

Polygenic Risk Score (PRS) Introduction 401

heterogeneity and PRS transportability

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 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`

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
```

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

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

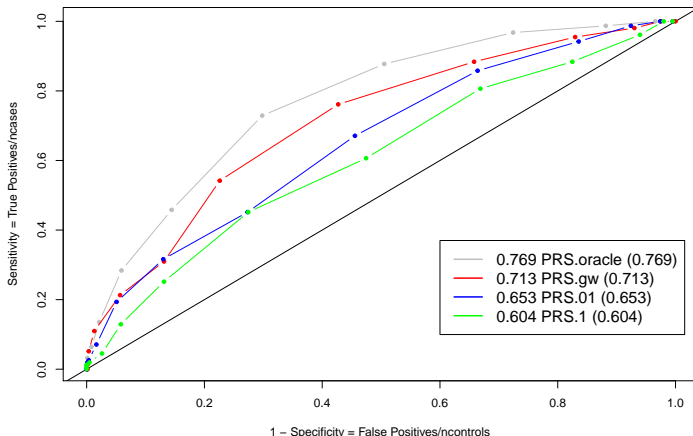
The oracle one (benchmarking the upper bound)

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

The baseline model ROC and AUC

```
# generate the ROC plots
```

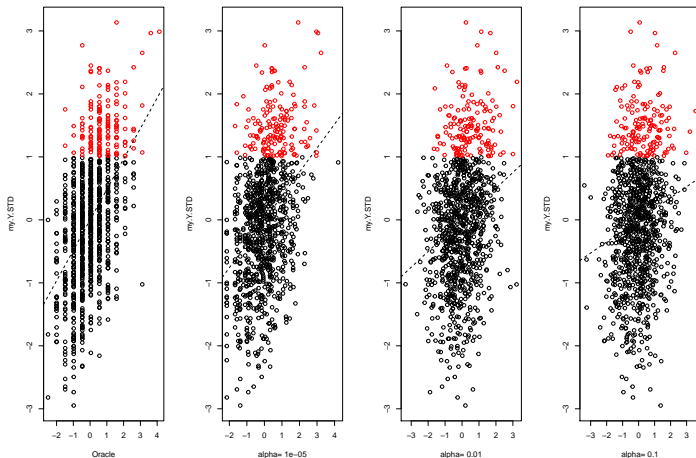
```
my.PRS.output=generate.my.PRS.output(ex.sumstat,my.data,alpha.level,l.threshold)  
generate.ROC.plot(my.PRS.output)
```



```
##      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

Getting to heterogeneity & transportability; first

Live Quiz 5: If $\text{my.MAF} = 1 - \text{ex.MAF}$ but everything else stay the same, the AUC of the PRS's will

A: drop a bit

B: drop a lot

C: stay \sim the same

D: stay exactly the same

my.MAF = 1-ex.MAF

```
# 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= 1-ex.sumstat["MAF"] # HERE is the change
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
```

Total and SNP h^2 of my model (Quiz: Why the same?)

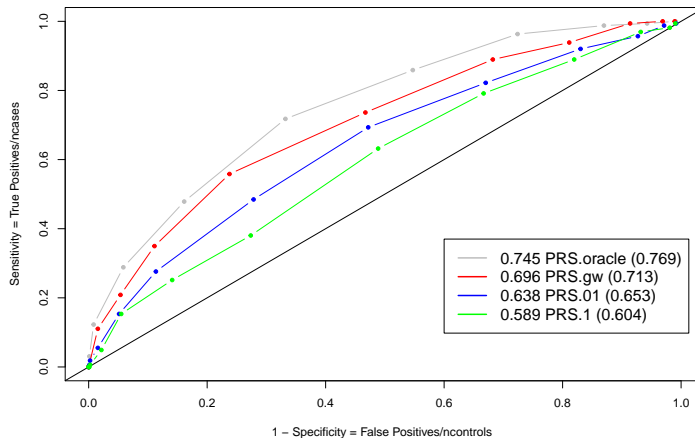
```
## [1] 0.243
```

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

my.MAF = 1-ex.MAF

```
# 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
```

