Module 12: Comp'l Pipeline for WGS

Association Tests

Ken Rice

University of Washington, TOPMed DCC

July 2018

Overview

This material comes in several parts:

Part One – background

- Primer* (reminder?) of significance testing – why we use this approach, what it tells us
- A little more about confounding (see earlier material on structure)
- Review of multiple testing

* *"A short informative review of a subject." Not the RNA/DNA version, or explosives, or undercoat*

Overview

Part Two – single variant association tests

- How tests work
- Adjusting for covariates; how and why
- Allowing for relatedness
- Trait transformation
- Issues of mis-specification and small-sample artefacts
- Binary outcomes (briefly)

– then a hands-on session, with discussion

Part Three – multiple variant association tests

- Burden tests, as a minor extension of single-variant methods
- Introduction to SKAT and why it's useful
- Some issues/extensions of SKAT (also see later material on variant annotation)

– then another hands-on session, with discussion

Overview

Part Four – faster/cheaper ways to compute

- Why is it slow?
- Exploiting sparsity
- Hand-coding projections
- FastSKAT, and related ideas

– no exercises for this material

Part 1: Background

About you: hands up please!

I have used:	P-values Linear regression Logistic regression Mixed models
I have experience with:	WGS/GWAS/expression data
I honestly understand p-values:	Yes/No (...*honestly!*)
I last took a stats course:	1/2/4/10+ years ago
$p < 5 \times 10^{-8}$ gets me into *Nature*:	Yes/No

Aims

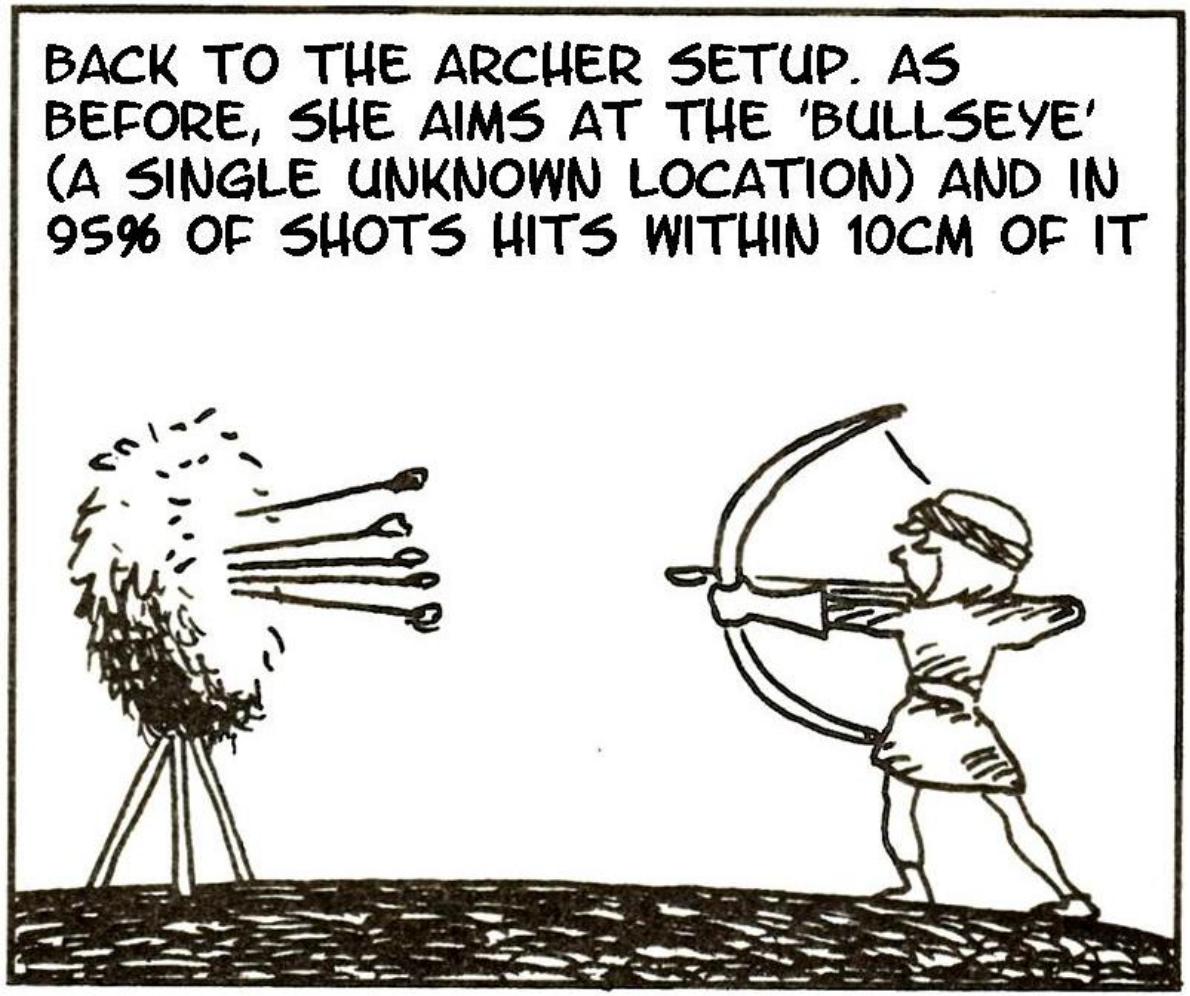
Teaching inference in $\approx$ 30 minutes is not possible. But using some background knowledge – and reviewing some important fundamental isses – we can better understand *why* certain analysis choices are good/not so good for WGS data.

- Hands-on material also shows you *how* to implement particular analyses – mostly familiar, if you've run regressions before
- Some advanced topics may be skipped – but see later sessions, and consult these slides back home, if you have problems
- There are several ways to think about this material – the version you'll see connects most directly with WGS analysis, *in my opinion*

Please interrupt! Questions are helpful, and welcome.

Testing

Before the formalities, an example to make you quiver;

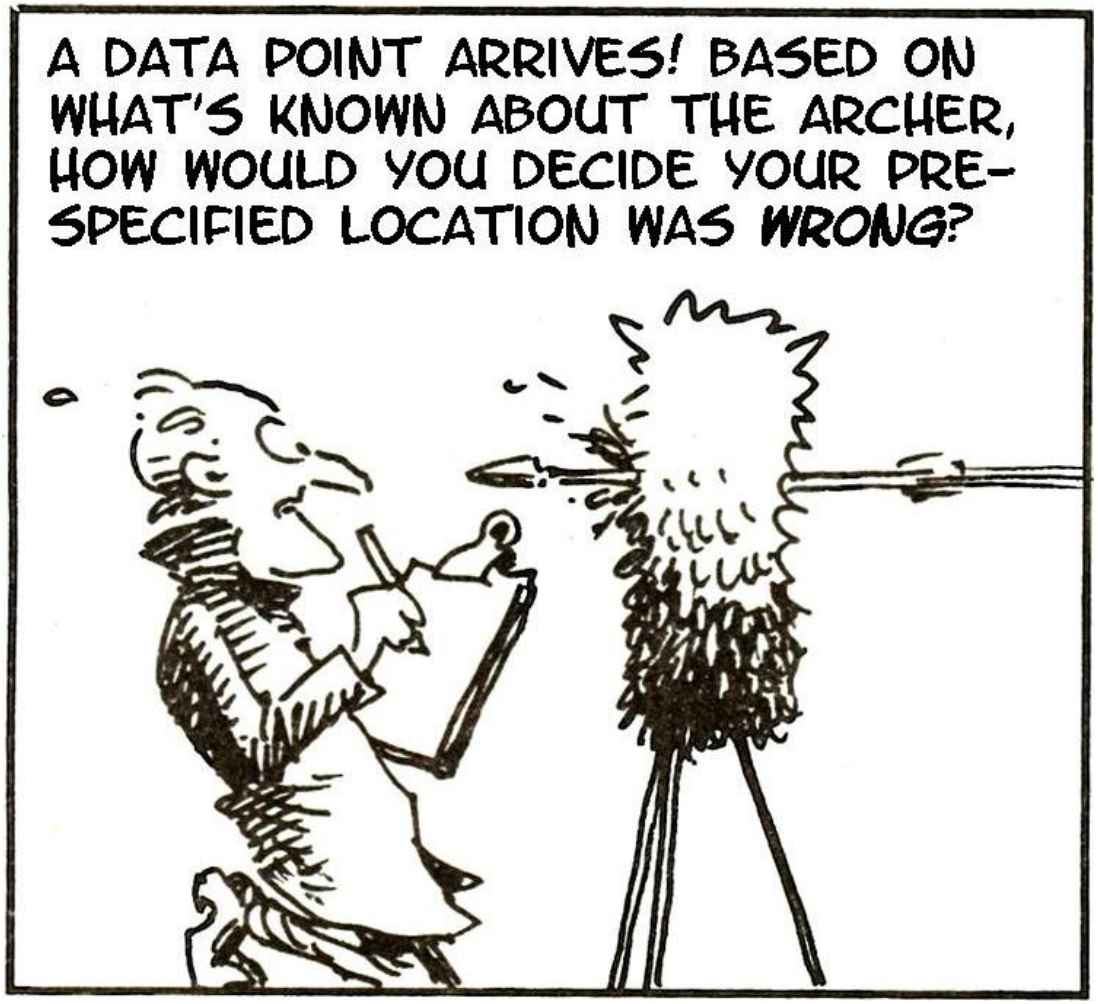


Testing

Before the formalities, an example to make you quiver;

