

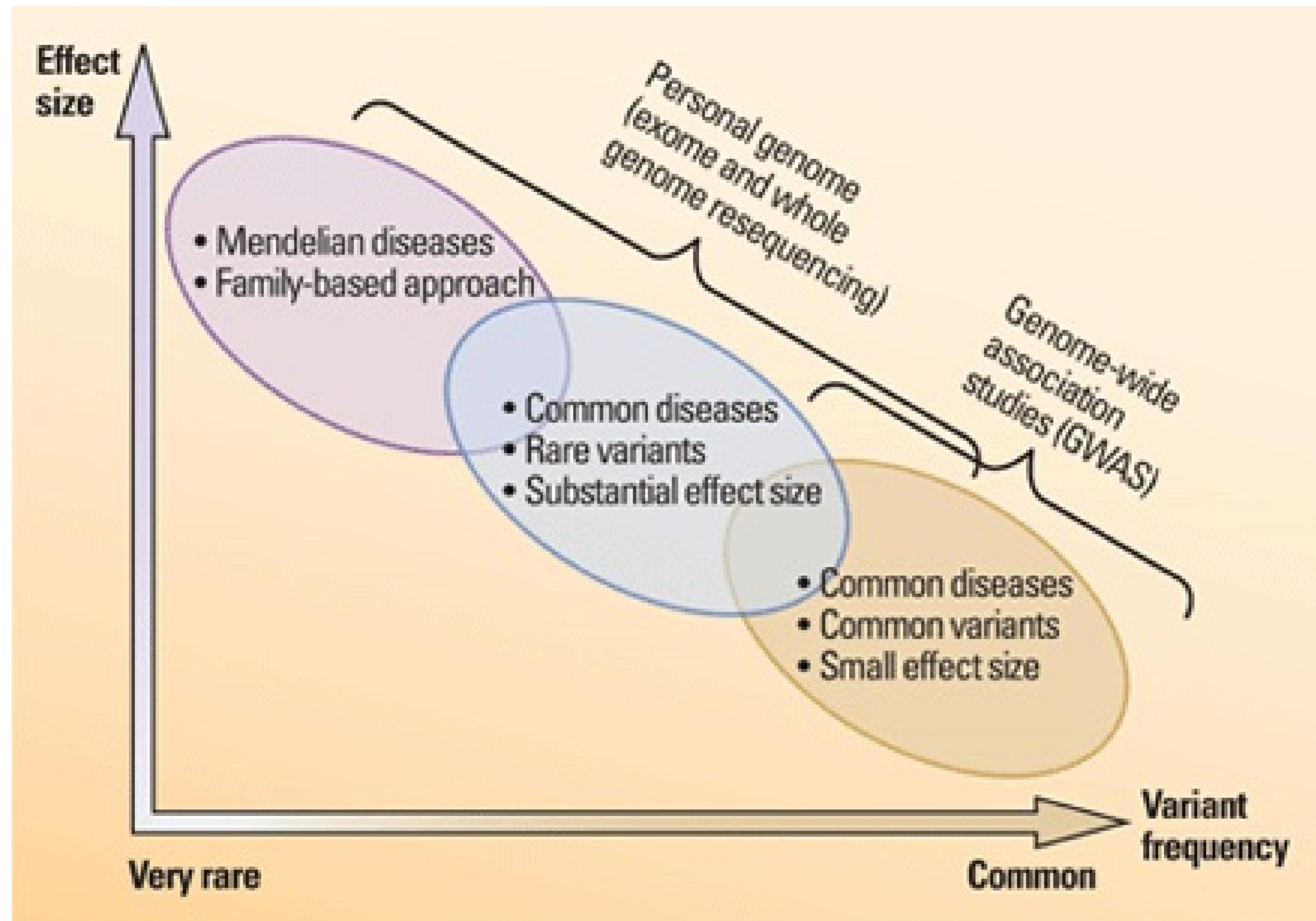
Predicting complex diseases: performance and robustness

Florian Privé, Hugues Aschard, Michael G.B. Blum

RECOMB-Genetics 2018

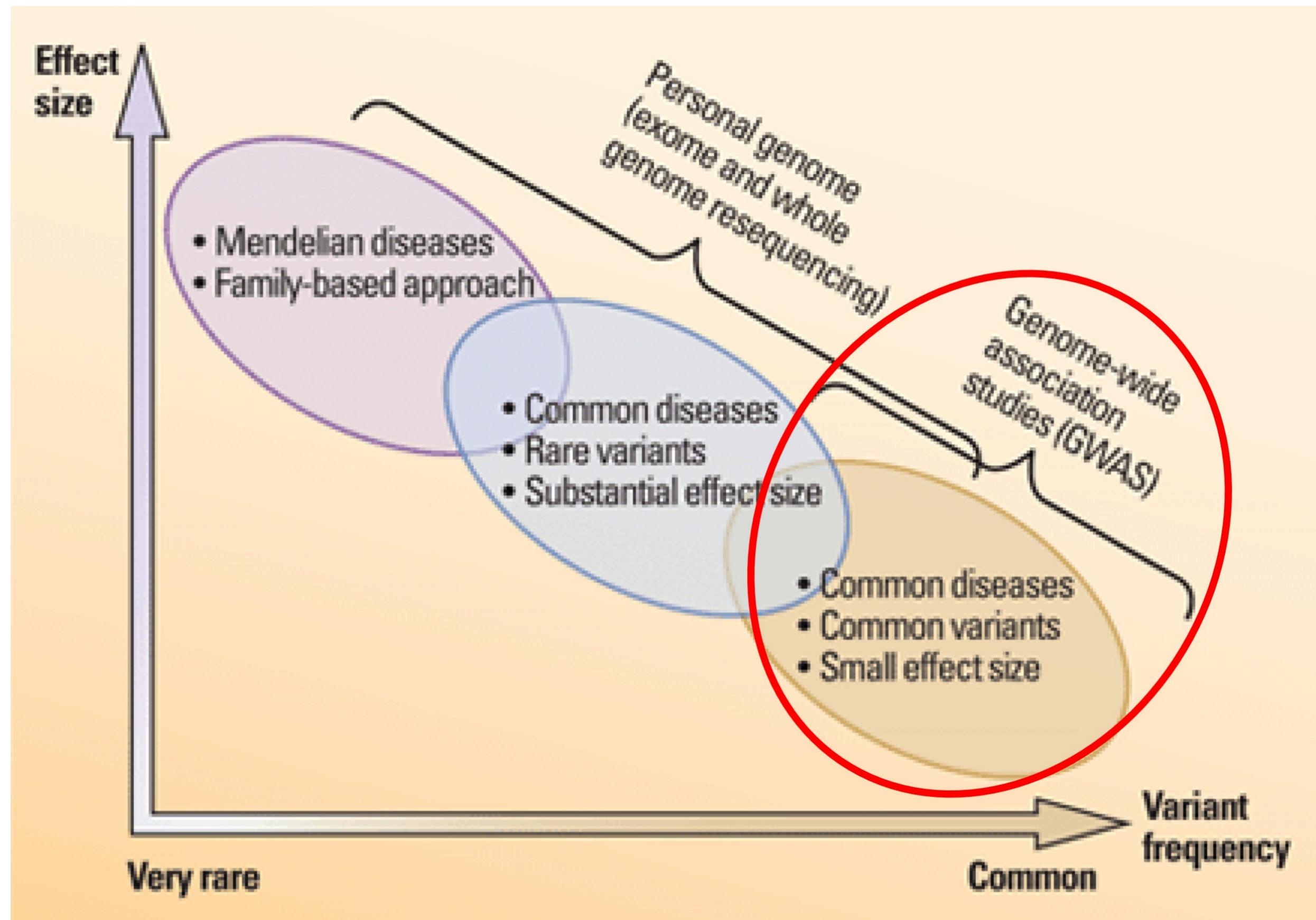
Introduction

Disease architectures



Source: 10.1126/science.338.6110.1016

Disease architectures



How to derive a genetic risk score for common diseases based on common variants with small effects?