my.MAF = 1-ex.MAF

```
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,]	0.7825107	0.7655	0.3	0.375996822	0.06271352	5.99546654	2.832450e-09
##	[2,]	0.9302788	0.9260	0.3	0.194755460	0.09911016	1.96504030	4.968663e-02
##	[3,]	0.6306422	0.6490	0.3	0.301510853	0.05425665	5.55712292	3.518434e-08
##	[4,]	0.6540393	0.6425	0.3	0.381795602	0.05499153	6.94280711	6.915674e-12
##	[5,]	0.8375649	0.8285	0.3	0.380162640	0.07006420	5.42591842	7.237061e-08
##	[6,]	0.8149753	0.8240	0.3	0.292733171	0.06814409	4.29579660	1.910812e-05
##	[7,]	0.6868100	0.6765	0.3	0.353171790	0.05514230	6.40473424	2.316153e-10
##	[8,]	0.7999398	0.7840	0.3	0.270011620	0.06248039	4.32154180	1.704508e-05
##	[9,]	0.6700946	0.6760	0.3	0.265927085	0.05320516	4.99814500	6.833292e-07
##	[10,]	0.7043772	0.6915	0.3	0.388615109	0.05571763	6.97472403	5.572054e-12
##	[11,]	0.5540919	0.5645	0.0	-0.009005340	0.05084476	-0.17711442	8.594545e-01
##	[12,]	0.6319064	0.6380	0.0	-0.002954014	0.05399585	-0.05470817	9.563819e-01
##	[13,]	0.6206123	0.6325	0.0	0.019359832	0.05316016	0.36417937	7.158012e-01

my.MAF = 1 - ex.MAF means MAF heterogeneity, BUT

$p_j(1 - p_j) = (1 - p_j)p_j$, and in general MAF heterogeneity alone does not have a huge impact on AUC, even if my.MAF independent of ex.MAF:

```
# 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.seed=105;set.seed(my.seed) # try other seeds and results in general stable.
my.maf= runif(my.nsnp,min=0.05,max=0.5) # HERE is the change
my.nsnp=5000; my.nsample=1000; my.sigma=1
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
```

Total and SNP h^2 of my model

```
## [1] 0.253
```

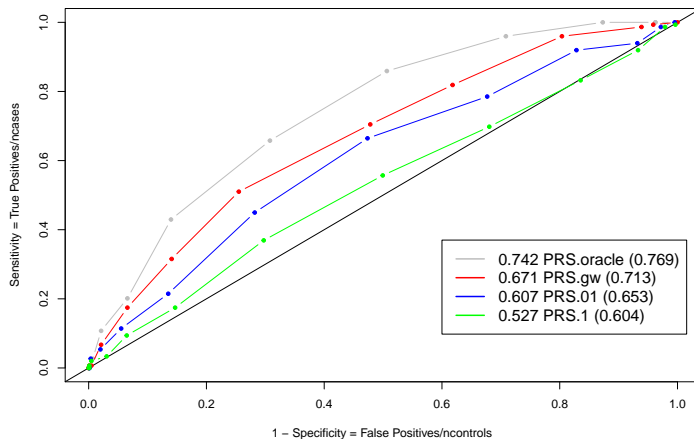
```
## [1] 0.011 0.034 0.021 0.024 0.028 0.032 0.033 0.033 0.028 0.008
```

my.MAF independent of ex.MAF

```
# 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
```

my.MAF independent of ex.MAF