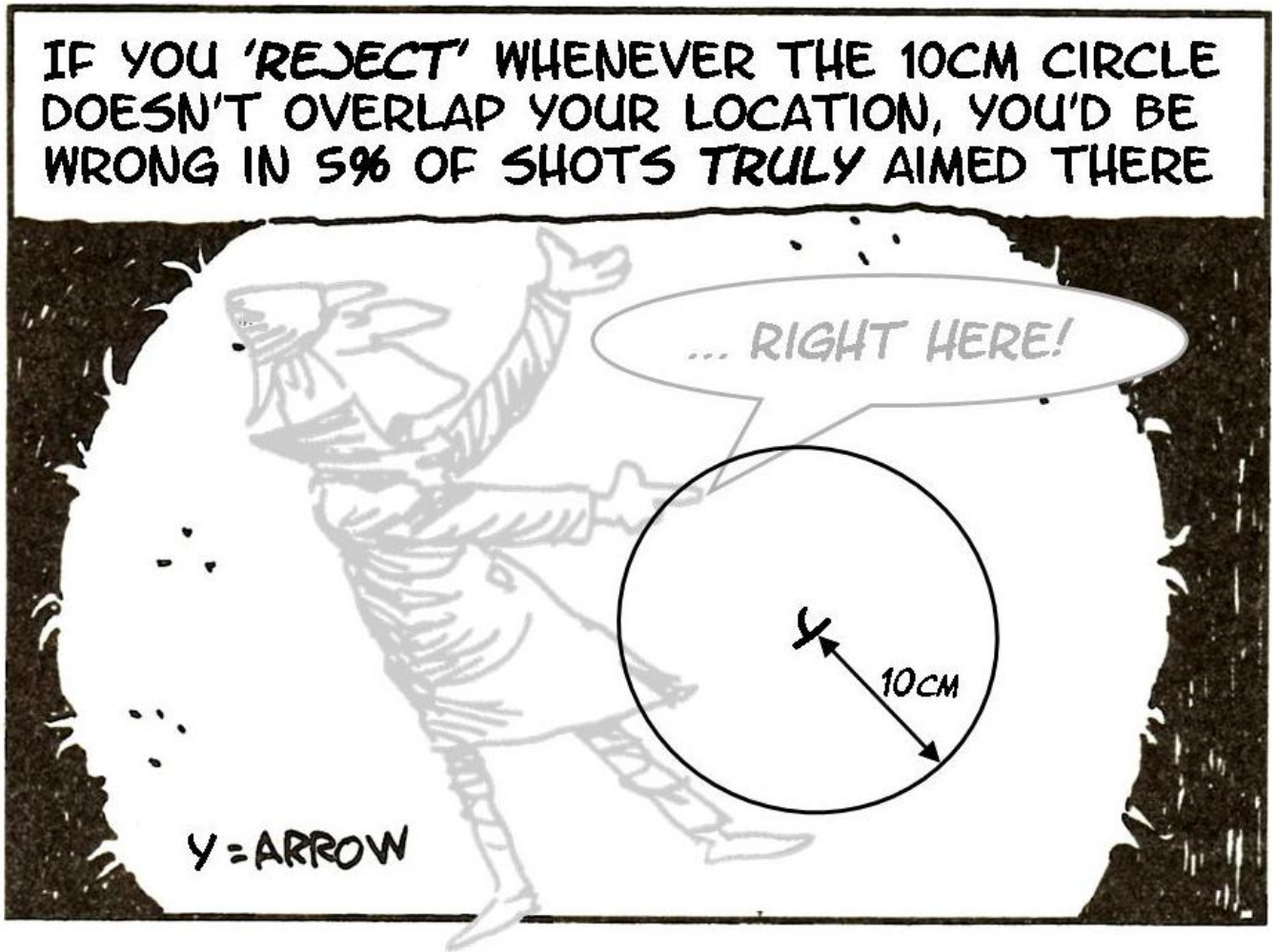
Testing

Before the formalities, an example to make you quiver;



Testing

Before the formalities, an example to make you quiver;



Testing: more formally

You must have something to test – a *null hypothesis*, describing something about the population. For example;

- **Strong Null**: Phenotype Y and genotype G totally unrelated (e.g. mean, variance etc of Y identical at all values of G)
- **Weak Null**: No trend in mean Y across different values of G (e.g. $\beta = 0$ in some regression)

The strong null seems most relevant, for WGS 'searches' – but it can be hard to pin down *why* a null was rejected, making interpretation and replication difficult.

Testing the weak null, it's easier to interpret rejections, and replication is straightforward – but have to be more careful that tests catch trends in the mean Y **and nothing else**.

(One-sided nulls, e.g. $\beta \leq 0$ are not considered here.)

Testing: more formally

Having establish a null hypothesis, choose a **test statistic** for which;

1. We know the distribution, if the null holds
2. We see a different distribution, if the null does not hold

For example, in a regression setting;

1. $Z = \widehat{\beta}_1$/est std error $\sim N(0, 1)$, if $\beta_1 = 0$
2. $Z = \widehat{\beta}_1$/est std error is shifted left/right, on average, if $\beta_1 < 0$, $\beta_1 > 0$ respectively

Particularly for strong null hypotheses, there are **many reasonable choices** of test statistic – more on this later.

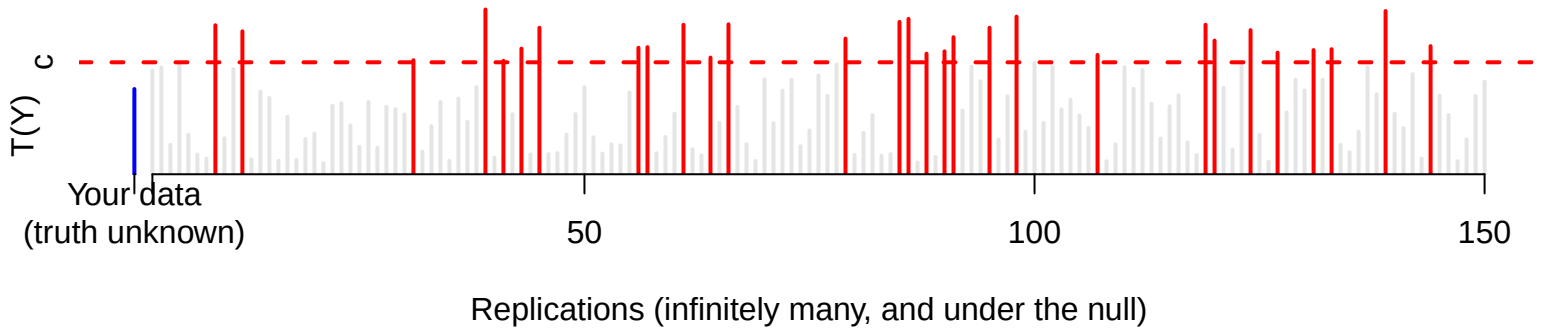
Testing: calibration

1. We know the distribution, if the null holds

This is important! It means;

- We can say how 'extreme' our data is, if the null holds
- If we 'reject' the null hypothesis whenever the data is in (say) the extreme 5% of this distribution, **over many tests** we control the Type I error rate at $\leq 5\%$

For a test statistic T – with the observed data on LHS;



Testing: calibration

P-values are just another way of saying the same thing. A p-value answers the following questions;

- Running the experiment again **under the null**, how often would data look **more extreme** than our data?
- What is the **highest** Type I error rate at which our data would **not** lead to rejection of the null hypothesis?

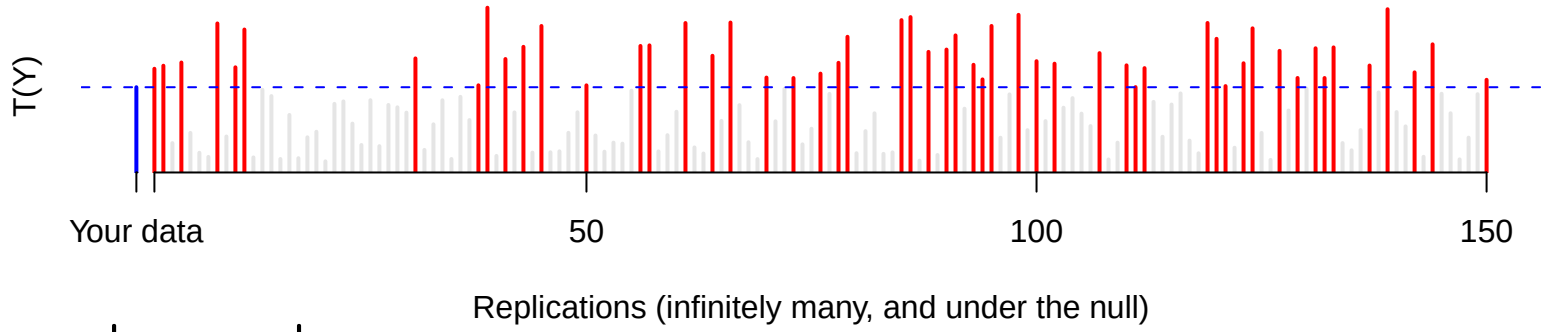
(These are actually equivalent.) All definitions of p-values involve replicate experiments where the null holds.

In WGS, we (or someone) **will do** replicate experiments, in which the null holds*, so this concept – though abstract – is not *too* 'out there'.

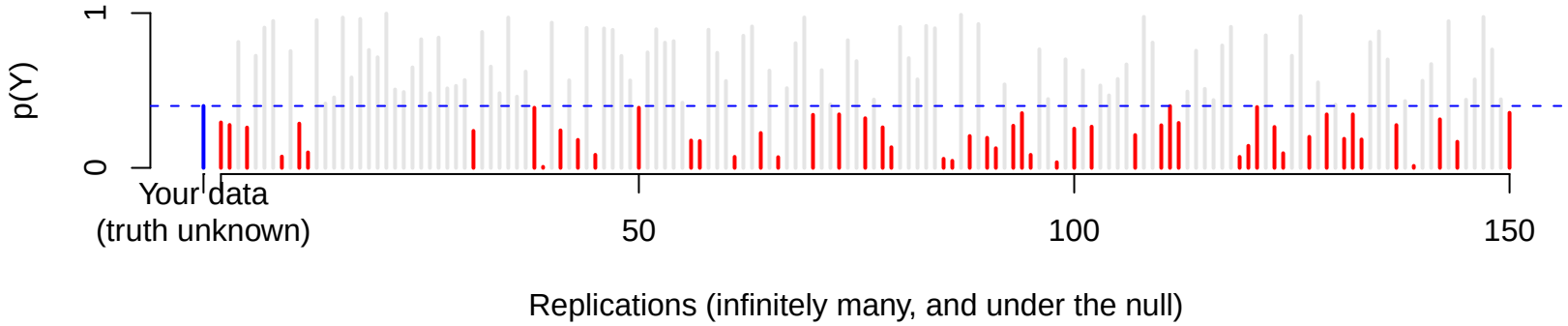
* *...or something* **very** *close to it, though we hope not*

Testing: calibration

Picturing the 'translation' between the test statistic T and p-value;



and p-value scale;



It's usually easiest to implement tests by calculating p, and seeing whether $p < \alpha$. And as *our* choice of α may not match the reviewer's, we should **also** always report p.