Interest in prediction: polygenic risk scores (PRS)

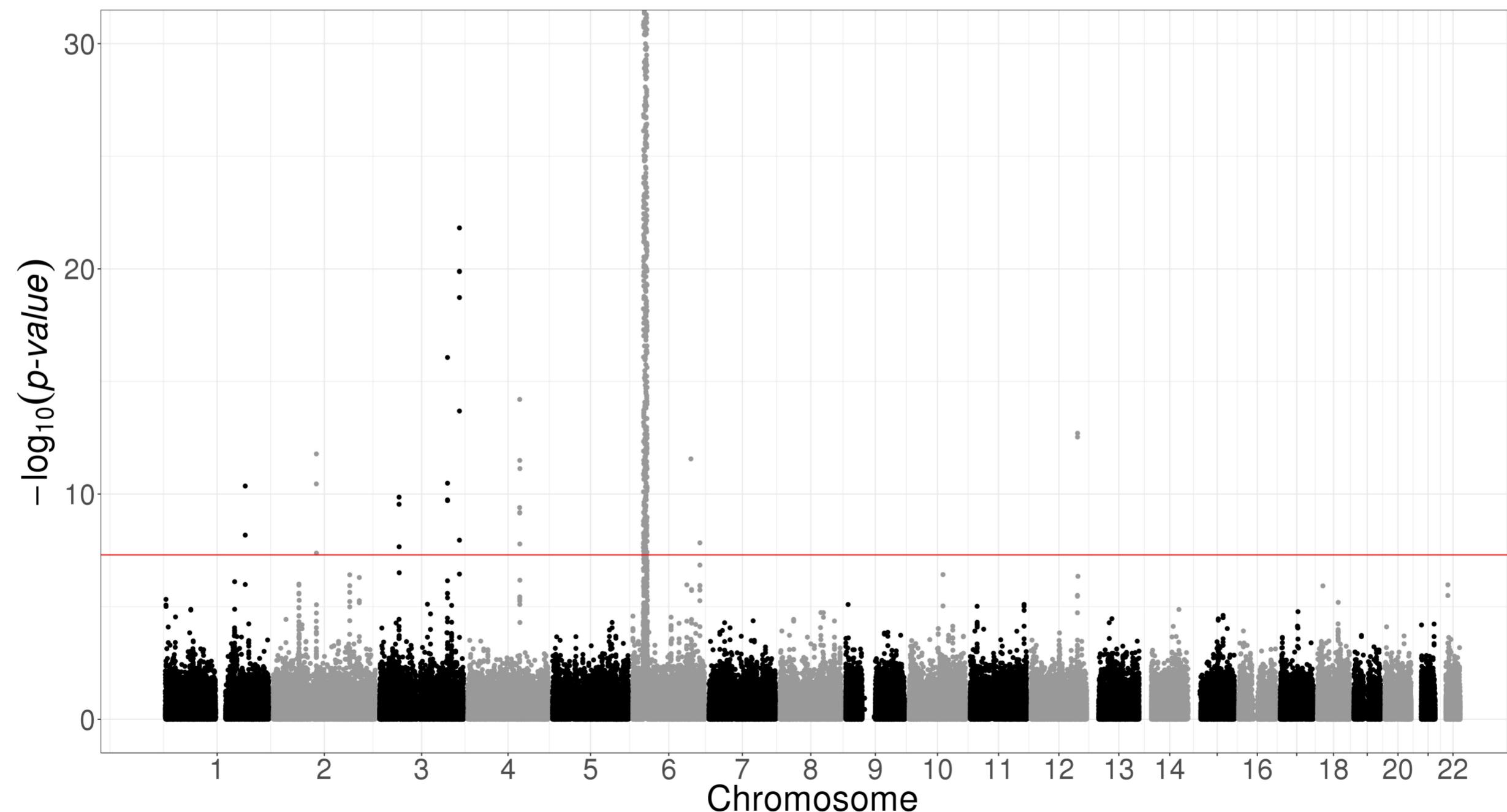
- Wray, Naomi R., Michael E. Goddard, and Peter M. Visscher. "**Prediction of individual genetic risk** to disease from genome-wide association studies." *Genome research* 17.10 (2007): 1520-1528.
- Wray, Naomi R., et al. "**Pitfalls of predicting complex traits** from SNPs." *Nature Reviews Genetics* 14.7 (2013): 507.
- Dudbridge, Frank. "**Power and predictive accuracy of polygenic risk scores.**" *PLoS genetics* 9.3 (2013): e1003348.
- Chatterjee, Nilanjan, Jianxin Shi, and Montserrat García-Closas. "**Developing and evaluating polygenic risk prediction models** for stratified disease prevention." *Nature Reviews Genetics* 17.7 (2016): 392.
- Martin, Alicia R., et al. "**Human demographic history impacts genetic risk prediction** across diverse populations." *The American Journal of Human Genetics* 100.4 (2017): 635-649.

Still a gap between current predictions and clinical utility.
Need more optimal predictions + larger sample sizes.

Standard PRS - part 1: estimating effects

Genome-wide association studies (GWAS)

