Polygenic Risk Score (PRS) Introduction 201

basic PRS calculation and performance evaluation

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

- the complexity of constructing a good PRS even under the simplest setting *without* LD or any heterogeneities..
- 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 gw-significant SNPs vs. 66 0.01-significant SNPS.
- the various over-fitting or selection biases, and winner’s curse in $\hat{\beta}$ for both false positives and true positives.

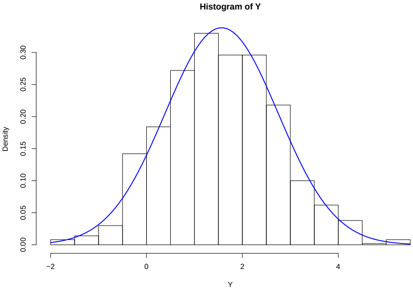
Recall the illustrative ‘polygenic’ model simulation study

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

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

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

```
nsnp.true=10 # number of truly associated SNPs
beta.true=0.3 # "large" effect (also MAF, the error term, and the sample size)
```



Recall (NOT realistic!) $PRS_{i,oracle} = \sum_{j=1}^{J=10} 0.3 \cdot G_{ij}$

The MAF of the 10 truly associated SNPs

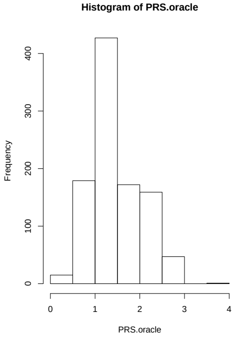
```
##  [1] 0.217 0.070 0.369 0.346 0.162 0.185 0.313 0.200 0.330 0.296
```

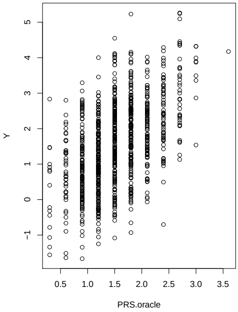
The SNP heritability vary, despite all $\beta_j = 0.3$

```
##  [1] 0.025 0.010 0.034 0.033 0.020 0.022 0.032 0.024 0.033 0.031
```

The true heritability, h^2

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

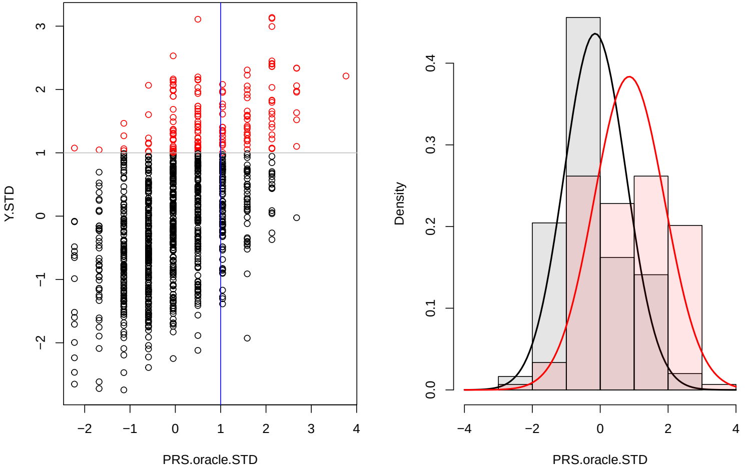




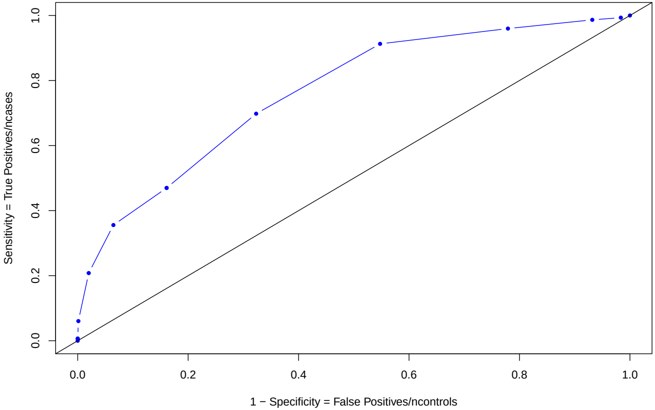
Recall the (highly significant) association between PRS.orcale and the trait

```
##
## Call:
## lm(formula = Y.STD ~ PRS.oracle.STD)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.71113 -0.60843 -0.01341  0.60283  2.86310
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -6.748e-17  2.750e-02    0.00        1
## PRS.oracle.STD  4.942e-01  2.752e-02   17.96   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8698 on 998 degrees of freedom
## Multiple R-squared:  0.2442, Adjusted R-squared:  0.2435
## F-statistic: 322.5 on 1 and 998 DF,  p-value: < 2.2e-16
```