```
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,]	0.09406173	0.0835	0.3	0.28606906	0.09344614	3.0613256	2.262561e-03
##	[2,]	0.49803509	0.4930	0.3	0.34668319	0.05006147	6.9251495	7.790725e-12
##	[3,]	0.19424863	0.1790	0.3	0.16372495	0.06851677	2.3895604	1.705334e-02
##	[4,]	0.22897688	0.2180	0.3	0.25331838	0.06088021	4.1609318	3.442646e-05
##	[5,]	0.29895136	0.3035	0.3	0.28989475	0.05716889	5.0708478	4.719622e-07
##	[6,]	0.40808856	0.3990	0.3	0.30475255	0.05207216	5.8525042	6.562553e-09
##	[7,]	0.45601252	0.4585	0.3	0.28647383	0.05104426	5.6122632	2.586697e-08
##	[8,]	0.42256943	0.4215	0.3	0.30389451	0.05223709	5.8176003	8.034968e-09
##	[9,]	0.29610631	0.2940	0.3	0.25611695	0.05720684	4.4770337	8.441016e-06
##	[10,]	0.06761723	0.0685	0.3	0.26169060	0.09992649	2.6188312	8.956916e-03
##	[11,]	0.34568337	0.3535	0.0	-0.07792184	0.05538050	-1.4070267	1.597308e-01
##	[12,]	0.45735690	0.4540	0.0	0.01296087	0.05229033	0.2478635	8.042910e-01
##	[13,]	0.31748569	0.3205	0.0	0.05404836	0.05355114	1.0092849	3.130826e-01

Make my.reference.allele \neq ex.reference.allele

Assume all reference alleles have been switched.

An easy way to achieve this is

changing the sign of the beta of ex.sumstat.

Live Quiz 6: Switch the reference allele between my.data and ex.data, the AUC of the PRS's will drop to

A: still $> 50\%$

B: $\sim 50\%$

C: $< 50\%$

Switch the reference allele

```
# 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)
ex.sumstat[, "beta"]=(-ex.sumstat[, "beta"]) # HERE is the change
ex.sumstat[, "beta.hat"]=(-ex.sumstat[, "beta.hat"]) # HERE is the change

# 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 (recall: $\beta_j^2 p_j(1 - p_j)$)

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

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

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
```


Switch the reference allele

```
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

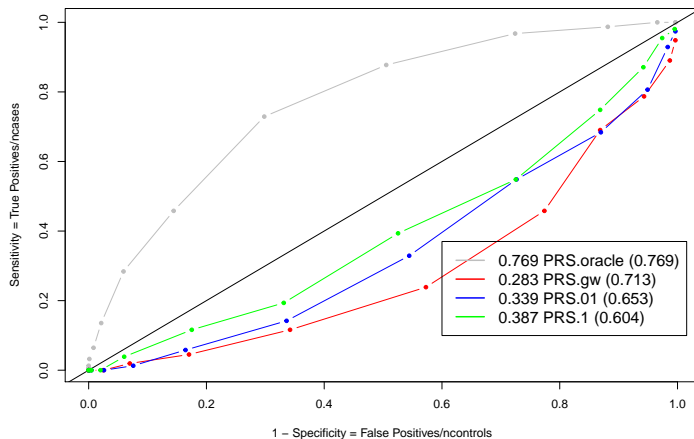
```
my.data$my.sumstat[1:13,]
```

##		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

Switch the reference allele

```
# generate the ROC plots
```

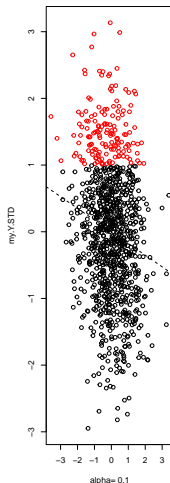
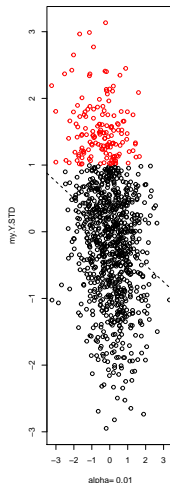
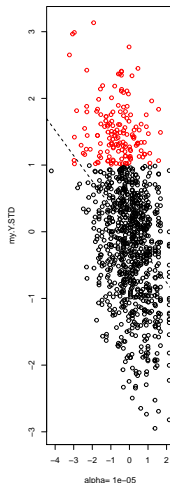
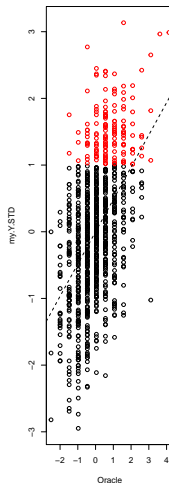
```
my.PRS.output=generate.my.PRS.output(ex.sumstat,my.data,alpha.level,l.threshold)  
generate.ROC.plot(my.PRS.output)
```



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

Switch the reference allele