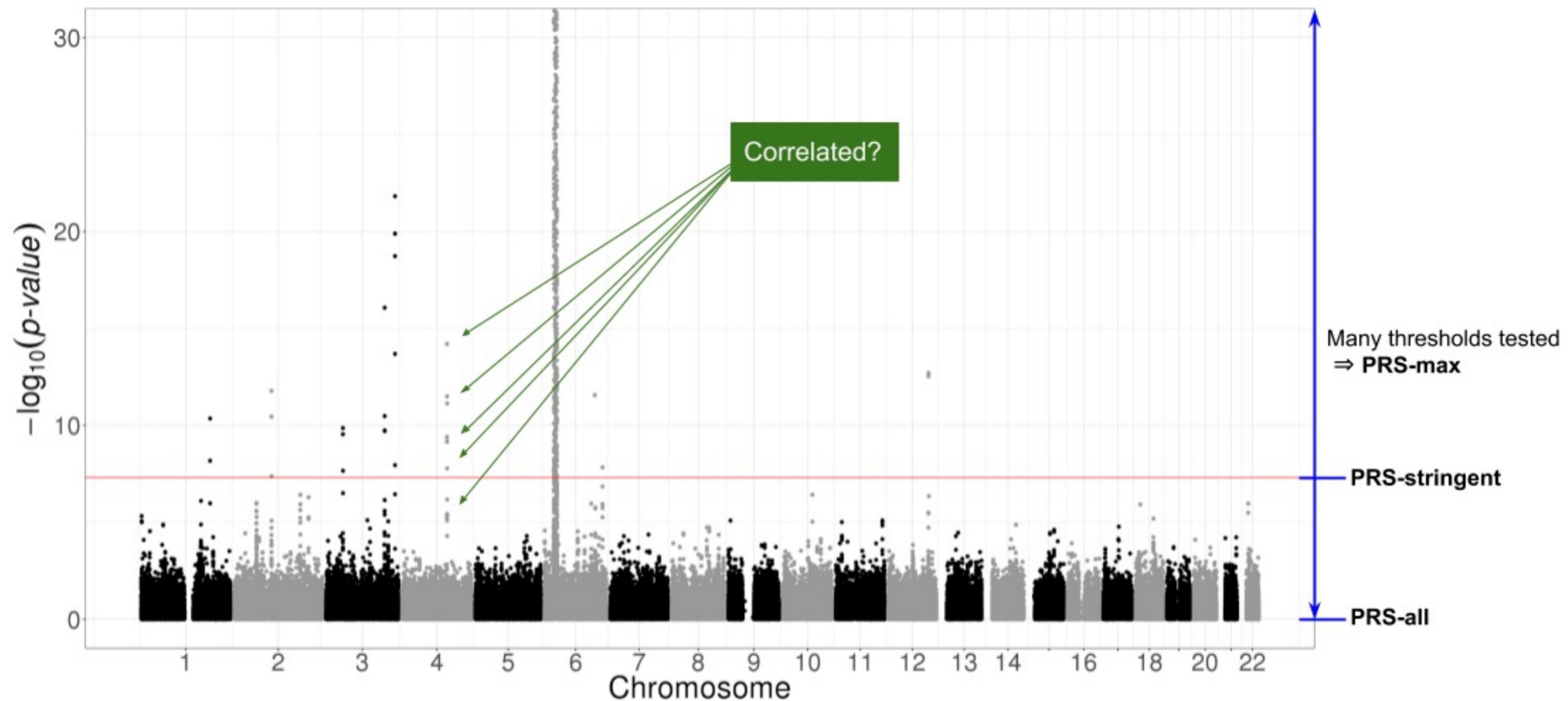
In a GWAS, each single-nucleotide polymorphism (SNP) is tested **independently**, resulting in one **effect size** $\hat{\beta}$ and one **p-value** p for each SNP.



Easy combining: $PRS_i = \sum \hat{\beta}_j \cdot G_{i,j}$

Standard PRS - part 2: restricting predictors

Clumping + Thresholding ("C+T" or just "PRS")



$$PRS_i = \sum_{\substack{j \in S_{\text{clumping}} \\ p_j < p_T}} \hat{\beta}_j \cdot G_{i,j}$$

A more optimal approach to computing PRS?

In C+T: weights learned independently and heuristics for correlation and regularization.

Statistical learning

- joint models of all SNPs at once
- use regularization to account for correlated and null effects
- already proved useful in the litterature (Abraham et al. 2013; Okser et al. 2014; Spiliopoulou et al. 2015)

Our contribution

- a memory- and computation-efficient implementation to be used for biobank-scale data
- an automatic choice of the regularization hyper-parameter
- a comprehensive comparison for different disease architectures

Methods

Penalized Logistic Regression

$$\operatorname{argmin}_{\beta_0, \beta}(\lambda, \alpha) \left\{ \underbrace{- \sum_{i=1}^n (y_i \log(p_i) + (1 - y_i) \log(1 - p_i))}_{\text{Loss function}} + \underbrace{\lambda \left((1 - \alpha) \frac{1}{2} \|\beta\|_2^2 + \alpha \|\beta\|_1 \right)}_{\text{Penalization}} \right\}$$

-
- $p_i = 1 / (1 + \exp(-(\beta_0 + x_i^T \beta)))$
 - x is denoting the genotypes and covariates (e.g. principal components),
 - y is the disease status we want to predict,
 - λ is a regularization parameter that needs to be determined and
 - α determines relative parts of the regularization $0 \leq \alpha \leq 1$.

Efficient algorithm

- sequential strong rules for discarding predictors in lasso-type problems
(Tibshirani et al. 2012; Zeng et al. 2017)
- implemented in our R package {bigstatsr}

Efficient analysis of large-scale genome-wide data
with two R packages: **bigstatsr** and **bigsnpr** 

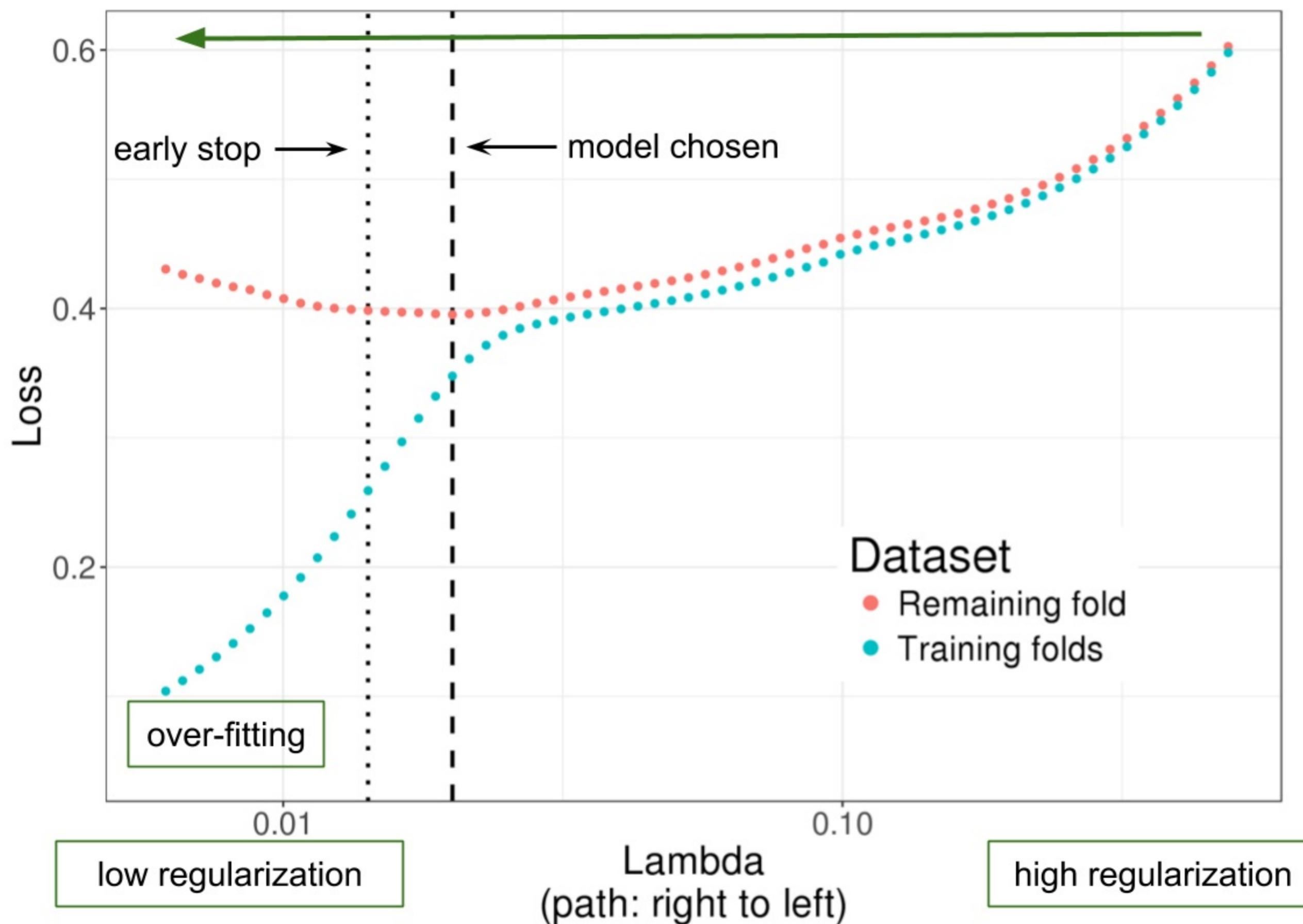
Florian Privé , Hugues Aschard, Andrey Ziyatdinov, Michael G B Blum 