Recall the liability model, and the case-control stratified PRS distributions



Recall the ROC curve and AUC using PRS_{oracle}



```
## [1] "AUC of ROC.oracle=" "0.763"
```

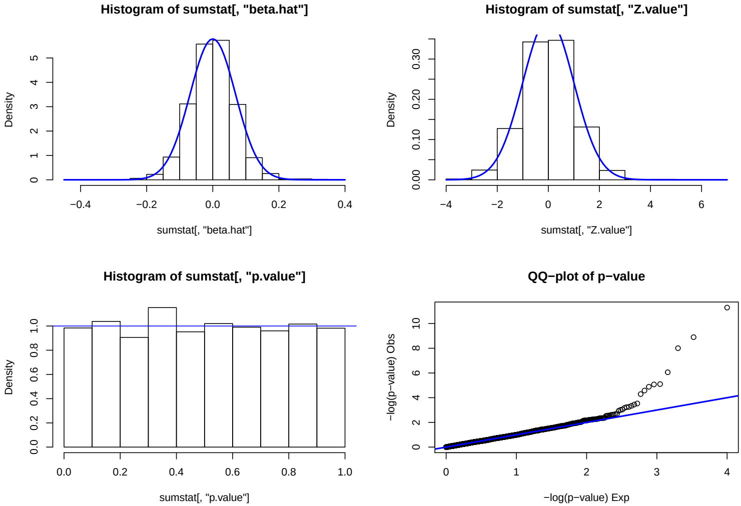
Recall the **BUT**,

$$\mathbf{PRS_{i,oracle}} = \sum_{j=1}^{J=10} \beta_j (= 0.3) G_{ij} \text{ \textbf{is NOT} } \mathbf{PRS_{i,parctice}}!$$

- J is unknown, to be determined
- β_j is unknown, to be estimated
- G_{ij} **cannot be directly from the same data used to infer** J **and** β_j.

  Otherwise: over-fitting/double-dipping/data-dredging/p-hacking/selection-bias!
- Not to mention LD and other considerations in real data settings.

What we have: $\hat{\beta}$, Z and p-values



Histogram of sumstat[, "beta.hat"]

Histogram of sumstat[, "Z.value"]

Histogram of sumstat[, "p.value"]