Interpreting tests

Everyone's favorite topic;

... a puzzle with missing peas?

Interpreting tests

The version of testing we do in WGS is basically **significance testing**, invented by Fisher.

We interpret results as;

- $p < \alpha$: Worth claiming there is something interesting going on (as we're then only wrong in $100 \times \alpha\%$ of replications)
- $p \geq \alpha$: Make **no conclusion at all**

... for a small value of α, e.g. $\alpha = 0.00000001 = 10^{-8}$

For $p \geq \alpha$, note we are **not** claiming that there is truly zero association, in the population – there could be a modest signal, and we missed it, due to lack of power.

So, don't say 'accept the null' – the conclusion is instead, that it's *just not worth* claiming anything, based on the data we have.

Interpreting tests

Just not worth claiming anything, based on the data we have

- This is a disappointing result – but will be a realistic and common one
- More positively, think of WGS as 'searching' for signals – *needles in haystacks* – and finding a few
- We focus on the signals with highest signal:noise ratios – the *low-hanging fruit*
- We acknowledge there are (plausibly) many more needles/fruit/clichés out there to be found, when more data become available – and we are not 'wrong', because not claiming anything **is not** 'accepting the null hypothesis' and so can't be incorrectly accepting the null, a.k.a. 'making a Type II error'

Beware! This is all **not the same** as a negative result in well-powered clinical trial – Stat 101 may not apply.

Interpreting tests

Slightly more optimistically, say we do get $p < \alpha$. Is our 'hit' a signal? It's actually difficult to tell, based on just the data;

- Most p-values are **not** very informative;
  - There's at least (*roughly*) an order of magnitude 'noise' in small p-values; even for a signal where we expect $p \approx 10^{-6}$, expect to see anything from 10^{-5}–10^{-7}; a broad range
  - Confounding by ancestry possible (i.e. we found a signal but not an interesting one)
  - If you **do** get $p = 10^{-\text{ludicrous}}$, there may be issues with the validity of your reported p-values
- Does the signal look like ones we had power to detect? Good question, but hard to answer because of Winners' Curse
- So most journals/other scientists rely on replication – quite sensibly – and are **not** interested in the precise value of $p < \alpha$

Interpreting tests

What does my WGS have power to detect?

- Do power calculations. These found that, for plausible* situations, doing efficient tests, we might have e.g. 20% power
- This means;
  - 1 in 5 chance of finding that signal, rejecting its null
  - **Have to be lucky** to find that particular true signal
  - P-values for this signal look quite similar to those from noise – a *little* smaller, on average
- However, if there are e.g. 50 variants for which we have 20% power, we expect to find $50 \times 0.2 = 10$ signals, and we would be **very unlucky** not to find any.

So WGS is worth doing – but won't provide all the true signals, or even most of them. (For single variant tests, Quanto is recommended for quick power calculations)

*... we don't know the truth; these are best guesses

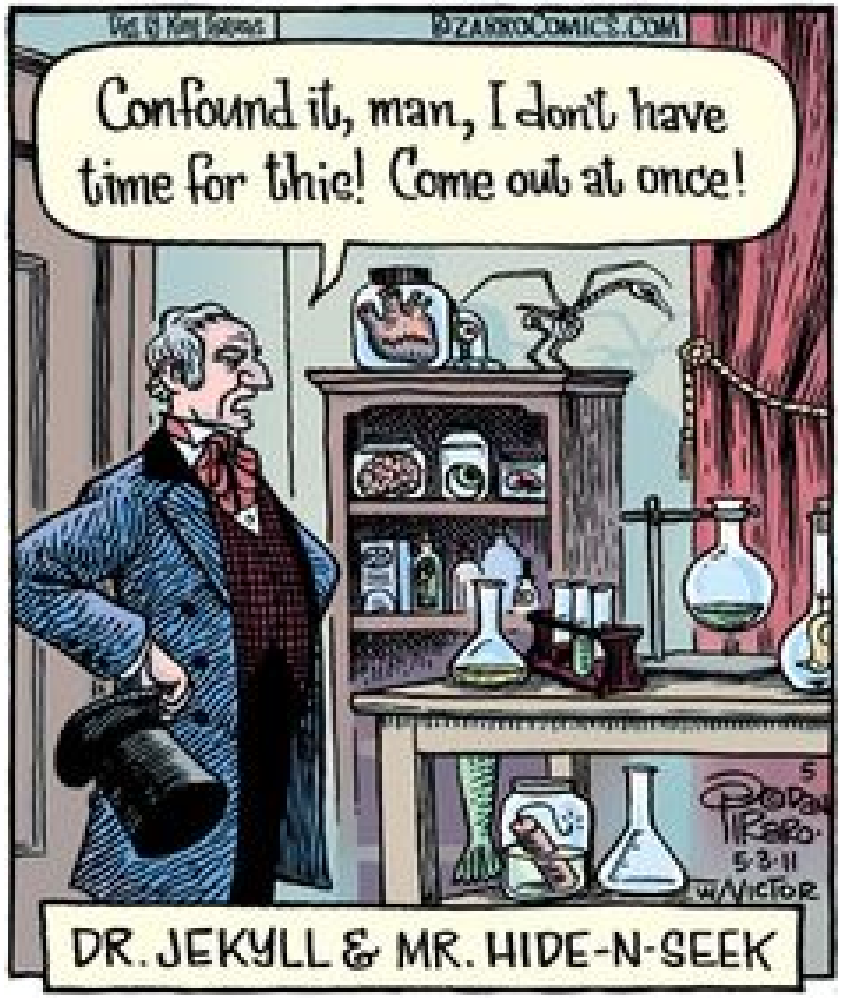
Interpreting tests

Most problems with p-values arise when they are *over*-interpreted

- Getting $p < \alpha$ is not 'proof' of anything. $p < \alpha$ is entirely possible, even if no signals exist
- p-values measure signal:noise, badly;
  - $\widehat{\beta}$ and its std error measure signal and noise *much better*
  - Smaller p-values in one study do **not** indicate that study is 'better' – even if the difference is not noise, that study may just be larger
  - Smaller p-values in one *analysis* have to be checked carefully – are they calibrated accurately, under the null? Is the difference more than noise?
- p-values are not probabilities that the null hypothesis is true. This is "simply a misconception" [David Cox]

Confounding

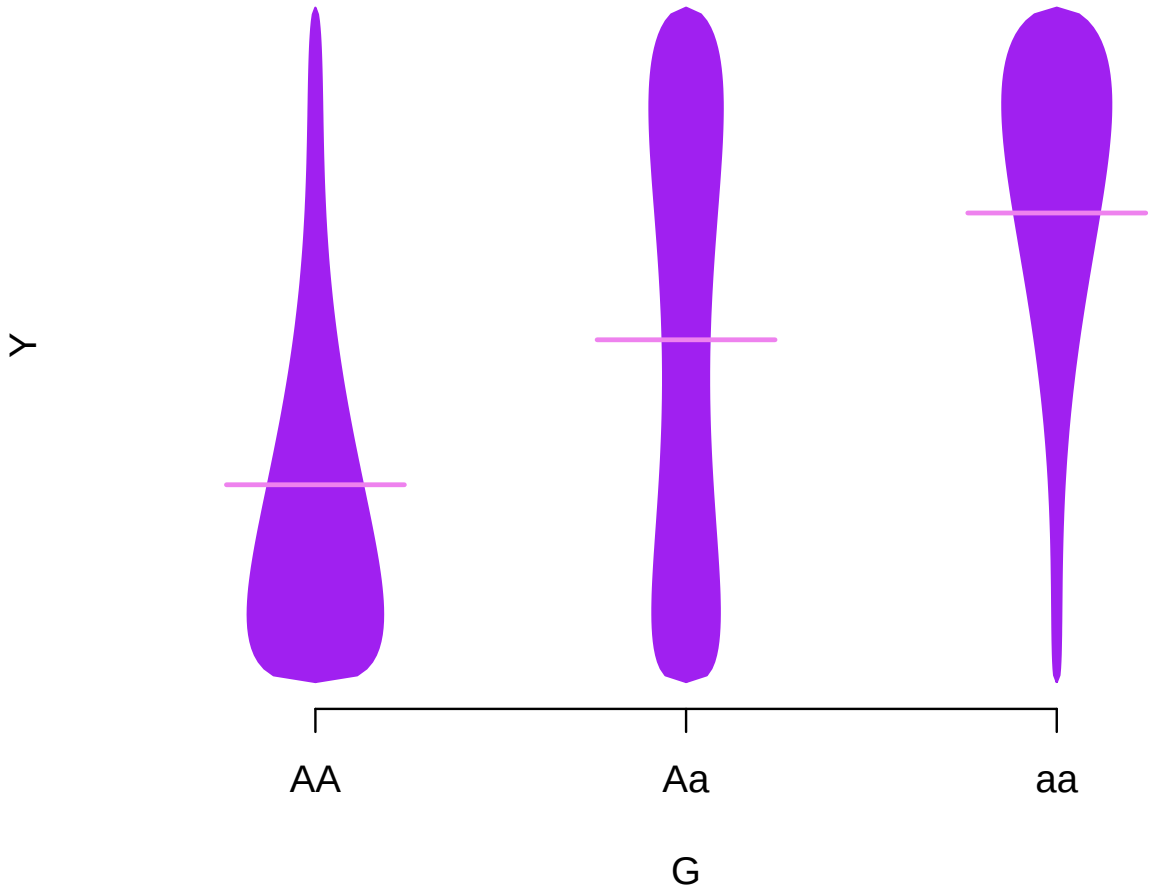
Confounding: the 'pouring together' of two signals.