```
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"	" "	

Lessons learned so far and more questions

Reference allele (and genome build) matching is really important!

(Population) heterogeneity can lead to MAF heterogeneity, which on its own does not have a big impact on PRS performance.

(Population) heterogeneity can also lead to **my.beta.true** \neq **ex.beta.true**.

- ▶ As ex.beta.true decreases, PRS performance decreases: obvious
- ▶ **As ex.beta.true increases, PRS performance increases?**
- ▶ Quiz: as ex.beta.true increases, can AUC increase to ~100%?

(Population) heterogeneity: decrease ex.beta.true to 0.3/2

```
# external data
ex.nsnp.true=10; ex.beta.true=0.3/2 # HERE is the only change
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 ($\frac{1}{4} V_G / (\frac{1}{4} V_G + \sigma^2) = \frac{1}{4} 0.321 / (\frac{1}{4} 0.321 + 1) = 0.074$)

```
## [1] 0.074
```

```
## [1] 0.007 0.003 0.010 0.009 0.006 0.006 0.009 0.007 0.009 0.009
```

Total and SNP h^2 of my model

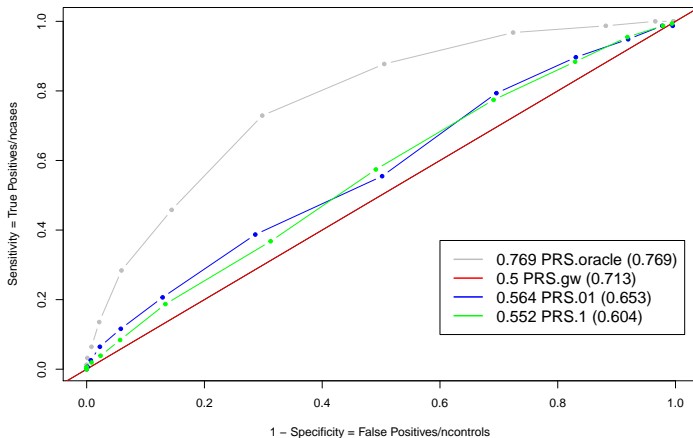
```
## [1] 0.243
```

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

(Population) heterogeneity: decrease ex.beta.true to 0.3/2

```
# 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    0  0   0  
## [2,] 1e-02   60  4  56  
## [3,] 1e-01  511 10 501
```

(Population) heterogeneity: decrease ex.beta.true to 0.3/2

```
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

```
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

(Population) heterogeneity: increase ex.beta.true to 0.3*2

```
# external data
ex.nsnp.true=10; ex.beta.true=0.3*2 # HERE is the only change
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 ($\frac{4}{1} V_G / (\frac{4}{1} V_G + \sigma^2) = \frac{4}{1} 0.321 / (\frac{4}{1} 0.321 + 1) = 0.562$)

```
## [1] 0.562
```

```
## [1] 0.054 0.020 0.073 0.071 0.043 0.048 0.068 0.050 0.070 0.066
```