QQ−plot of p−value

Recall the sumstat (the true beta and MAF are here thanks to simulation)

```
##              MAF MAF.hat beta    beta.hat         se     Z.value      p.value
##  [1,] 0.21748927  0.2215  0.3  0.29257288 0.06536792  4.47578705 8.489445e-06
##  [2,] 0.06972117  0.0610  0.3  0.33145758 0.10935747  3.03095507 2.500692e-03
##  [3,] 0.36935781  0.3780  0.3  0.23908858 0.05323916  4.49084039 7.922031e-06
##  [4,] 0.34596068  0.3480  0.3  0.38889542 0.05565755  6.98728935 5.116550e-12
##  [5,] 0.16243508  0.1695  0.3  0.30892955 0.07052329  4.38053229 1.308960e-05
##  [6,] 0.18502467  0.1995  0.3  0.37606430 0.06503910  5.78212649 9.859505e-09
##  [7,] 0.31318998  0.3375  0.3  0.33166110 0.05410586  6.12985587 1.264930e-09
##  [8,] 0.20006021  0.2020  0.3  0.28159164 0.06670313  4.22156565 2.647447e-05
##  [9,] 0.32990538  0.3360  0.3  0.23025579 0.05661344  4.06715744 5.134017e-05
## [10,] 0.29562285  0.2905  0.3  0.28906539 0.05841261  4.94868102 8.766086e-07
## [11,] 0.44590808  0.4445  0.0  0.09584075 0.05424572  1.76678916 7.756916e-02
## [12,] 0.36809363  0.3745  0.0 -0.02245388 0.05302784 -0.42343559 6.720687e-01
## [13,] 0.37938767  0.3750  0.0 -0.06366768 0.05424574 -1.17368994 2.407993e-01
## [14,] 0.46923549  0.4740  0.0  0.03095466 0.05222091  0.59276375 5.534736e-01
## [15,] 0.25480427  0.2485  0.0  0.05966600 0.06226877  0.95820104 3.381935e-01
## [16,] 0.31564388  0.3205  0.0 -0.03353920 0.05695716 -0.58884957 5.560954e-01
## [17,] 0.41919624  0.4345  0.0 -0.08589125 0.05307934 -1.61816710 1.059426e-01
## [18,] 0.15085332  0.1410  0.0  0.03167344 0.07632138  0.41500089 6.782304e-01
## [19,] 0.23525007  0.2515  0.0 -0.05445552 0.05876396 -0.92668222 3.543156e-01
## [20,] 0.06737475  0.0740  0.0 -0.10570983 0.10173334 -1.03908733 2.990158e-01
## [21,] 0.36532020  0.3595  0.0  0.06726877 0.05527611  1.21695922 2.239074e-01
## [22,] 0.48057686  0.4785  0.0  0.00804454 0.05286361  0.15217538 8.790794e-01
## [23,] 0.14600840  0.1400  0.0 -0.06318882 0.07458090 -0.84725206 3.970578e-01
## [24,] 0.34747768  0.3405  0.0 -0.00162502 0.05702567 -0.02849630 9.772720e-01
## [25,] 0.46549350  0.4590  0.0  0.07744989 0.05301480  1.46091065 1.443547e-01
## [26,] 0.40807389  0.4185  0.0  0.02135427 0.05247701  0.40692616 6.841495e-01
## [27,] 0.08204565  0.0820  0.0 -0.00372138 0.09588876 -0.03880935 9.690502e-01
## [28,] 0.22523350  0.2240  0.0 -0.09720419 0.06372866 -1.52528210 1.275056e-01
## [29,] 0.23290305  0.2455  0.0 -0.03611566 0.05999003 -0.60202767 5.472925e-01
## [30,] 0.34670979  0.3425  0.0  0.04700733 0.05368057  0.87568618 3.814114e-01
```

Determine J and Estimate β_j using GW significance level

```
J.index=which(sumstat[,"p.value"]<=0.05/nsnp) #10^-5 here for 5000 SNPs
J.index # the index for the significant SNPs
```

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

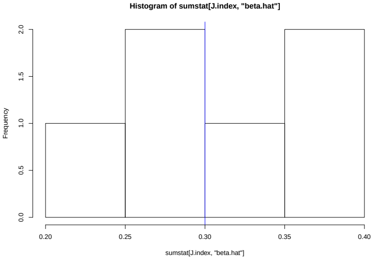
```
c(length(J.index),sum(J.index<=nsnp.true)) # positives, true positives
```

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

```
round(sumstat[J.index,"beta.hat"],2) # beta estimates for the significant SNPs
```

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

```
hist(sumstat[J.index,"beta.hat"]); abline(v=beta.true,col="blue")
```



A less stringent significance level, say $\alpha = 0.01$?

Trade-off: between false positives (56) and power (10 out 10)

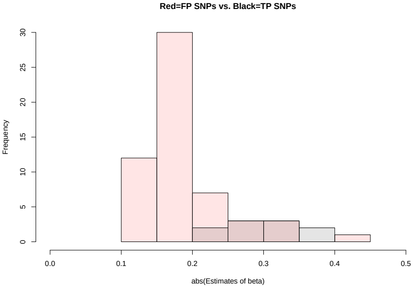
```
J.index=which(sumstat[,"p.value"]<=0.01)
J.index # the index for significant SNPs
```

```
##  [1]    1    2    3    4    5    6    7    8    9   10  324  349  358  385  509
## [16]  610  681  709  720  803  923  941 1248 1249 1275 1284 1346 1388 1451 1575
## [31] 1597 1651 1673 1702 1764 1782 1784 1835 1945 2343 2390 2518 2531 2561 2606
## [46] 2708 2726 2827 2909 3125 3207 3358 3372 3453 3584 3622 3646 3656 3871 3879
## [61] 3889 4182 4304 4472 4588 4935
```

```
c(length(J.index),sum(J.index<=nsnp.true)) # positives, true positives
```

```
## [1] 66 10
```

Trouble ahead: $|\hat{\beta}_j|$ of the FP SNPs are competitive!