Bioinformatics, bty185, <https://doi.org/10.1093/bioinformatics/bty185>

Our implementation

- uses memory-mapping to matrices stored on disk
- functioning choice of the hyper-parameter λ
- early stopping criterion

Choice of the hyper-parameter λ



Comprehensive simulations: varying many parameters

Simulation models

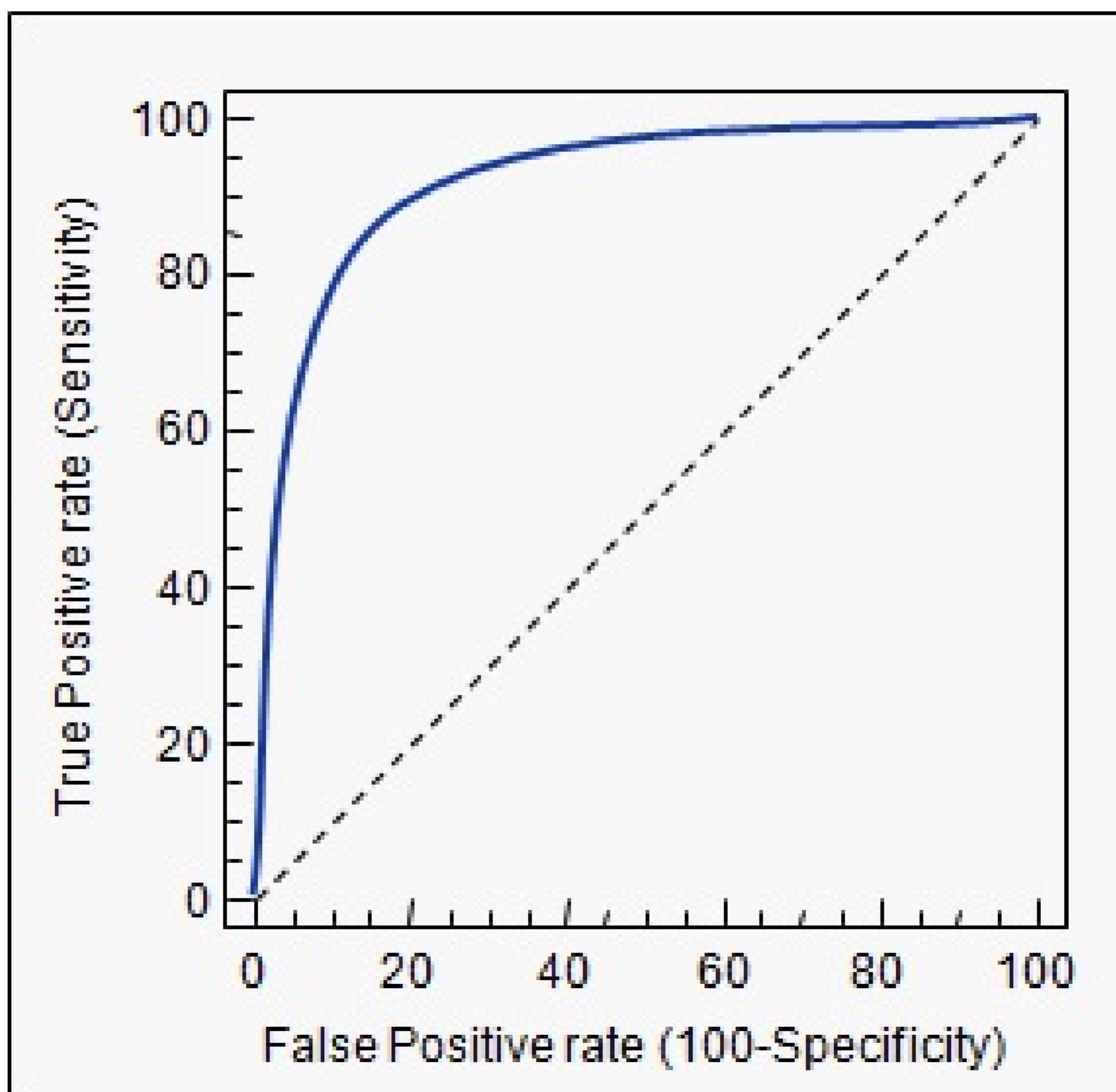
Numero of scenario	Dataset	Size of training set	Causal SNPs (number and location)	Distribution of effects	Heritability	Simulation model
1	All 22 chromosomes	6000	30 in HLA 30 in all 300 in all 3000 in all	Gaussian Laplace	0.8 0.5	simple fancy
2	Chromosome 6 only	-	-	-	-	simple
3	All 22 chromosomes	1000 2000 3000 4000 5000	300 in all	-	-	-

Methods

- PRS:
 - PRS-all (no p-value thresholding),
 - PRS-stringent (GWAS threshold of significance) and
 - PRS-max (best prediction for all thresholds, considered as an upper-bound)
- logit-simple: penalized logistic regression with automatic of selection λ