Total and SNP h^2 of my model

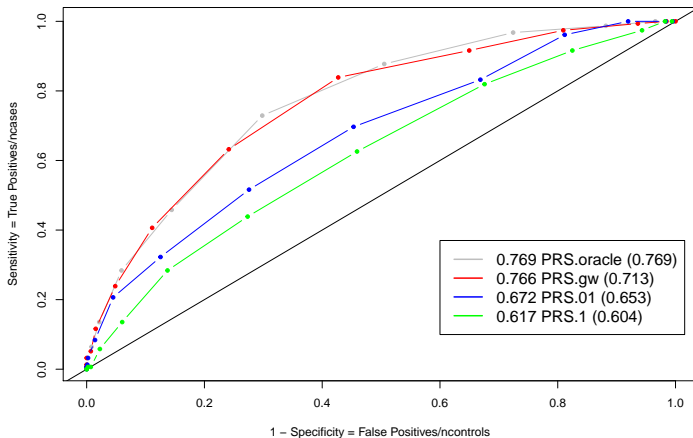
```
## [1] 0.243
```

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


(Population) heterogeneity: increase ex.beta.true to 0.3*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  10 10   0  
## [2,] 1e-02  62 10  52  
## [3,] 1e-01 522 10 512
```

(Population) heterogeneity: increase ex.beta.true to 0.3*2

```
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	0.3	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

(Locus) heterogeneity: $\text{my.nsnp.true} \neq \text{ex.nsnp.true}$; ex.nsnp.true reduced to 5

```
# external data
ex.nsnp.true=5 # HERE is the only change
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] 0.13
```

```
## [1] 0.027 0.010 0.036 0.035 0.021
```

Total and SNP h^2 of my model

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

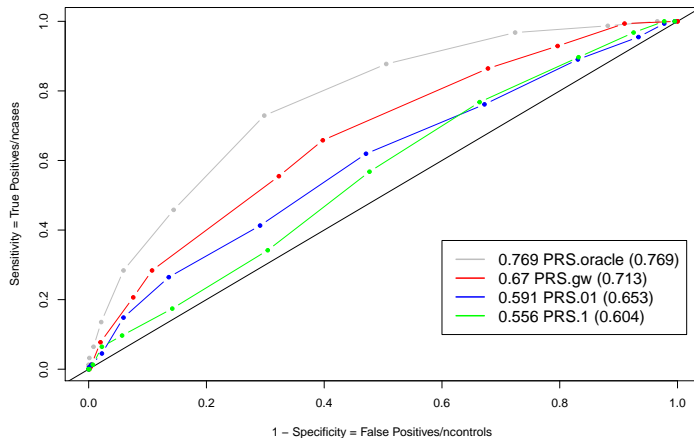
```
## [1] 0.023 0.009 0.032 0.031 0.019 0.021 0.029 0.022 0.030 0.028
```

(Locus) heterogeneity: ex.nsnp.true reduced to 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   4  4   0
## [2,] 1e-02  55  5  50
## [3,] 1e-01 516  5 511
```

(Locus) heterogeneity: ex.nsnp.true reduced to 5

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

##		MAF	MAF.hat	beta	beta.hat	se	Z.value	p.value
##	[1,]	0.21748927	0.2215	0.3	0.302768613	0.06128614	4.9402457	9.144468e-07
##	[2,]	0.06972117	0.0610	0.3	0.339750083	0.10265888	3.3095052	9.680458e-04
##	[3,]	0.36935781	0.3780	0.3	0.293808185	0.04966138	5.9162303	4.522428e-09
##	[4,]	0.34596068	0.3480	0.3	0.385230175	0.05215124	7.3867879	3.171951e-13
##	[5,]	0.16243508	0.1695	0.3	0.375489557	0.06583105	5.7038366	1.542737e-08
##	[6,]	0.18502467	0.1995	0.0	0.060458207	0.06209435	0.9736507	3.304659e-01
##	[7,]	0.31318998	0.3375	0.0	0.005293237	0.05178405	0.1022175	9.186046e-01
##	[8,]	0.20006021	0.2020	0.0	-0.024170068	0.06322464	-0.3822888	7.023286e-01
##	[9,]	0.32990538	0.3360	0.0	-0.054514012	0.05360346	-1.0169868	3.094062e-01
##	[10,]	0.29562285	0.2905	0.0	-0.050072695	0.05552933	-0.9017342	3.674156e-01
##	[11,]	0.44590808	0.4445	0.0	0.030738418	0.05103795	0.6022660	5.471339e-01
##	[12,]	0.36809363	0.3745	0.0	-0.030640735	0.04981833	-0.6150494	5.386623e-01
##	[13,]	0.37938767	0.3750	0.0	-0.082596631	0.05093570	-1.6215863	1.052078e-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

A milestone Quiz

Live Quiz 7: compared with the baseline model, if $\text{ex.snp.true}=50$ the AUC of say PRS.gw will

- A: increase
- B: decrease
- C: ~same

(Locus) heterogeneity: $\text{my.nsnp.true} \neq \text{ex.nsnp.true}$,
 ex.nsnp.true increased to 50