Top associated SNPs for $PRS_i = \sum_{j=1}^{J} \hat{\beta}_j G_{ij}$: J

Genome-wide significance level ($\alpha = 10^{-5}$ here for 5000 SNPs)

- $J = 6$
- find only 6 out 10 truly associated SNPs
- but 0 false positives

A less stringent significance level ($\alpha = 0.01$)

- $J = 66$
- find all 10 truly associated SNP
- but 56 false positives

Live Quiz 1: Which α threshold will leads to a better PRS (higher AUC)?

A: using 6 SNPS with GW significance
B: using 66 SNPs with p< 0.01
C: ~same

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(sumstat[,"p.value"]<=0.05/nsnp)
round(sumstat[J.index,"beta.hat"],2)
```

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

A less stringent significance level

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

That was too easy! **More considerations** later:

- Winner's curse (a result of low power) and the MAF connection
- heterogeneity and transportability
- LD

Quiz: patterns for the signicant SNPs?

MAF of the 10 truly associated SNPs

```
##  [1] 0.22 0.07 0.37 0.35 0.16 0.19 0.31 0.20 0.33 0.30
```

MAF of the 6 significant SNPs at the GW level

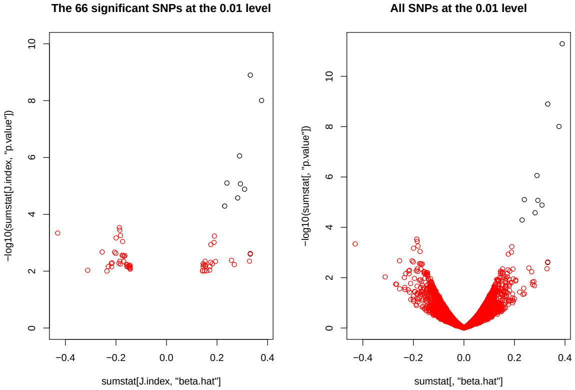
```
## [1] 0.22 0.37 0.35 0.19 0.31 0.30
```

Sample estimates of the 6 significant SNPs at the GW level

```
## [1] 0.22 0.38 0.35 0.20 0.34 0.29
```

Quiz cont'd, -log(GWAS p-value) vs. $\hat{\beta}_j$

(Red = FP SNPs vs. Black = TP SNPs)



Sun et al. (2011). *Human Genetics*. BR-squared: a practical solution to the winner's curse in genome-wide scans.

Can we construct $PRS_i = \sum_{j=1}^{J} \hat{\beta}_j G_{ij}$ now?

What we have using GW $\alpha = 10^{-5}$ (for 5000 SNPs):

The number of SNPs, J

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

Which specific SNPs

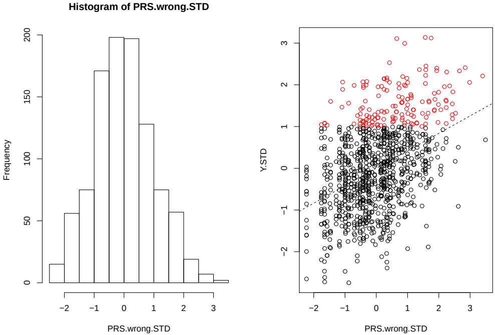
```
## [1]  1  3  4  6  7 10
```

$\hat{\beta}_j$ **of these SNPS**

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

WRONG if using G_{ij} from the same data!

$$PRS_{i,wrong} = \sum_{j=1}^{6} \hat{\beta}_j G_{ij}$$



This PRS.wrong appears to be ~normally distributed and highly predictive of the outcome, BUT **due to over-fitting!**

If you are not fully convinced of the over-fitting issue:

Using $\alpha = 0.01$ BUT exclude the all the true positives. That is, **using only the following 56 false positive SNPs to construct PRS:**