Predictive performance measures

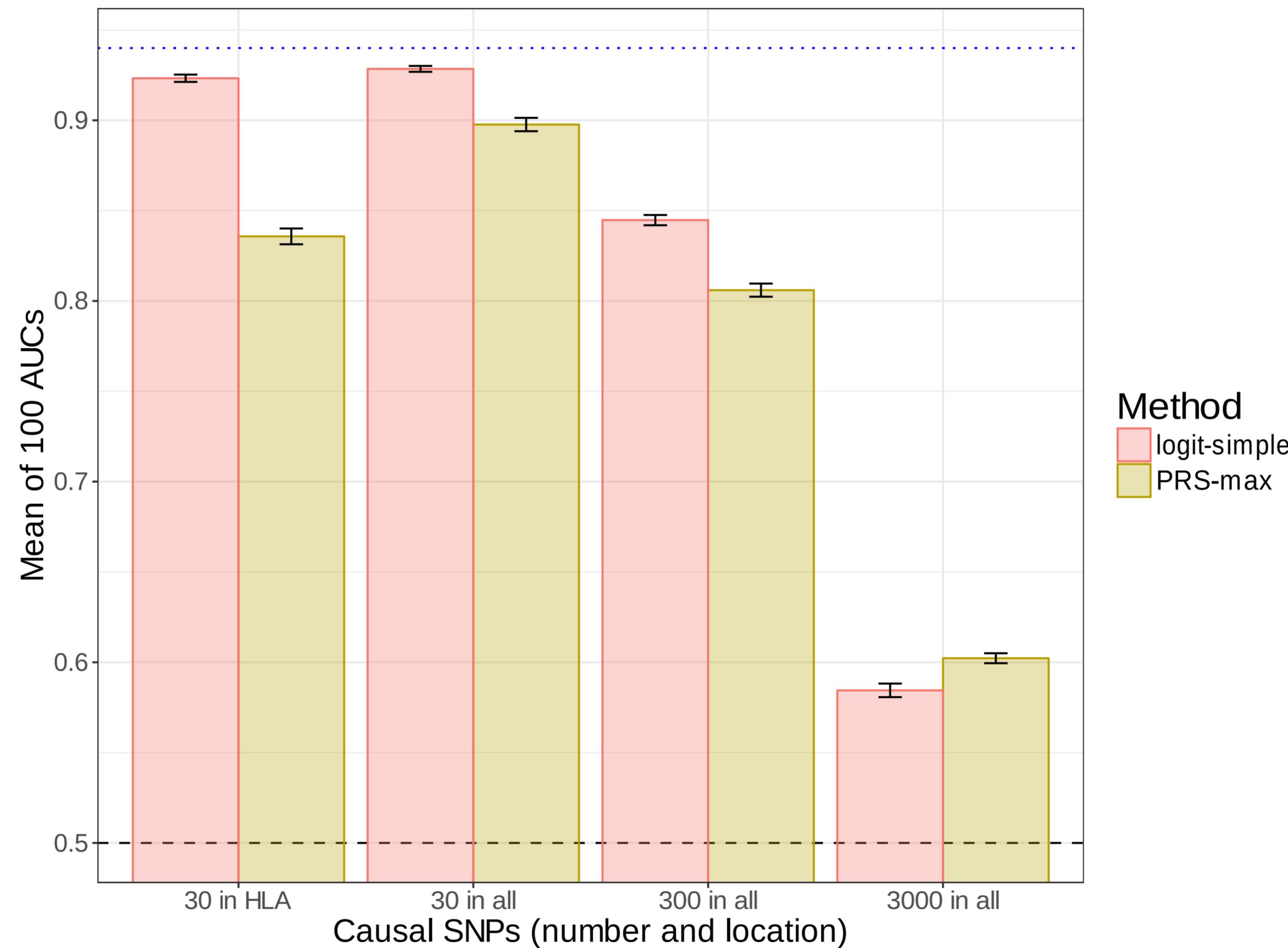
AUC (Area Under the ROC Curve) and partial AUC (FPR < 10%) are used.