```
# external data
ex.nsnp.true=50 # HERE is the only change
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] 0.623

## [1] 0.012 0.004 0.016 0.015 0.009 0.010 0.015 0.011 0.015 0.014 0.017 0.016
## [13] 0.016 0.017 0.013 0.015 0.017 0.009 0.012 0.004 0.016 0.017 0.008 0.015
## [25] 0.017 0.016 0.005 0.012 0.012 0.015 0.012 0.011 0.008 0.007 0.014 0.017
## [37] 0.008 0.017 0.004 0.017 0.013 0.013 0.016 0.011 0.012 0.009 0.005 0.017
## [49] 0.016 0.006
```

Total and SNP h^2 of my model

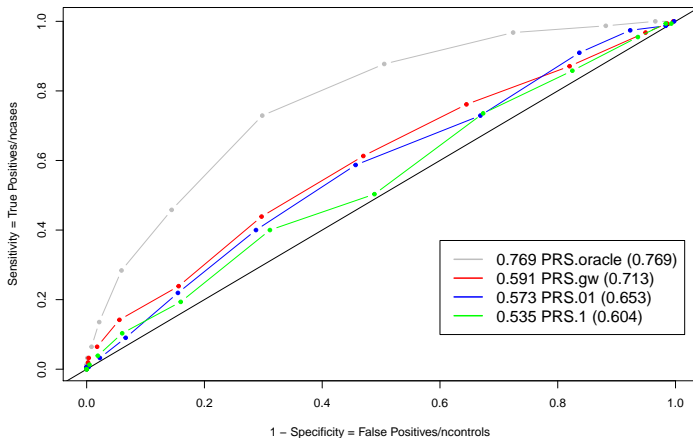
```
## [1] 0.243

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

(Locus) heterogeneity: ex.nsnp.true increased to 50

```
# 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  13  3  10  
## [2,] 1e-02  80  6  74  
## [3,] 1e-01 534  9 525
```


(Locus) heterogeneity: ex.nsnp.true increased to 50

```
ex.sumstat[c(1:10,49,50,51,52),] # the actual 1:14 rows
```