In WGS: finding a signal, but **not** one that leads to novel understanding of the phenotype's biology.

Confounding

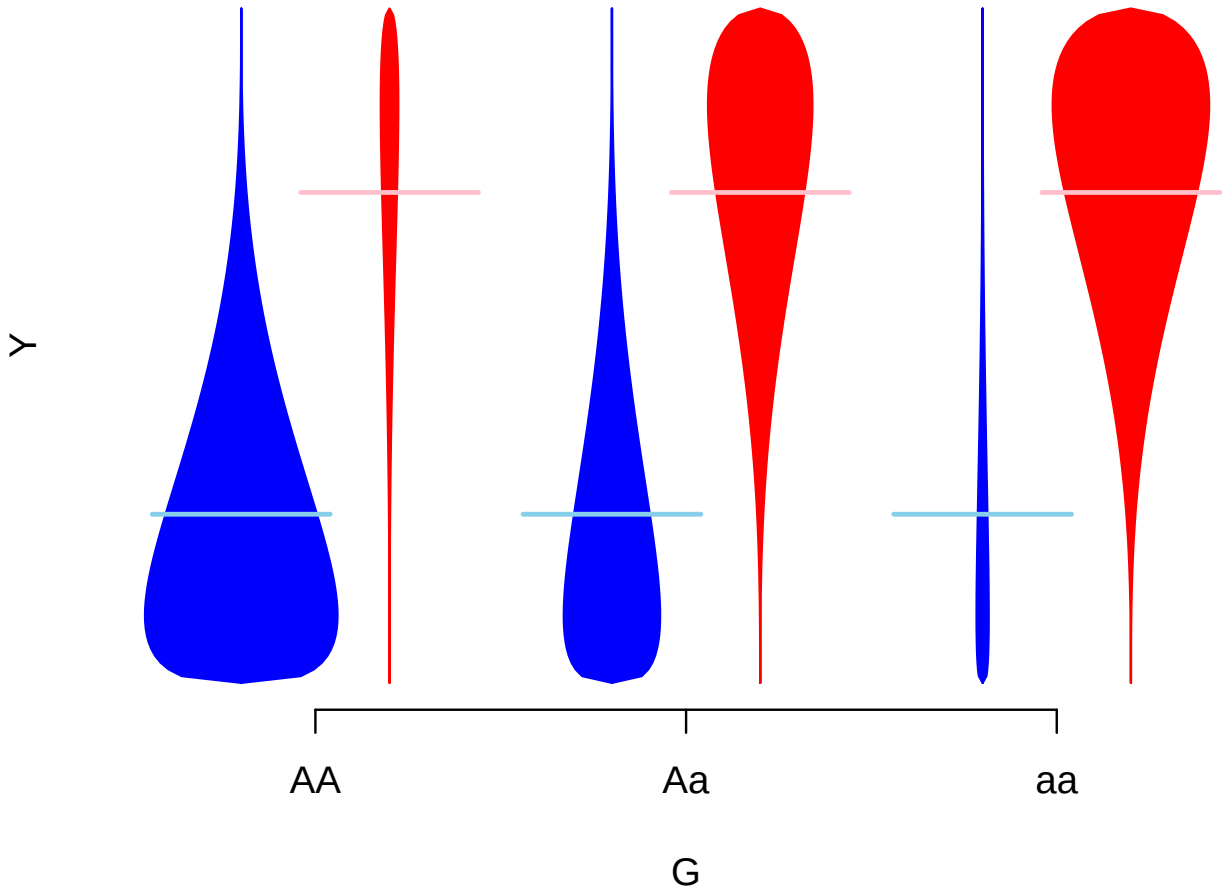
A population where there is a real association between phenotype and genotype;



Linear regression (or other regressions) should find this signal.

Confounding

But the same population looks much less interesting *genetically*, broken down into ancestry groups;



This effect is *population stratification* a form of *confounding*.

Confounding

Of course, this association is not interesting – within each group, there is no association.

- It's still an association! Confounding – getting the wrong answer to a causal question – does not mean there's 'nothing going on'

Suppose, among people of the same ancestry, we see how mean phenotype levels differ by genotype;

- Imagine ancestry-specific regressions (there may be a lot of these)
- Average the β_1 from each ancestry group
- If we test whether this **average** is zero, we'll get valid tests, *even if there is stratification*

Confounding

In practice, we don't have a fixed number of ancestry groups. Instead we fit models as follows;

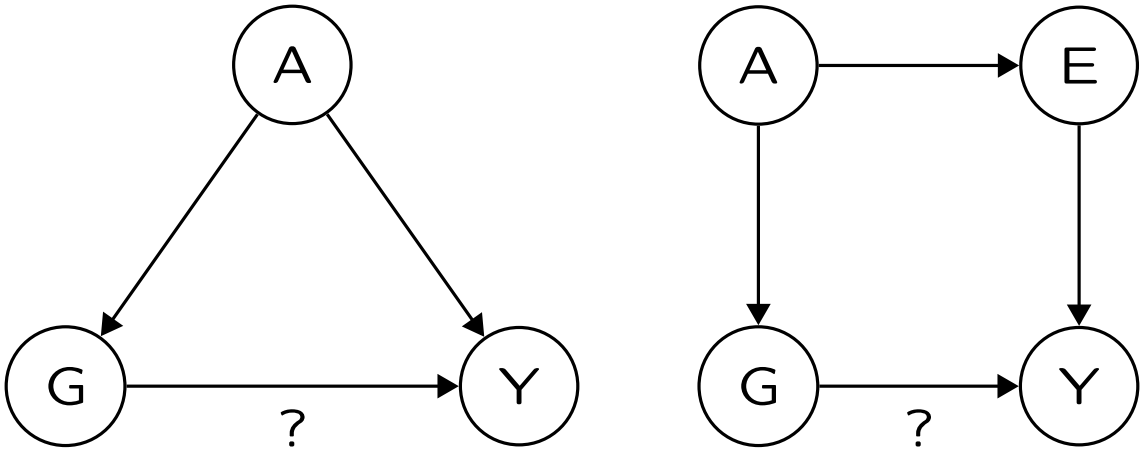
$$\text{Mean}(Y) = \beta_0 + \beta_1 G + \gamma_1 PC_1 + \gamma_2 PC_2 + \gamma_3 PC_3 + ...$$

... i.e. do regression adjusting for PC_1, PC_2, PC_3 etc. (Logistic, Cox regression can be adjusted similarly)

- Among people with the same ancestry (i.e. the same PCs) β_1 tells us the difference in mean phenotype, per 1-unit difference in G
- If the effect of G differs by PCs, β_1 tells us a (sensible) average of these genetic effects
- If the PCs don't capture all the confounding effects of ancestry on Y – for example through non-linear effects – then this approach won't work perfectly. But it's usually good enough

Confounding

Why not adjust for other variables? Ancestry (and/or study) affects genotype, and often phenotype – through e.g. some environmental factor;

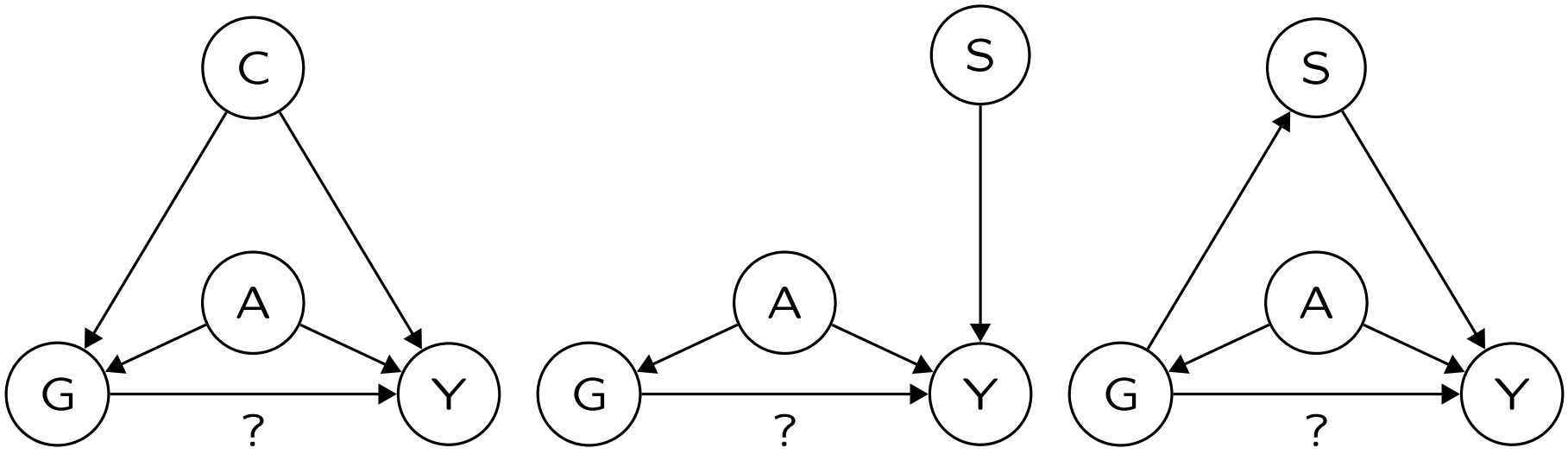


If we could adjust (adequately) for E, the confounding would be prevented. But genotype is *very* well-measured, compared to almost all environmental variables* – so adjusting for ancestry (and study) is a better approach.