$$\text{AUC} = P(S_{\text{case}} > S_{\text{control}})$$

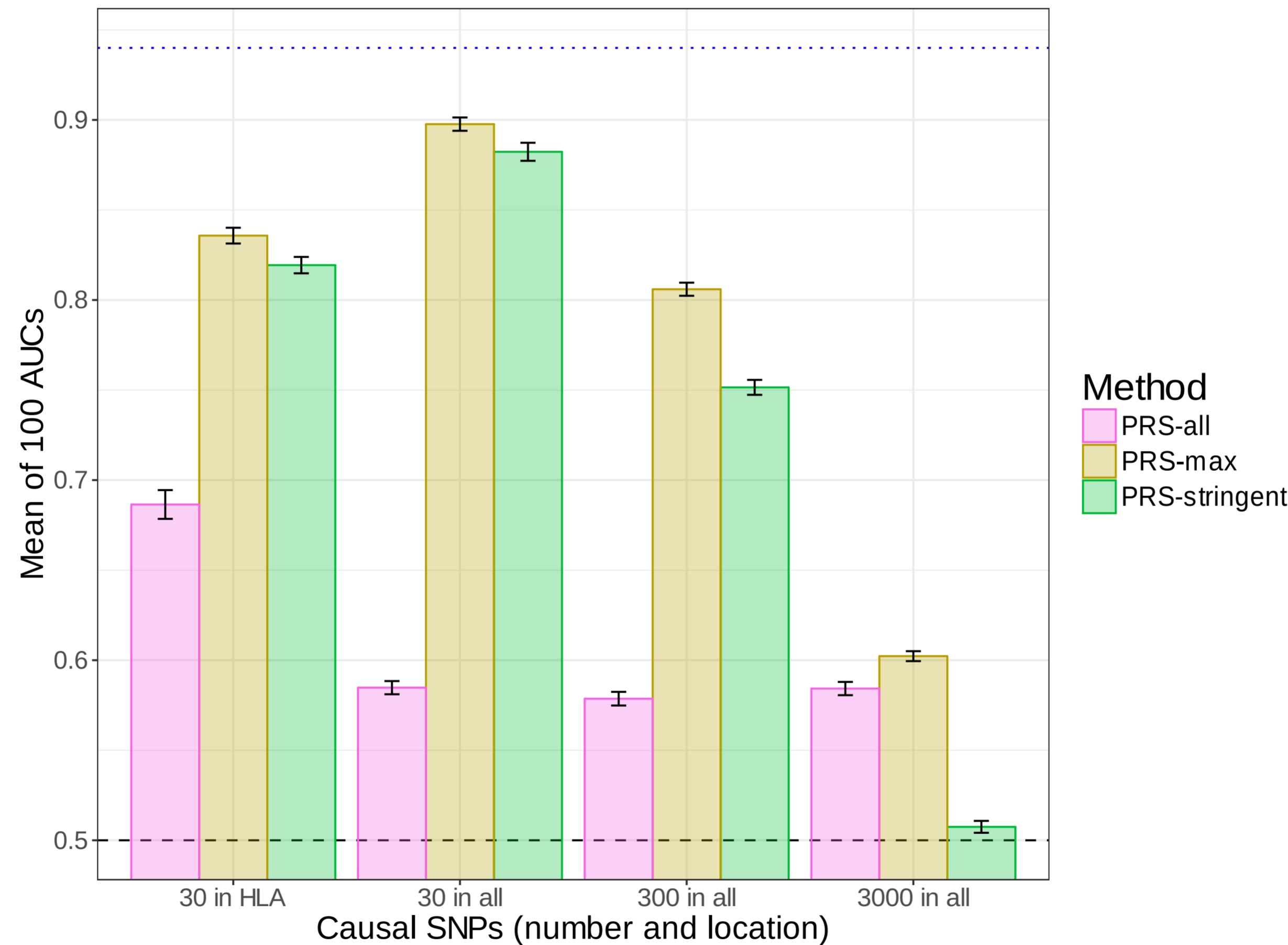
Results

Higher predictive performance with logit-simple



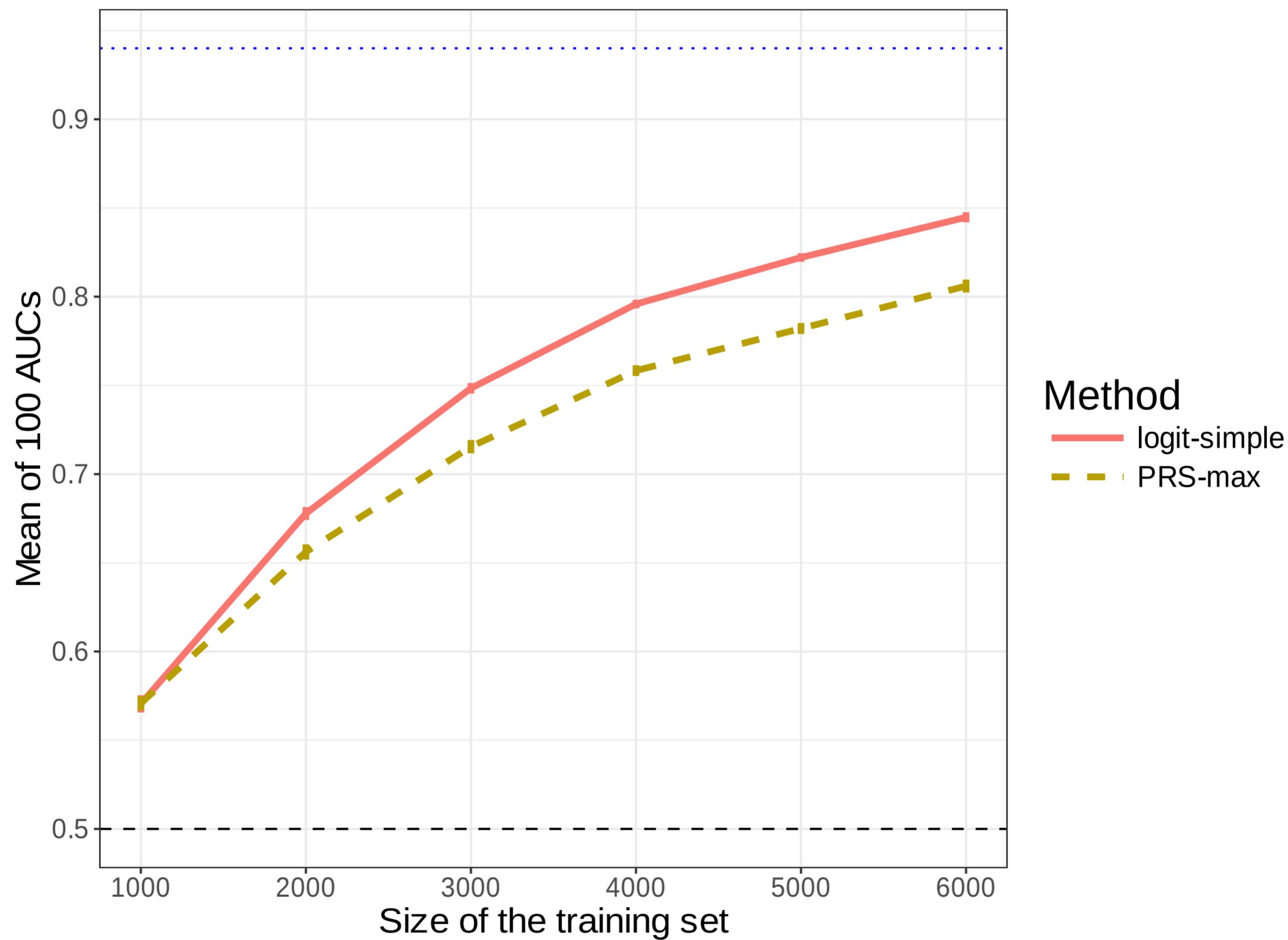
Penalized logistic regression provides higher predictive performance in the cases that matter, especially when there are correlated variables.

Predictive performance of C+T method varies with threshold



Recall that prediction of PRS-max is an upper-bound of the prediction provided by the C+T method.

Prediction with logit-simple is improving faster



Performance of methods improve with larger sample size. Yet, penalized logistic regression is improving faster than the C+T method.

Real data

Celiac disease

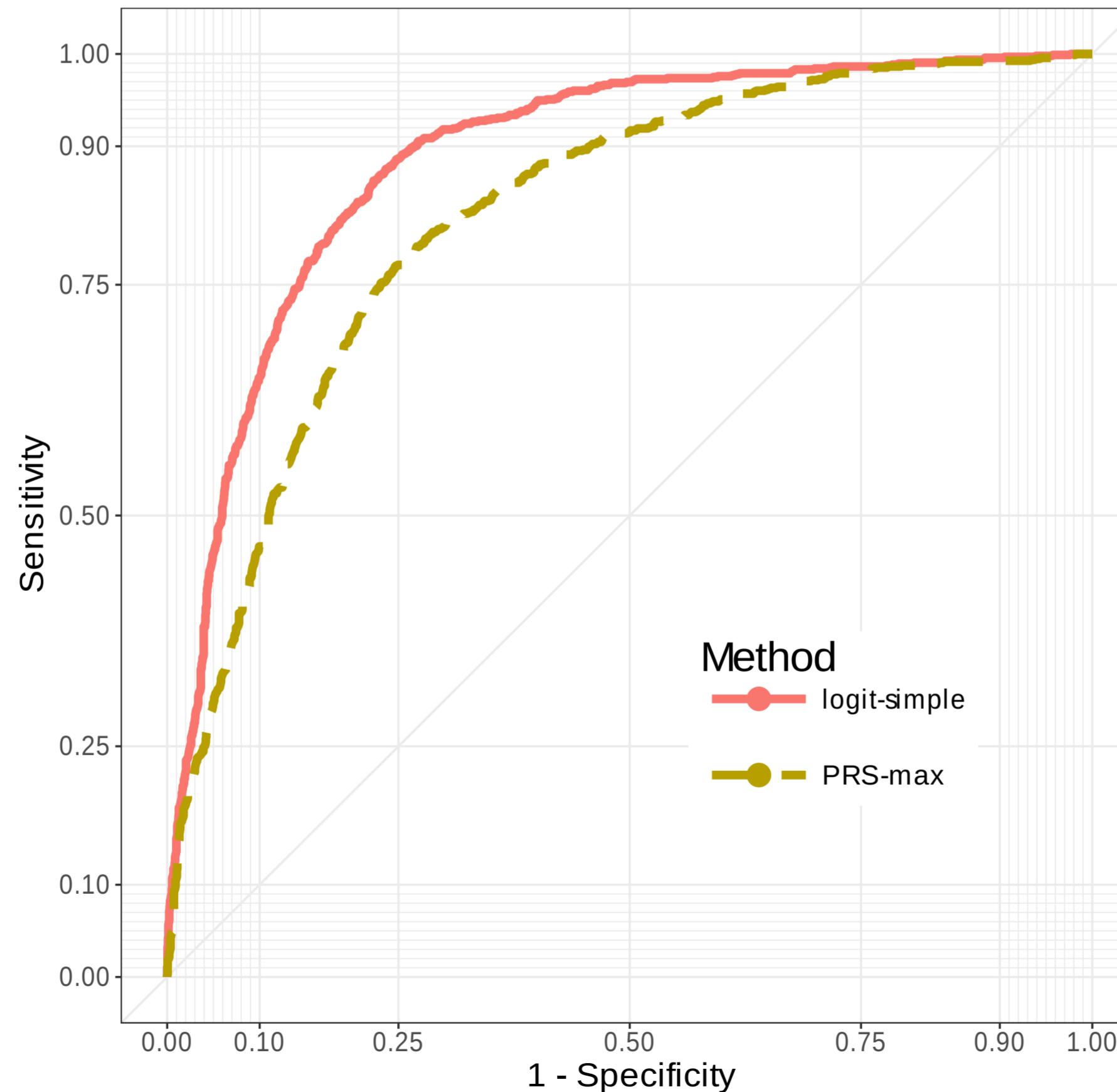
- intolerance to gluten
- only treatment: gluten-free diet
- heritability: 57-87% (Nisticò et al. 2006)
- prevalence: 1-6%

Case-control study for the celiac disease (WTCCC, Dubois et al. 2010)

- ~15,000 individuals
- ~280,000 SNPs
- ~30% cases

Results: real Celiac phenotypes

Method	AUC	pAUC	# predictors	Execution time (s)
PRS-max	0.824 (0.000704)	0.0286 (0.00016)	9850 (781)	148 (0.414)
logit-simple	0.888 (0.000468)	0.0414 (0.000164)	3220 (62)	83.8 (1.27)



Discussion

Summary of our penalized regression as compared to the C+T method

- A more **optimal** approach for predicting complex diseases
- models that are **linear** and very **sparse**
- **very fast**
- **automatic choice** for the regularization parameter
- can be extended to capture also recessive and dominant effects

Prospects: future work using the UK Biobank

- use of external summary statistics to improve models
- better generalization to multiple populations
- integration of clinical and environmental data

Thanks!