```
## [1] 56
```

```
##  [1]  324  349  358  385  509  610  681  709  720  803  923  941 1248 1249 1275
## [16] 1284 1346 1388 1451 1575 1597 1651 1673 1702 1764 1782 1784 1835 1945 2343
## [31] 2390 2518 2531 2561 2606 2708 2726 2827 2909 3125 3207 3358 3372 3453 3584
## [46] 3622 3646 3656 3871 3879 3889 4182 4304 4472 4588 4935
```

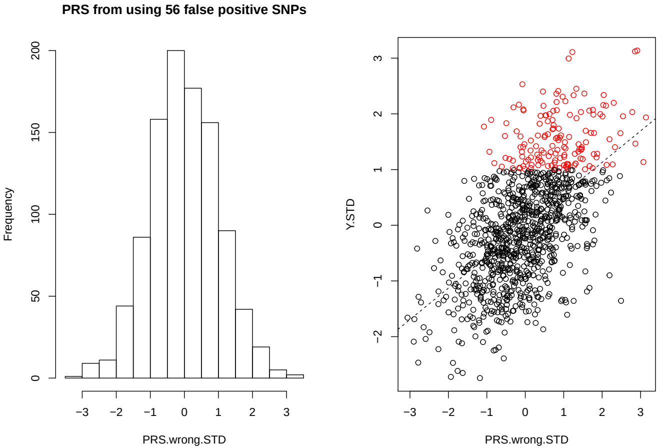
Their effect size sample estimates:

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

Their MAF sample estimates:

```
##  [1] 0.29 0.37 0.34 0.47 0.10 0.12 0.28 0.30 0.19 0.05 0.20 0.41 0.37 0.29 0.29
## [16] 0.23 0.08 0.50 0.20 0.30 0.35 0.34 0.06 0.33 0.40 0.23 0.20 0.43 0.46 0.32
## [31] 0.36 0.10 0.37 0.28 0.41 0.37 0.12 0.32 0.45 0.23 0.19 0.46 0.32 0.47 0.37
## [46] 0.19 0.14 0.05 0.06 0.29 0.46 0.37 0.19 0.13 0.09 0.21
```

PRS, using only null SNPs, is predictive: clearly WRONG!



Obtaining a significant result $\neq$ a correct result!

This PRS.wrong, constructed from the 56 null SNPs, is actually more significantly associated with the phenotype than PRS.oracle: clearly wrong!

```
##
## Call:
## lm(formula = Y.STD ~ PRS.wrong.STD)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.76357 -0.52506 -0.01203  0.54337  2.57385
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.493e-16  2.611e-02    0.00        1
## PRS.wrong.STD  5.646e-01  2.613e-02   21.61   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8258 on 998 degrees of freedom
## Multiple R-squared:  0.3188, Adjusted R-squared:  0.3181
## F-statistic:   467 on 1 and 998 DF,  p-value: < 2.2e-16
```

Cannot be overemphasized

A predictive (and normally distributed) PRS, on its own, is not evidence for correct PRS calculation!

Remember the superscripts in your PRS calculation:

$$\hat{\beta}_j^{external\,(base,\,discovery)} \times G_{ij}^{my.data\,(target,\,validation)}$$

Surely we will not make this rookie mistake! BUT,

over-fitting can appear in other (subtler) forms, e.g. overlapping samples between the external and my data, or pleitropy studies of multiple phenotypes from a single sample

How to construct $PRS_i = \sum_{j=1}^{J} \hat{\beta}_j G_{ij}$ then? e.g. **The simplest scenario**

Obtain the J and $\hat{\beta}_j$ from **an external data set**.

The external data set resembles our own data set perfectly, i.e. **no heterogeneity** in population, sampling design etc.

Calculate the PRS_i for each individual i in our own sample for prediction:

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

Simulate an independent set of data, my.data

```
# Assume the previous data was the external data
# the external model was
# nsnp=5000; nsnp.true=10; beta.true=0.3; beta.0=0; sigma=1