* the relevant 'E' may not even be known, let alone measured

Confounding

What about other confounders? (Er... there aren't any)

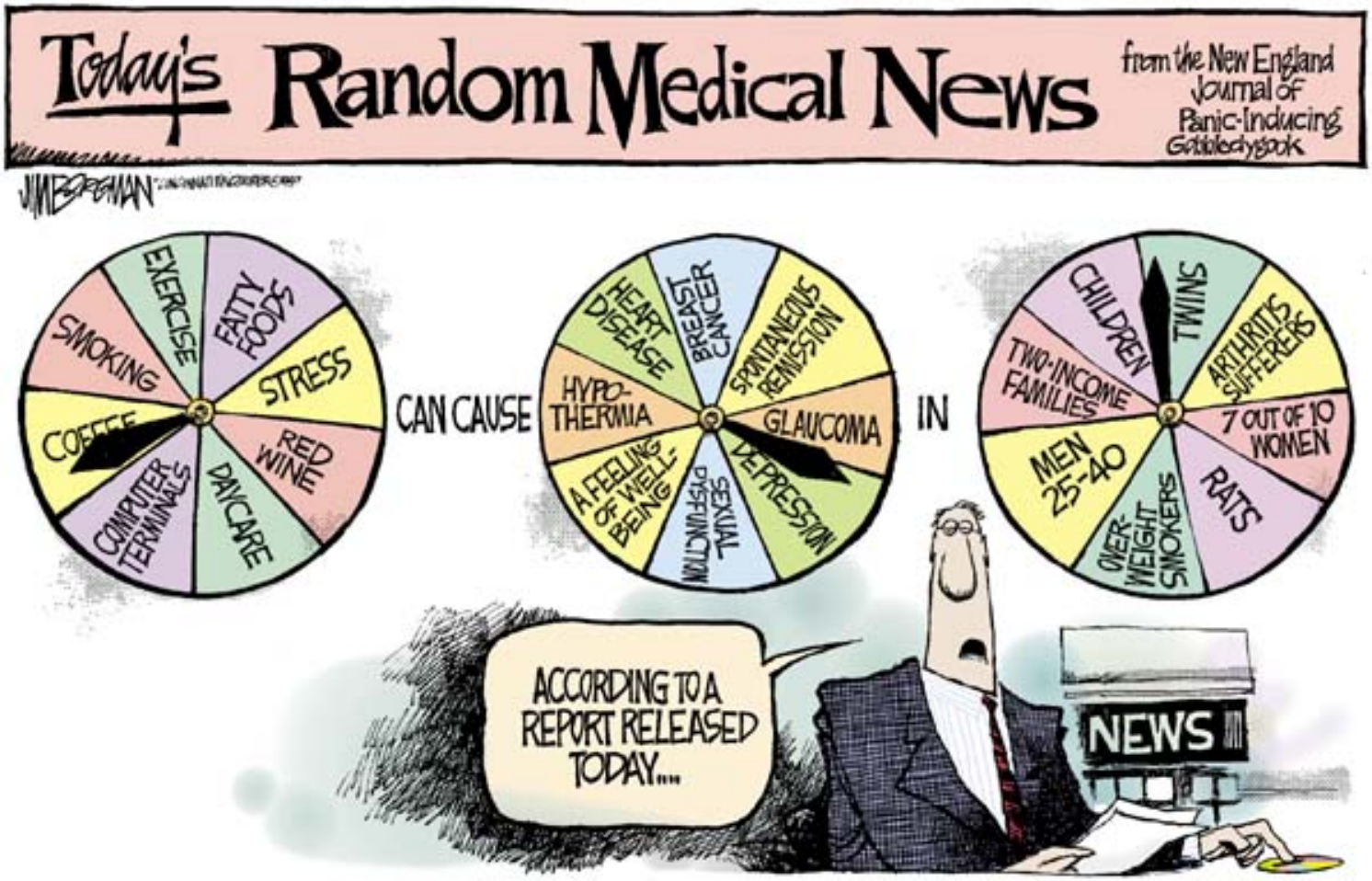


Any other confounder C affects G and Y – and is not ancestry.

Adjusting for sex (or smoking, or age) can aid precision – though often not much – but also removes any real genetic signals that act through that pathway, i.e. reduces power. **Plan carefully!**

Multiple tests: small α

A classical view;



Multiple tests: small α

Why does α have to be so small?

- We want to have *very few* results that don't replicate
- Suppose, with entirely null data, we test $10M$ variants at $\alpha = 1/10M$; we expect $= 1/10M \times 10M = 1$ false positives, by chance alone, *regardless of any correlation*
- If we are well-powered to detect only 10 true signals, that's a false positive rate around $1/11 \approx 9\%$. If there was just one such true signal, the false positive rate would be about 50%, i.e. about half the results would not be expected to replicate

Thresholds *have* to be roughly 1/n.tests; this threshold is only conservative because *our haystack contains a lot of non-needles we search through*.

(See also upcoming empirical work by Chen, Mukherjee & Boehnke, arguing that 5×10^{-8} is about right for lipids.)

Multiple tests: small α

But 'multiple testing is conservative'

- No it's not, here
- Bonferroni correction *is* conservative, if;
  - We care about the probability of making any Type I errors ('Family-Wise Error Rate')
  - High proportion of test statistics are closely correlated
  - We have modest α, i.e. $\alpha = 0.05$
- If we have **all three** of these, Bonferroni corrects the actual FWER well below the nominal level. But we don't have all three, so it's not.

(A corollary from this: filters to reduce the multiple-testing burden must be **severe** and **accurate** if they are to appreciably improve power – see later material on annotation.)

Multiple tests: Winners' curse

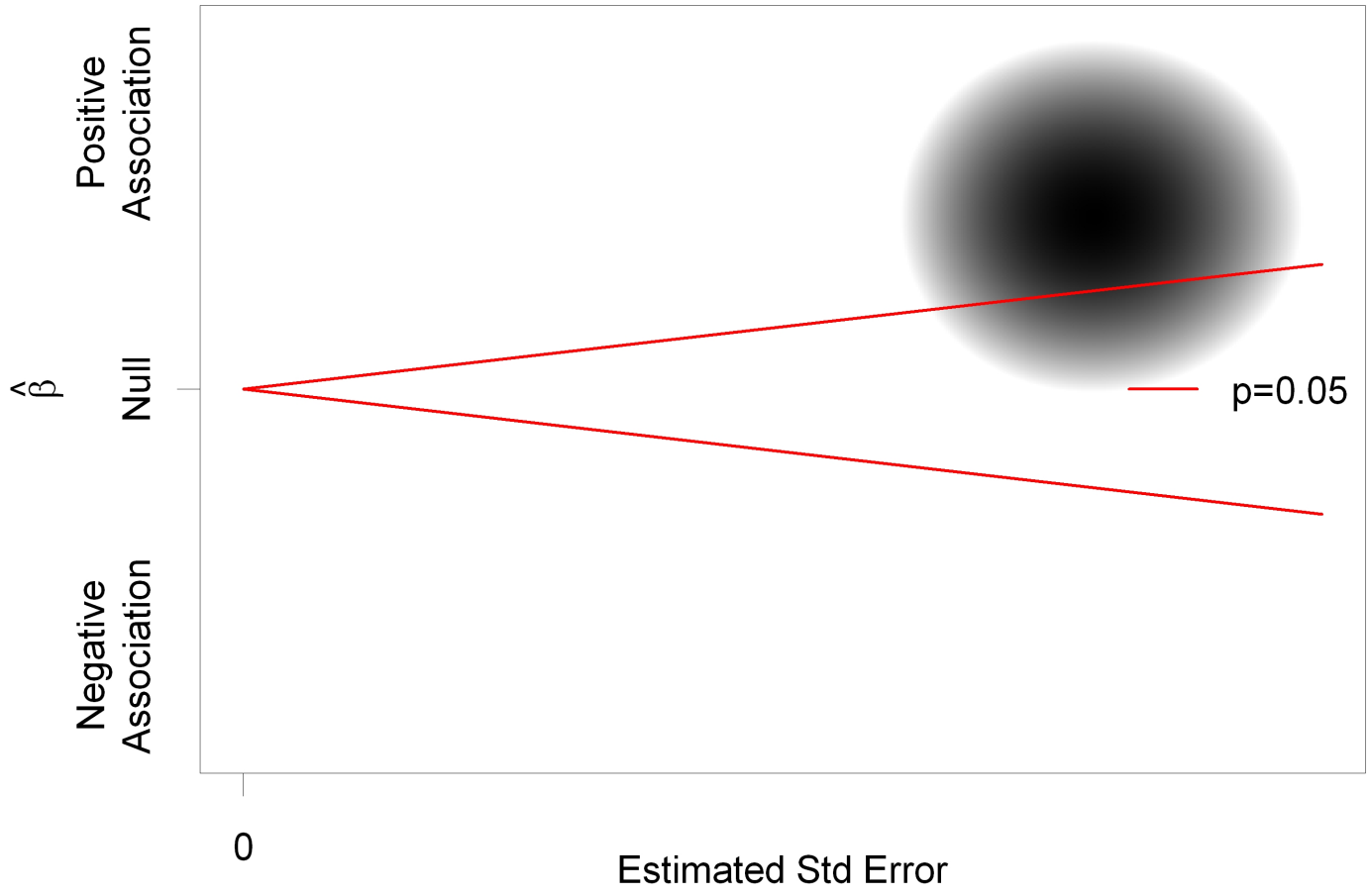
A consequence of using small α is that **Winner's Curse*** – effect estimates in 'hits' being biased away from zero – is a bigger problem than we're used to in standard epidemiology

- Why is it such a big problem?
- Roughly how big a problem is it?