##		MAF	MAF.hat	beta	beta.hat	se	Z.value	p.value
##	[1,]	0.21748927	0.2215	0.3	0.21788417	0.09072936	2.4014736	1.651122e-02
##	[2,]	0.06972117	0.0610	0.3	0.60489133	0.15019527	4.0273661	6.068227e-05
##	[3,]	0.36935781	0.3780	0.3	0.16176083	0.07393591	2.1878521	2.891170e-02
##	[4,]	0.34596068	0.3480	0.3	0.45140665	0.07725190	5.8433077	6.922784e-09
##	[5,]	0.16243508	0.1695	0.3	0.12825792	0.09804262	1.3081853	1.911117e-01
##	[6,]	0.18502467	0.1995	0.3	0.40532713	0.09021956	4.4926749	7.855419e-06
##	[7,]	0.31318998	0.3375	0.3	0.33538343	0.07521510	4.4589904	9.168462e-06
##	[8,]	0.20006021	0.2020	0.3	0.15251450	0.09262311	1.6466140	9.995224e-02
##	[9,]	0.32990538	0.3360	0.3	0.20948171	0.07838985	2.6723066	7.655972e-03
##	[10,]	0.29562285	0.2905	0.3	0.35472150	0.08070992	4.3950176	1.226195e-05
##	[11,]	0.39726211	0.3895	0.3	0.32145485	0.07312508	4.3959589	1.220993e-05
##	[12,]	0.09839149	0.1045	0.3	0.43233264	0.11790384	3.6668243	2.585398e-04
##	[13,]	0.08564551	0.0800	0.0	-0.20429334	0.13410654	-1.5233660	1.279840e-01
##	[14,]	0.24525025	0.2390	0.0	-0.06334466	0.08518101	-0.7436476	4.572647e-01

```
my.data$my.sumstat[1:12,]
```

##		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

Understanding the 'disturbing' results, e.g. at 10^{-5}

10 ex.data & 10 my.data: 6 Ps in ex.data, all 6 are TPs in my.data

50 ex.data & 10 my.data: only 13 Ps in ex.data, and only 3 are TPs in my.data

50 causal SNPs leads to higher total heritability

$$(\text{narrow}) h^2 = \frac{V_G}{V_G + V_e} = \frac{\sum_j^{50} \beta_j^2 \text{Var}(G_j)}{\text{Var}(Y)} = \frac{\sum_j^{50} \beta_j^2 2p_j(1-p_j)}{\sum_j^{50} \beta_j^2 2p_j(1-p_j) + \sigma^2}.$$

BUT, smaller SNP heritability:

$$(\text{narrow}) h_j^2 = \frac{\beta_j^2 2p_j(1-p_j)}{\sum_j^{50} \beta_j^2 2p_j(1-p_j) + \sigma^2}.$$

Total and SNP h^2 of ex.model

```
## [1] 1.650 1.000 0.623
```

```
## [1] 0.012 0.004 0.016 0.015 0.009 0.010 0.015 0.011 0.015 0.014 0.017 0.016
```

```
## [13] 0.016 0.017 0.013 0.015 0.017 0.009 0.012 0.004
```

Total and SNP h^2 of my.model (the same as the baseline ex.model)

```
## [1] 0.321 1.000 0.243
```

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

Will a larger n help?

(Not dramatically, as although it helps identify more `ex.nsnp.true`, many of the `ex.nsnp.true` are NOT `my.nsnp.true`)

```
# external data
ex.nsnp.true=50 # HERE is the change & ex.nsample also increased
ex.beta.true=0.3
ex.nsnp=5000; ex.nsample=2000; 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] 0.623

## [1] 0.012 0.004 0.016 0.015 0.009 0.010 0.015 0.011 0.015 0.014 0.017 0.016
## [13] 0.016 0.017 0.013 0.015 0.017 0.009 0.012 0.004 0.016 0.017 0.008 0.015
## [25] 0.017 0.016 0.005 0.012 0.012 0.015 0.012 0.011 0.008 0.007 0.014 0.017
## [37] 0.008 0.017 0.004 0.017 0.013 0.013 0.016 0.011 0.012 0.009 0.005 0.017
## [49] 0.016 0.006
```

Total and SNP h^2 of my model (the true h^2 does not depend on n)