# Use the SAME MODEL but a DIFFERNT SEE to generate new independent data

set.seed(102)

my.nsample=1000 # my. is for my own data for prediction or validation
my.nsnp=nsnp # no heterogeneity: the same number of SNPs
my.maf=maf # no heterogeneity: the same MAF as before
my.nsnp.true=nsnp.true # no heterogeneity: the same number of truly associated S
my.beta.true=beta.true # no heterogeneity: the same effect size as before
my.beta=c(rep(my.beta.true,my.nsnp.true),rep(0,(my.nsnp-my.nsnp.true)))
```

Using the same model as above (no heterogeneity):
10 out 5000 SNPs are truly associated with ‘moderate-large’ effect

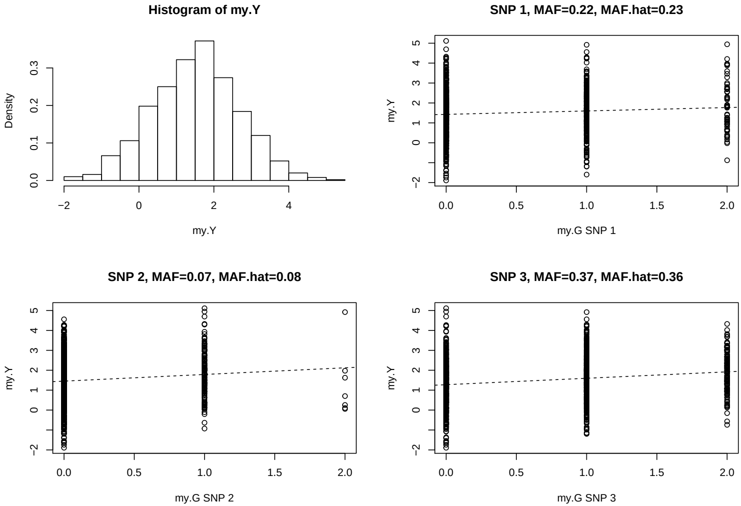
$$Y_i^{my.data} = \sum_{j=1}^{10} 0.3 \times G_{ij}^{my.data} + e^{my.data}, \text{ where } e^{my.data} \sim N(0,1)$$

MAFs stay the same as the external data (no heterogeneity), and recall the MAFs of the 10 truly associated SNPs:

```
##  [1] 0.22 0.07 0.37 0.35 0.16 0.19 0.31 0.20 0.33 0.30
```

The sample size does not have to be the same, but for now we use my.nsample=1000.

EDA (exploratory data analysis) of my.data



my.sumstat (the true beta and MAF are here thanks to simulation)

```
##              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
```

compared with the **ex.sumstat** from the external data

```
##              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
```

Finally, $PRS_i = \sum_{j=1}^{J} \hat{\beta}_j G_{ij}$

Using GW threshold on the external data

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

Using the $\alpha = 0.01$ threshold on the external data

$$my.PRS_{.01} = \sum_{j=\{1:10,324,\ldots,4935\}}^{66\ Positives\ (10\ TP)} \hat{\beta}_j^{external} \times G_{ij}^{my.data}$$

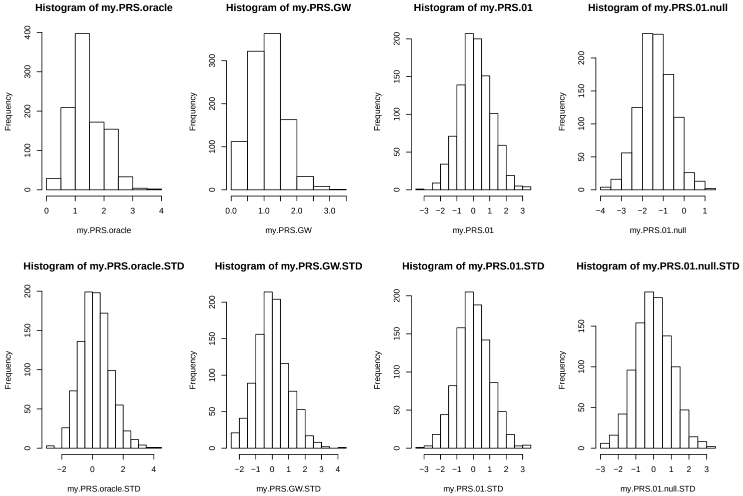
Using the $\alpha = 0.01$ threshold on the external data AND use only the false positives (made possible by the simulation and should NOT be predictive when calculated correctly!)

$$my.PRS_{.01.null} = \sum_{j=\{324,\ldots,4935\}}^{56\ False\ Positives} \hat{\beta}_j^{external} \times G_{ij}^{my.data}$$

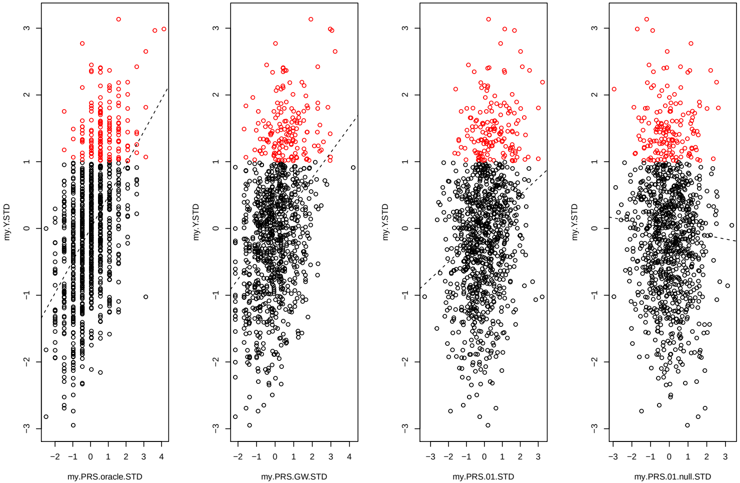
The oracle one (made possible by the simulation)

$$my.PRS_{Oracle} = \sum_{j=1}^{all\ 10\ causal\ ones} 0.3 \times G_{ij}^{my.data}$$

Raw and standardized (STD) of the PRSs constructed



Performance of the PRSs, from the association perspective



