 3.087963 2.043047e-03
## [13,] 0.37938767 0.37350  0.1 0.10949100 0.03821326 2.865262 4.210207e-03
```

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

```
##              MAF MAF.hat beta     beta.hat         se    Z.value      p.value
##  [1,] 0.21748927  0.2270  0.1 -0.018359121 0.06138469 -0.2990831 0.7649389309
##  [2,] 0.06972117  0.0755  0.1  0.126935376 0.09730439  1.3045185 0.1923576011
##  [3,] 0.36935781  0.3555  0.1  0.159905582 0.05347275  2.9904125 0.0028543881
##  [4,] 0.34596068  0.3545  0.1  0.076838091 0.05431019  1.4148007 0.1574388573
##  [5,] 0.16243508  0.1530  0.1  0.100955620 0.07081851  1.4255541 0.1543097864
##  [6,] 0.18502467  0.1800  0.1 -0.006995559 0.06791406 -0.1030061 0.9179788583
##  [7,] 0.31318998  0.3045  0.1  0.191669621 0.05646877  3.3942590 0.0007152209
##  [8,] 0.20006021  0.1780  0.1  0.154289405 0.06810668  2.2654080 0.0237012821
##  [9,] 0.32990538  0.3300  0.1  0.062084469 0.05328232  1.1651983 0.2442171393
## [10,] 0.29562285  0.2960  0.1  0.120550255 0.05564412  2.1664511 0.0305128398
## [11,] 0.44590808  0.4465  0.1  0.180504428 0.05052930  3.5722726 0.0003707974
## [12,] 0.36809363  0.3580  0.1  0.057220824 0.05550191  1.0309704 0.3028044996
## [13,] 0.37938767  0.3870  0.1  0.038532121 0.05305210  0.7263071 0.4678208168
```

Increased performance as expected, but with some interesting observations

Increase n_{ex} from 1000 to 2000 did not balance out the drop of β from 0.3 to 0.1: still no SNPs with p values less than 10^{-5}.

Quiz: how large the n should be to achieve similar performance with the earlier model of 10 SNPs, $\beta = 0.3$ and $n_{ex} = 2000$?
(More efficient R codes needed to demonstrate this empirically.)
(Analytical hint: $h^2 \propto \beta^2$ and $s.e. \propto \sqrt{n}$)

Another example of 'more is NOT always better':
AUC of $\text{PRS}_{0.01}$ (40+48=88 SNPs; 0.625) and
AUC of $\text{PRS}_{0.1}$ (69+465=534 SNPs; 0.610) are practically the same.

Recap the goal of this lecture, a **deeper** understanding of

- 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?

What's next: Effects of various (population and locus) **heterogeneities**, and the importance of reference allele (and genome build) matching.

- my.reference.allele $\neq$ reference.allele
- my.maf $\neq$ ex.maf
- my.beta.true $\neq$ ex.beta.true
- my.nsnp.true $\neq$ ex.nsnp.true