**...also known as regression towards the mean, regression toward mediocrity, the sophomore slump, etc etc*

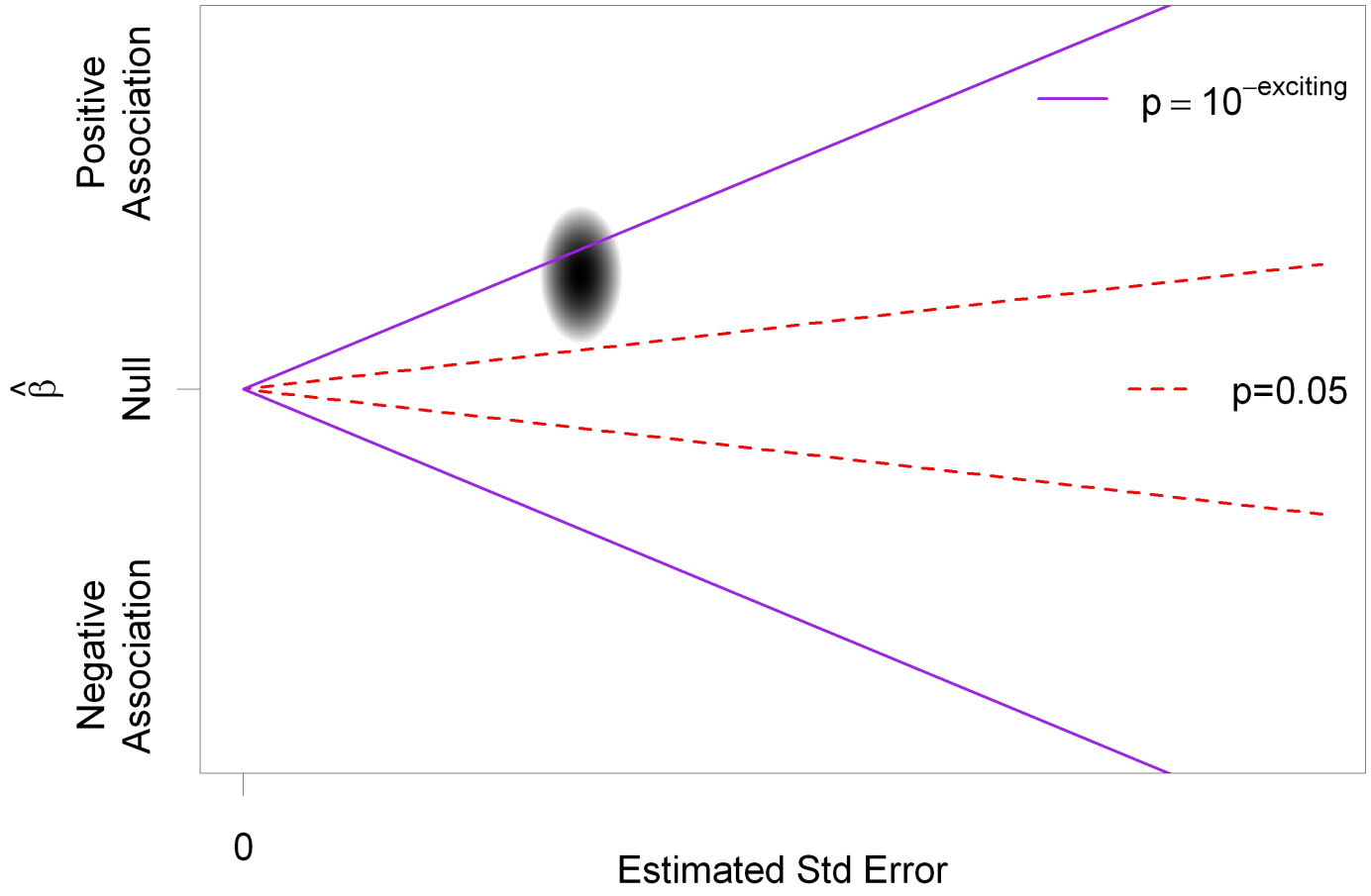
Multiple tests: Winners' curse

We reject the null when observed signal:noise is high; this isn't 100% guaranteed, even in classic Epi;



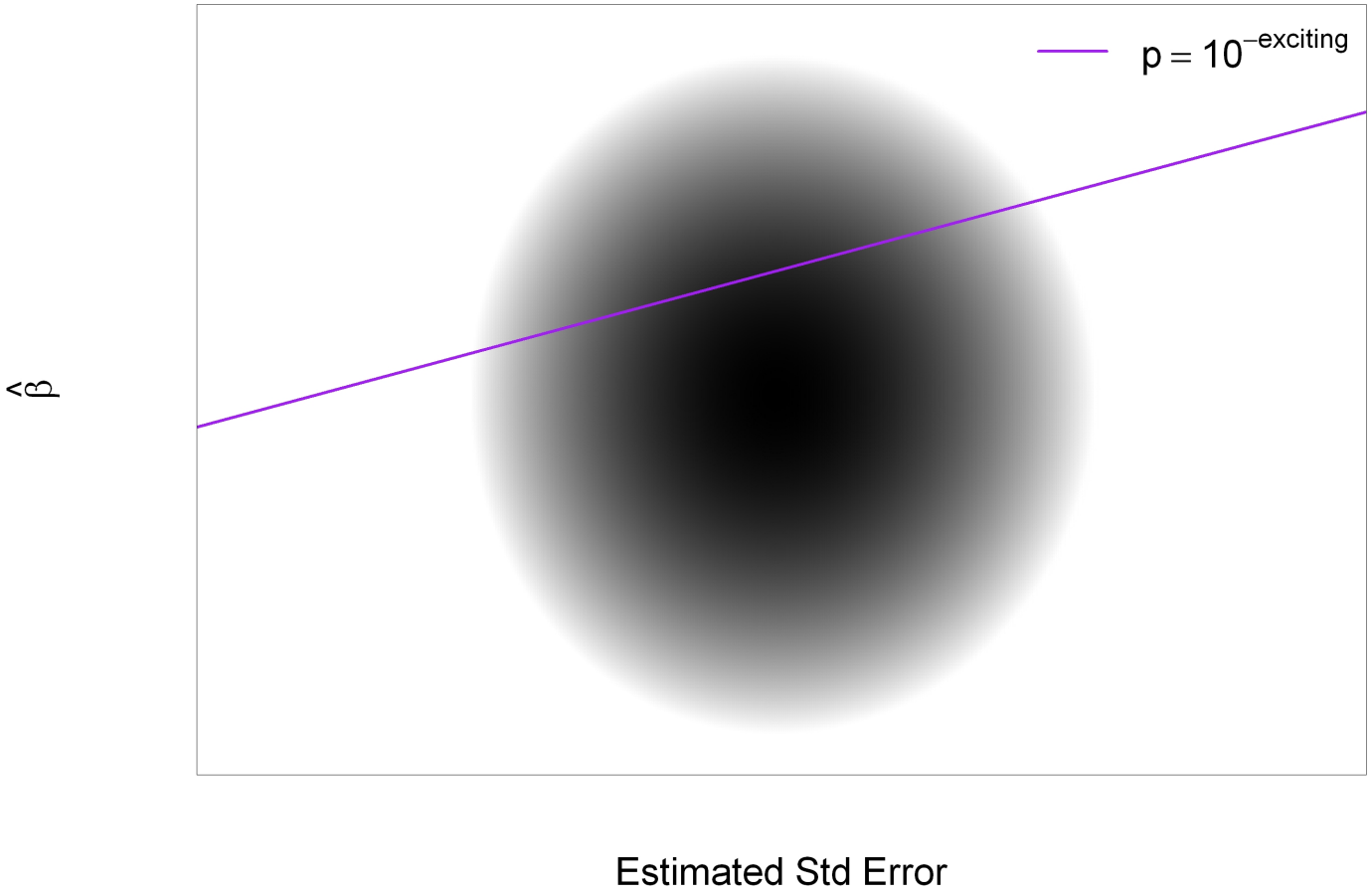
Multiple tests: Winners' curse

In rare-variant work, much bigger datasets and much smaller signals;



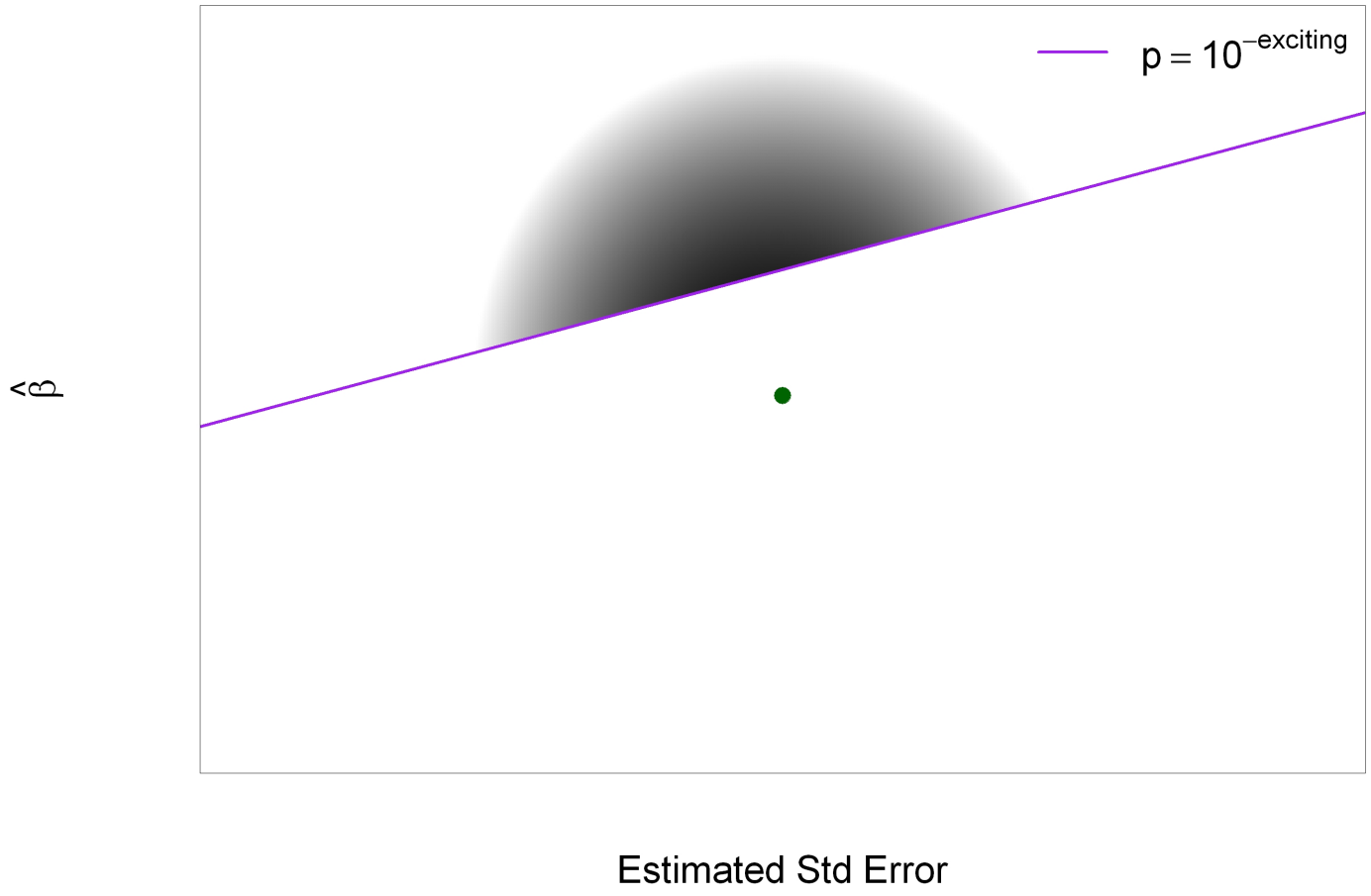
Multiple tests: Winners' curse

In rare-variant work, much bigger datasets and much smaller signals; (zoom)