Presentation: <https://privefl.github.io/thesis-docs/recomb18.html>

R package {bigstatsr}: <https://github.com/privefl/bigstatsr>

R package {bigsnpr}: <https://github.com/privefl/bigsnpr>



privefl



privefl

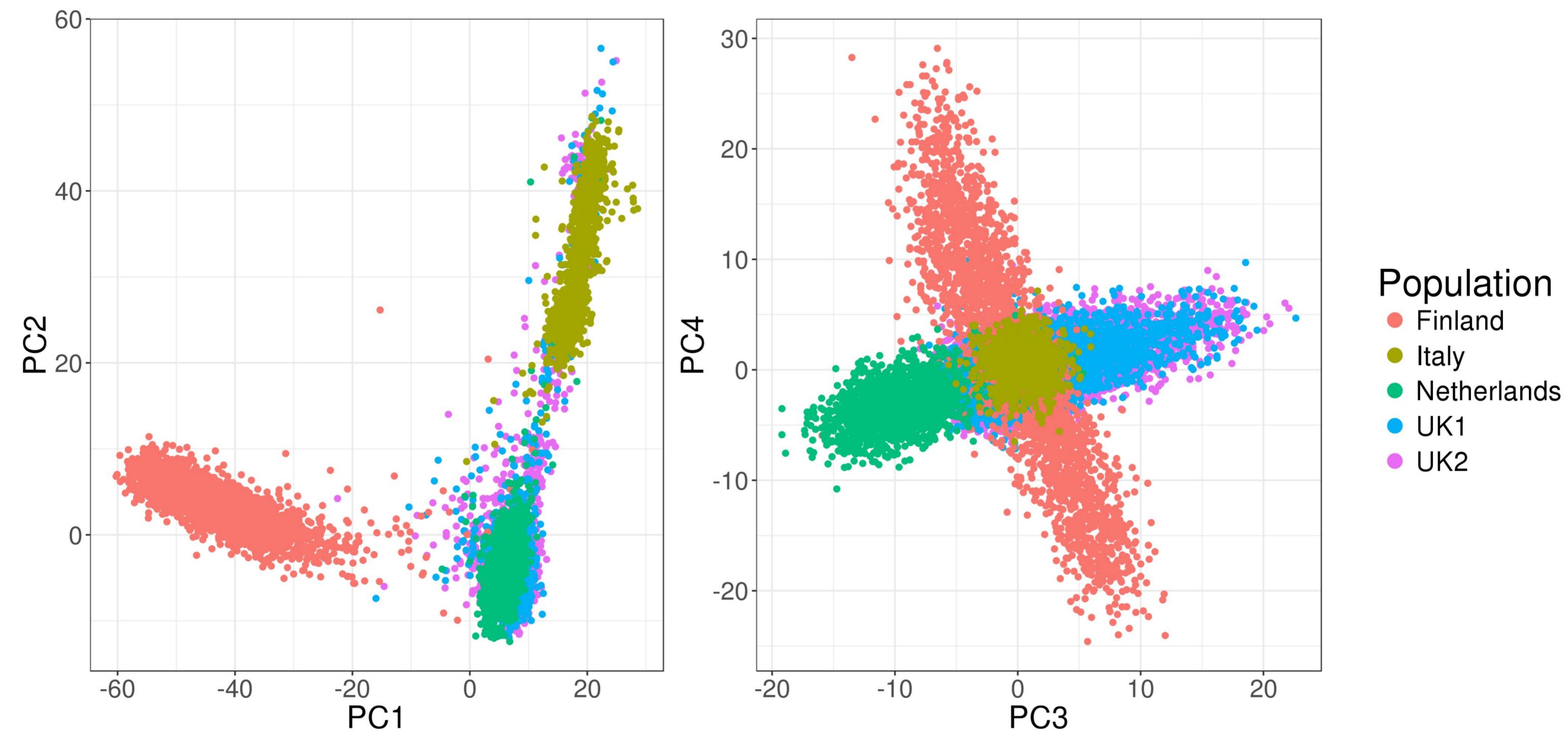


F. Privé

Slides created via the R package **xaringan**.

Real genotype data

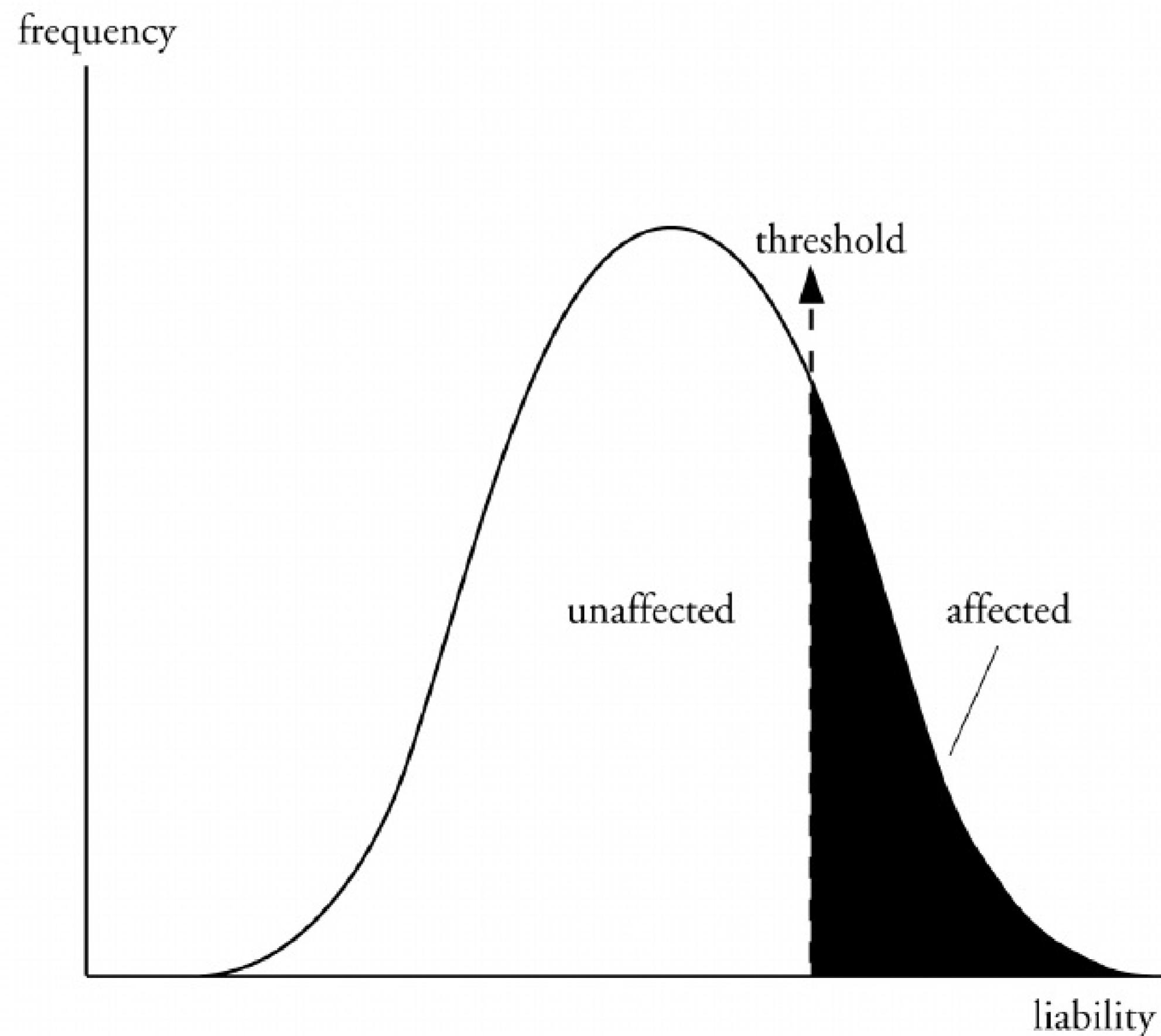
Use real data from a case-control study for the Celiac disease.