```
## [1] "slope.hat" "0.484"     "0.38"      "0.251"     "-0.053"

## [1] "Z.value" "17.467"  "12.983"  "8.185"   "-1.675"

## [1] "p.value"   "8.131e-60" "1e-35"     "8.22e-16"  "0.094"
```

$PRS.01.null$ is NOT associated with the trait as expected!

```
##                          Estimate Std. Error       t value   Pr(>|t|)
## (Intercept)         1.345526e-17 0.03159422  4.258771e-16 1.00000000
## my.PRS.01.null.STD -5.295421e-02 0.03161003 -1.675234e+00 0.09420154
```

PRS_{oracle} is the best, but PRS_{oracle} is not realistic!

```
##                       Estimate Std. Error      t value     Pr(>|t|)
## (Intercept)       1.642237e-15 0.02768826 5.931169e-14 1.000000e+00
## my.PRS.oracle.STD 4.838675e-01 0.02770211 1.746681e+01 8.130893e-60
```

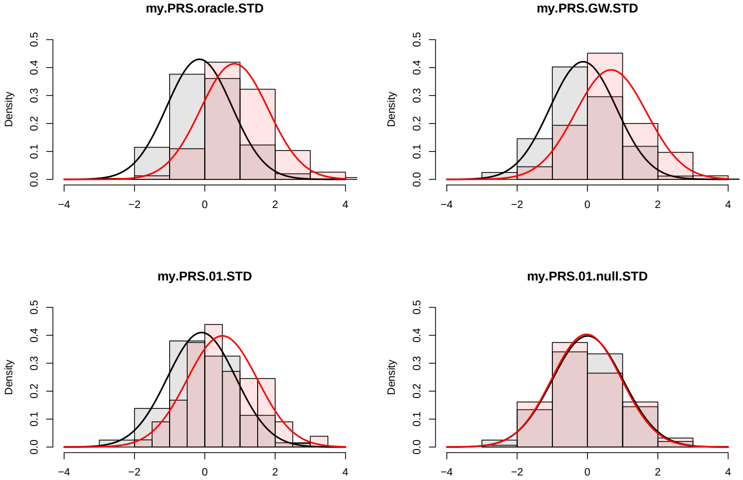
PRS_{GW} is significantly associated with the phenotype, but less so than PRS_{oracle} as it should be