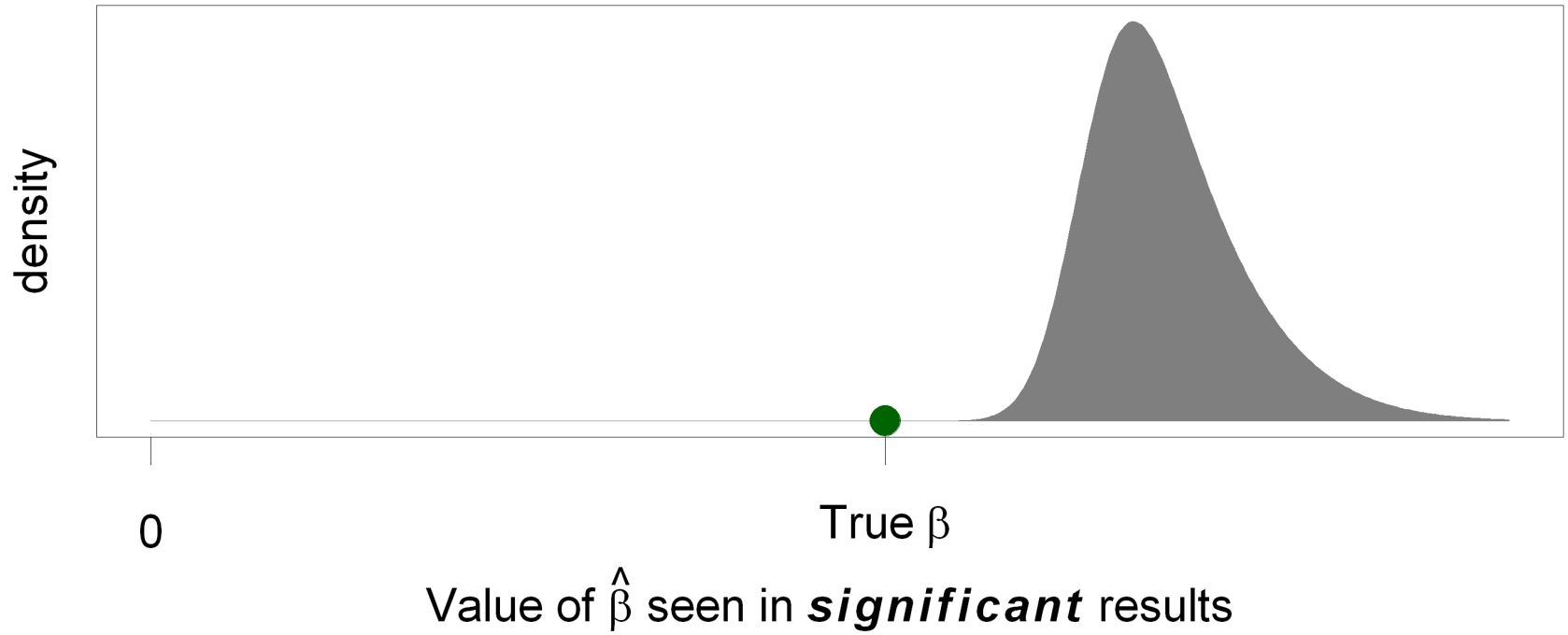
Multiple tests: Winners' curse

Now restrict to everything we called significant – which is all we focus on, in practice;



Multiple tests: Winners' curse

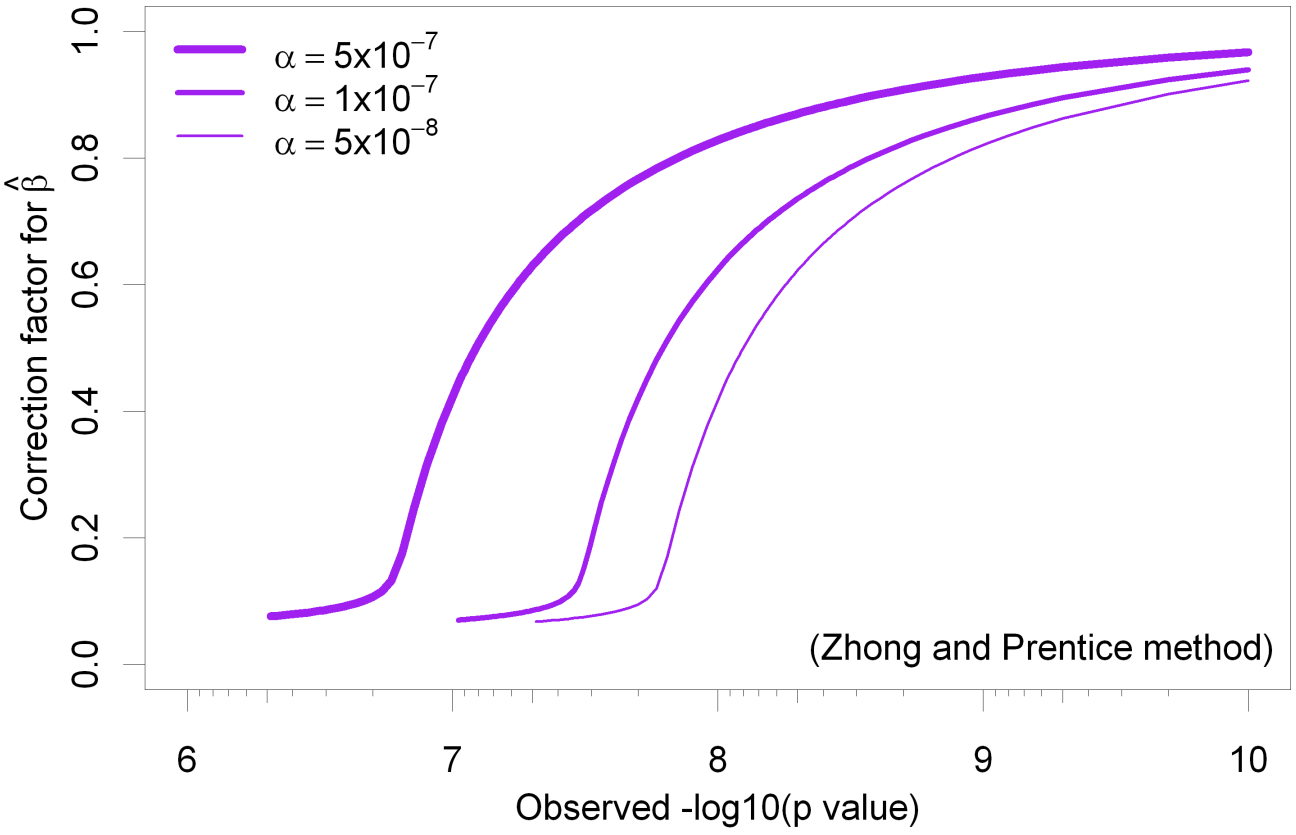
The 'significant' results are the ones **most likely** to have exaggerated estimates – a.k.a. bias away from the null;



Objects in the rear-view mirror are *vastly* less important than they appear to be. The issue is worst with lowest power (see previous pics) – so $\widehat{\beta}$ from TOPMed/GWAS 'discovery' work are not *and cannot be* very informative.

Multiple tests: Winners' curse

How overstated? Zhong and Prentice's p-value-based estimate;



This is a serious concern, unless you beat α by about 2 orders of magnitude.

Multiple tests: Winners' curse

Winner's curse doesn't just affect point estimates;

- If we claim significance based on unadjusted analyses, adjusted analyses may give reduced estimates *even when the other covariates do nothing* – because the second analysis is less biased away from the null
- In analysis of many studies, significant results will tend to appear homogeneous *unless* we are careful to use measures of heterogeneity are independent of the original analysis

Part 1: Summary

- P-values assess deviations from a null hypothesis
- ... but in a convoluted way, that's often over-interpreted
- Associations may be real but due to confounding; need to replicate and/or treat results with skepticism
- Power matters and will be low for many TOPMed signals (though not all)
- At least for discovery work, point estimates $\widehat{\beta}$ are often not very informative. When getting $p < \alpha$ 'exhausts' your data, it can't tell you any more.