Keep only **controls** from the UK and **not deviating from the robust Malahanobis distance**.

Simulate new phenotypes

The liability-threshold model



Two models of liability

A "simple" model

$$y_i = \underbrace{\sum_{j \in S_{\text{causal}}} w_j \cdot \widetilde{G}_{i,j}}_{\text{genetic effect}} + \underbrace{\epsilon_i}_{\text{environmental effect}}$$

A "fancy" model

$$y_i = \underbrace{\sum_{j \in S_{\text{causal}}^{(1)}} w_j \cdot \widetilde{G}_{i,j}}_{\text{linear}} + \underbrace{\sum_{j \in S_{\text{causal}}^{(2)}} w_j \cdot \widetilde{D}_{i,j}}_{\text{dominant}} + \underbrace{\sum_{\substack{k=1 \\ j_1=e_k^{(3.1)} \\ j_2=e_k^{(3.2)}}}^{k=|S_{\text{causal}}^{(3.1)}|} w_{j_1} \cdot G_{i,j_1} \widetilde{G}_{i,j_2}}_{\text{interaction}} + \epsilon_i$$

-
- w_j are **weights** (generated with a Gaussian or a Laplace distribution)
 - $G_{i,j}$ is the **allele count** of individual i for SNP j
 - $D_{i,j} = \mathbf{1}\{G_{i,j} \neq 0\}$

Extension via feature engineering

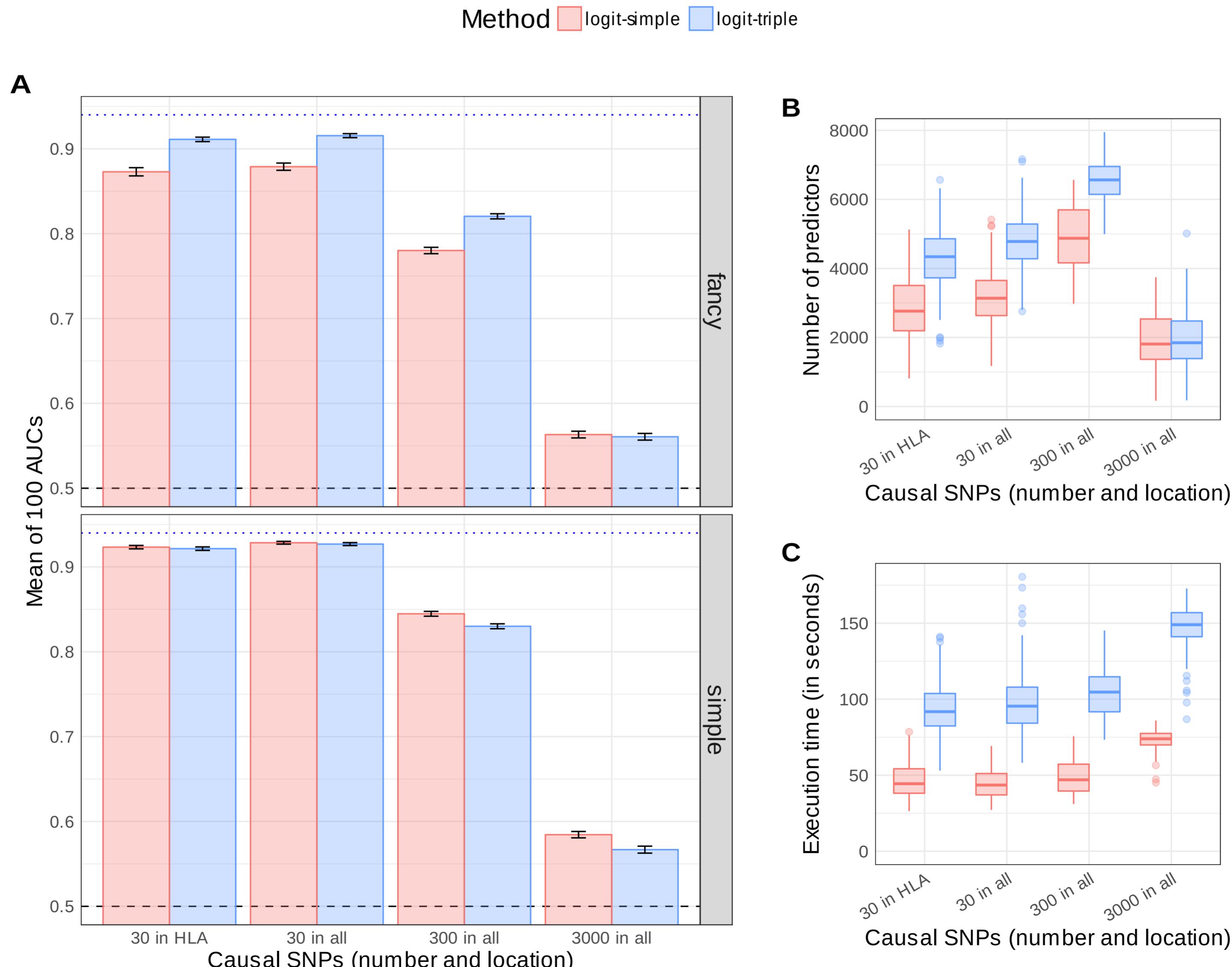
We construct a separate dataset with, for each SNP variable, two more variables coding for recessive and dominant effects.



	SNP1	SNP2		SNP1.1	SNP1.2	SNP1.3	SNP2.1	SNP2.2	SNP2.3	
[1.]	0	2		[1.]	0	0	0	2	1	1
[2.]	0	1		[2.]	0	0	0	1	1	0
[3.]	1	1		[3.]	1	1	0	1	1	0
[4.]	0	2		[4.]	0	0	0	2	1	1
[5.]	0	0		[5.]	0	0	0	0	0	0
[6.]	1	0		[6.]	1	1	0	0	0	0
[7.]	1	1		[7.]	1	1	0	1	1	0
[8.]	0	1		[8.]	0	0	0	1	1	0
[9.]	0	1		[9.]	0	0	0	1	1	0
[10.]	0	2		[10.]	0	0	0	2	1	1

We call these two methods "logit-simple" and "logit-triple".

Feature engineering improves prediction



Prediction with logit-simple is improving faster