```
##                    Estimate Std. Error       t value     Pr(>|t|)
## (Intercept)   -4.464004e-17 0.02926376 -1.525438e-15 1.000000e+00
## my.PRS.GW.STD  3.801176e-01 0.02927841  1.298287e+01 1.000642e-35
```

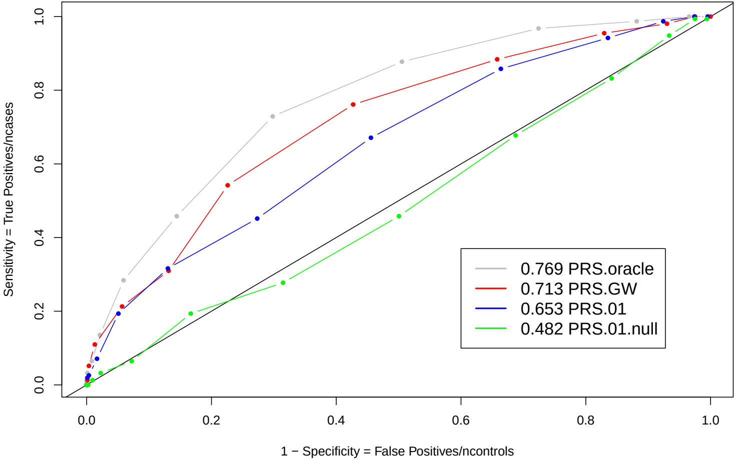
More is not necessarily better: $PRS_{.01}(J = 66)$ is worse than $PRS_{GW}(J = 6)$ in this case

```
##                    Estimate Std. Error       t value    Pr(>|t|)
## (Intercept)   -1.324645e-17 0.03062723 -4.325055e-16 1.00000e+00
## my.PRS.01.STD  2.508221e-01 0.03064256  8.185419e+00 8.22472e-16
```

Case-control stratified distributions of the different PRSs

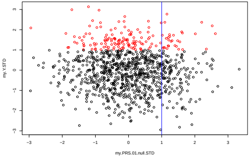


Performance of the PRSs, from the prediction perspective



Undertanding of the main diagonal line and AUC=50% of a non-predictive PRS

- Recall the scatter plot for the non-predictive PRS.0.01.null.STD



- Let $K < 1$ be the population prevalence of the disease, so out of a total of n samples, expect $n_{case} = n \cdot K$.

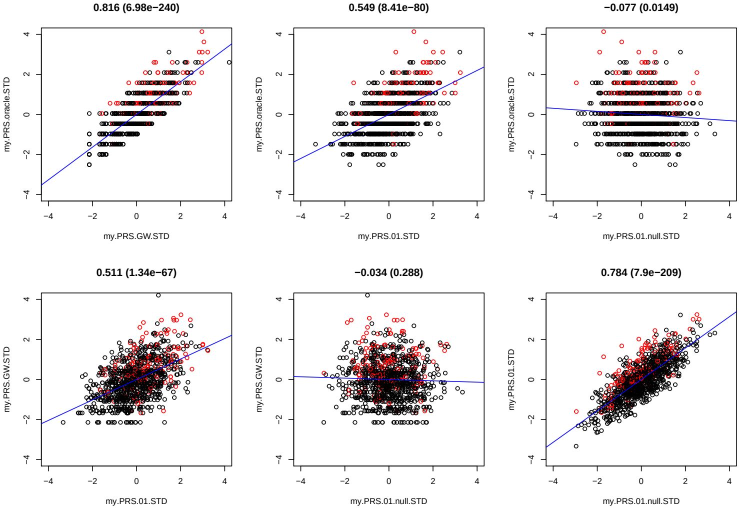
- For each threshold t used to call P_t samples positives (cases),

- Because the PRS used is not predictive, the expected true positives, $TP_t = P_t \cdot K$, and the expected sensitivity $= \frac{TP_t}{n_{case}} = \frac{P_t \cdot K}{n \cdot K} = \frac{P_t}{n}$.

- Similarly, the expected false positives, $FP_t = P_t \cdot (1 - K)$, and the expected $1 -$ specificity $= \frac{FP_t}{n_{control}} = \frac{P_t \cdot (1-K)}{n \cdot (1-K)} = \frac{P_t}{n}$.

- Thus, sensitivity (x) = 1-specificity (y) across the whole $\frac{P_t = 0}{n} = 0$ to $\frac{P_t = n}{n} = 1$ range. That is, ROC of a non-predictive PRS is (expected) to be the main diagonal line $(x = y)$, and AUC=50%.

Quiz: How can two PRSs with very different predictive performance be highly correlated?



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

- the complexity of constructing a good PRS even under the simplest setting *without* LD or any heterogeneties.
- 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 gw-significant SNPs vs. 66 0.01-significant SNPS.
- the various over-fitting or selection biases, $\hat{\beta}$ for a false positive or a true positive.

What's next

- 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?