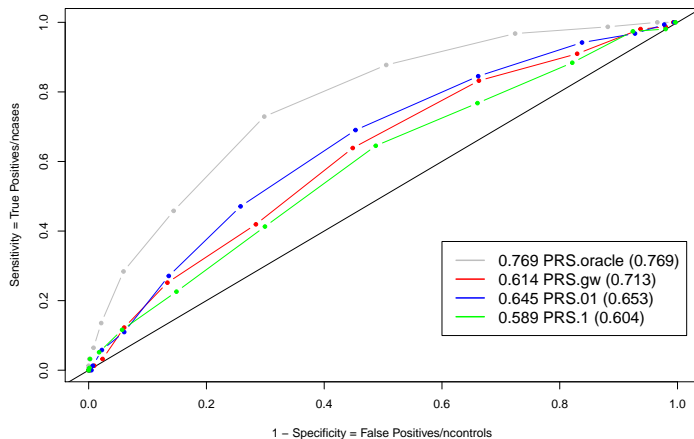
```
## [1] 0.243

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

ex.nsnp.true increased to 50 and ex.nsample to 2000

```
# 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  37  6  31  
## [2,] 1e-02  94 10  84  
## [3,] 1e-01 535 10 525
```

ex.nsnp.true increased to 50 and ex.nsample to 2000

```
ex.sumstat[c(1:10,49,50,51,52),] # the actual 1:14 rows
```

##		MAF	MAF.hat	beta	beta.hat	se	Z.value	p.value
##	[1,]	0.21748927	0.22250	0.3	0.30610269	0.06219016	4.9220436	9.266545e-07
##	[2,]	0.06972117	0.07800	0.3	0.25181781	0.09667408	2.6048121	9.260642e-03
##	[3,]	0.36935781	0.36475	0.3	0.22345457	0.05249734	4.2564931	2.172716e-05
##	[4,]	0.34596068	0.34050	0.3	0.28927208	0.05460681	5.2973626	1.304287e-07
##	[5,]	0.16243508	0.16675	0.3	0.31562277	0.07029548	4.4899442	7.530325e-06
##	[6,]	0.18502467	0.18600	0.3	0.27003449	0.06574550	4.1072697	4.165068e-05
##	[7,]	0.31318998	0.31050	0.3	0.37931162	0.05483976	6.9167267	6.199325e-12
##	[8,]	0.20006021	0.20175	0.3	0.27267812	0.06497857	4.1964314	2.830332e-05
##	[9,]	0.32990538	0.34025	0.3	0.34960140	0.05360905	6.5213129	8.803090e-11
##	[10,]	0.29562285	0.29000	0.3	0.28605347	0.05624801	5.0855746	4.007568e-07
##	[11,]	0.39726211	0.39875	0.3	0.35883432	0.05155750	6.9598862	4.600627e-12
##	[12,]	0.09839149	0.09575	0.3	0.31067196	0.08683432	3.5777554	3.548094e-04
##	[13,]	0.08564551	0.08850	0.0	-0.12752862	0.09056357	-1.4081669	1.592372e-01
##	[14,]	0.24525025	0.24825	0.0	0.03852528	0.05891236	0.6539422	5.132243e-01

```
my.data$my.sumstat[1:12,]
```

##		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

What does this tell us?

- ▶ More truly associated SNPs lead to high trait heritability, but it does not translate to improved power of (standard) GWAS or PRS performance.
- ▶ Power of standard GWAS depends on the SNP heritability.
- ▶ Thus, using ex.data for a trait with high trait heritability does necessarily lead to better PRs performance in my.data.
- ▶ Heterogeneity (ex.data and my.data not matching) is damaging and can lead to counter-intuitive results.
- ▶ e.g. Larger n may not be better if heterogeneity is not properly addressed.

Should study more complex and realistic settings:

Population heterogeneity likely lead to simultaneous heterogeneities in MAF and β , and locus heterogeneity.

(Codes need to be upgraded if we were to study these more complex settings.)

BUT, with the **deeper** understanding of all the building blocks, do we really need it?!

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

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`

What's next? There are some limited slides on LD, but **here is a quiz about your current understanding of LD and its impact on PRS.**

- ▶ If we were to duplicate (in perfect LD with $r^2 = 1$) each of the 10 truly associated SNPs,
- ▶ We then proceed as if all SNPs are independent, i.e. LD-unaware PRS
- ▶ What would be the AUC of our PRS.oracle, PRS.gw, PRS.01, and PRS.1?