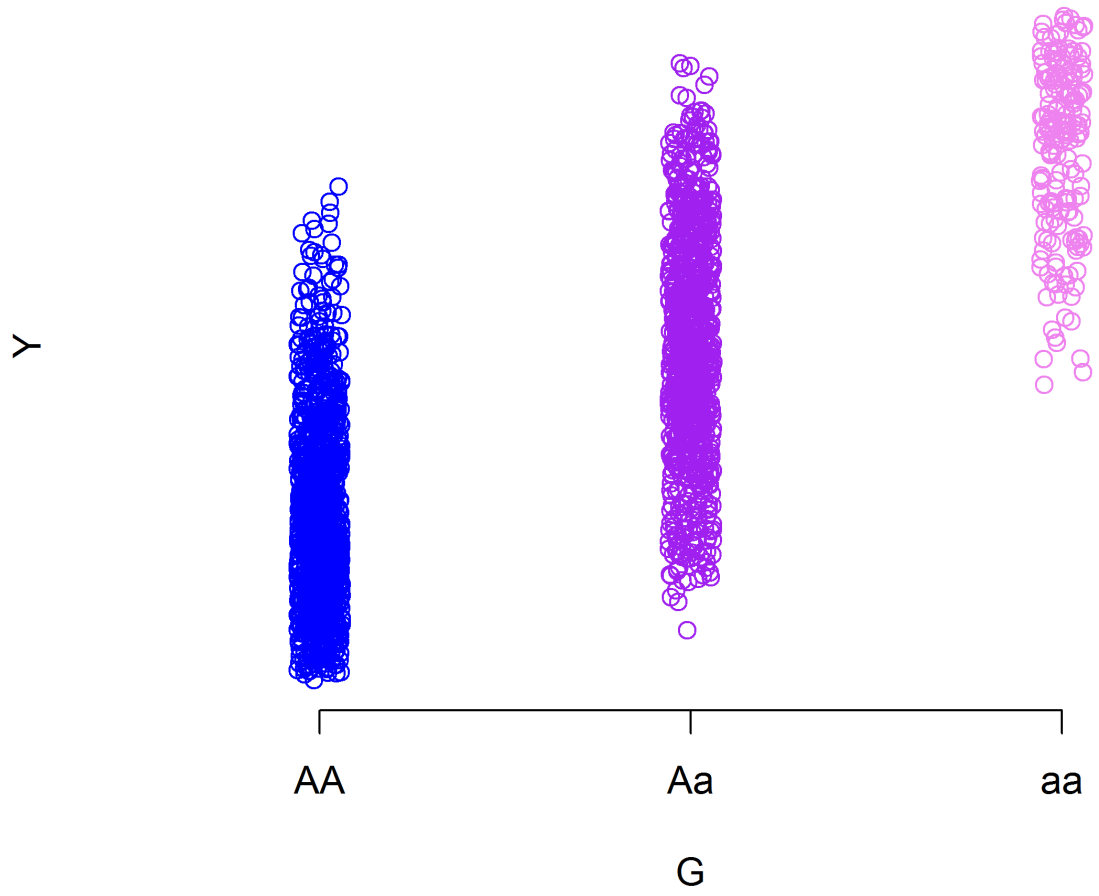
Part 2: Single Variant Assocation Tests

Overview

- How single variant tests work
- Adjusting for covariates; how and why
- Allowing for relatedness
- Trait transformation
- Issues of mis-specification and small-sample artefacts
- Binary outcomes (briefly)

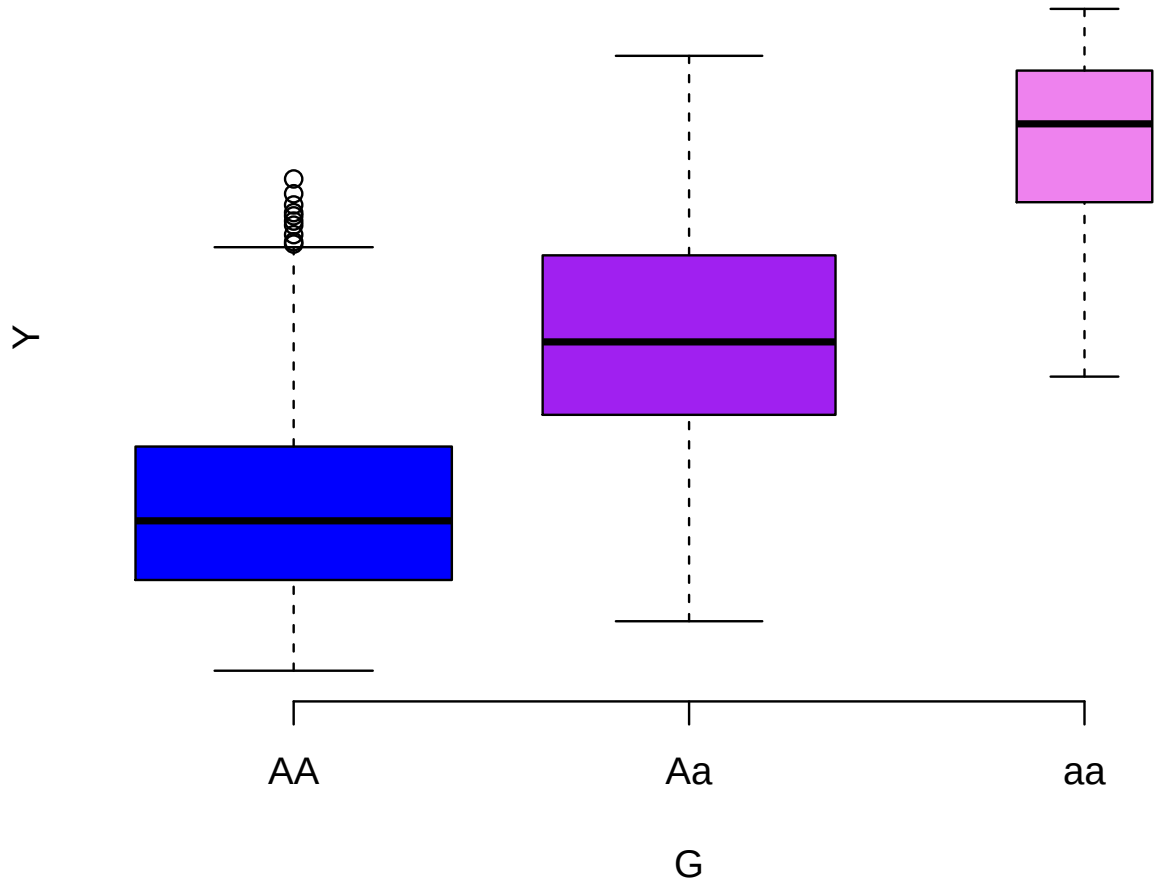
How single variant tests work

Regression tells us about differences. How do the phenotypes (Y) of these people differ, according to their genotype (G)?



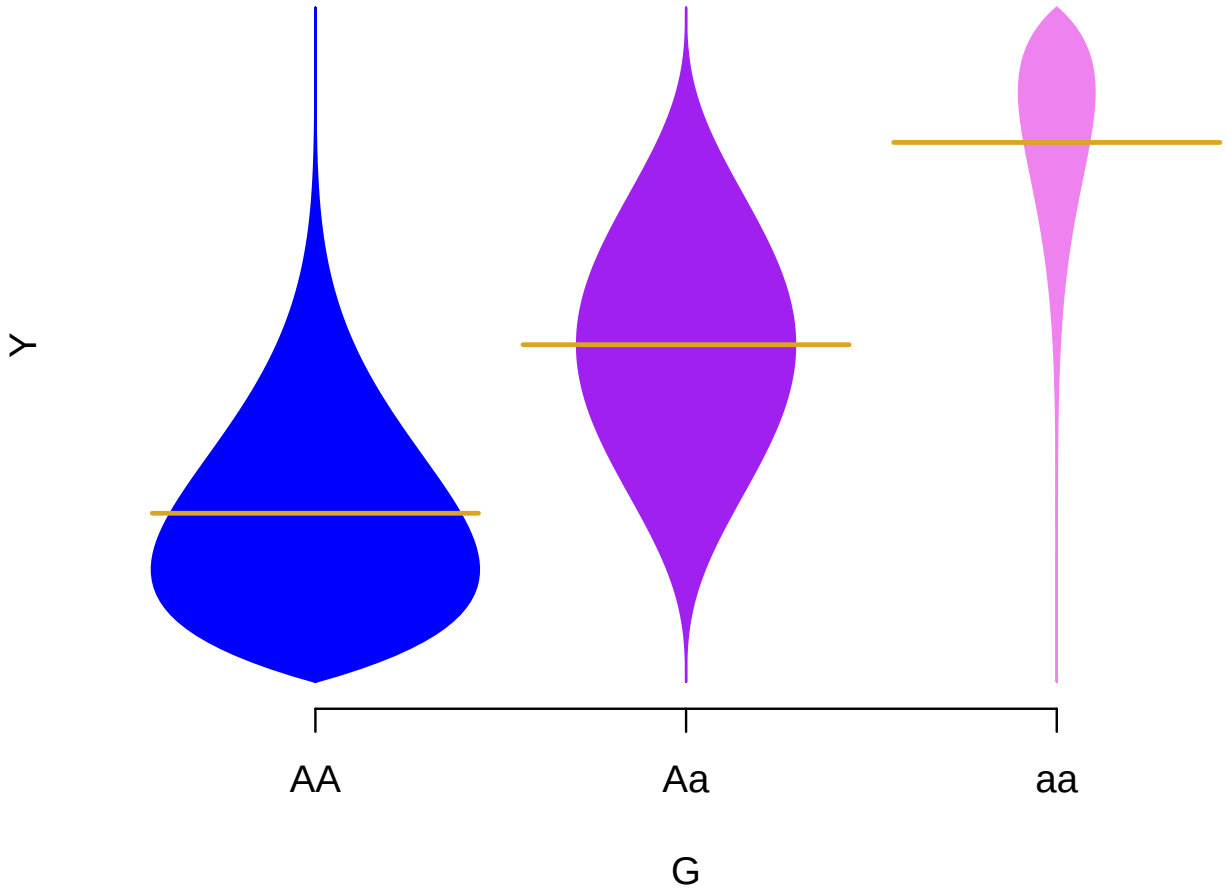
How single variant tests work

Here's the same sample of data, displayed as a boxplot – how do the medians differ?



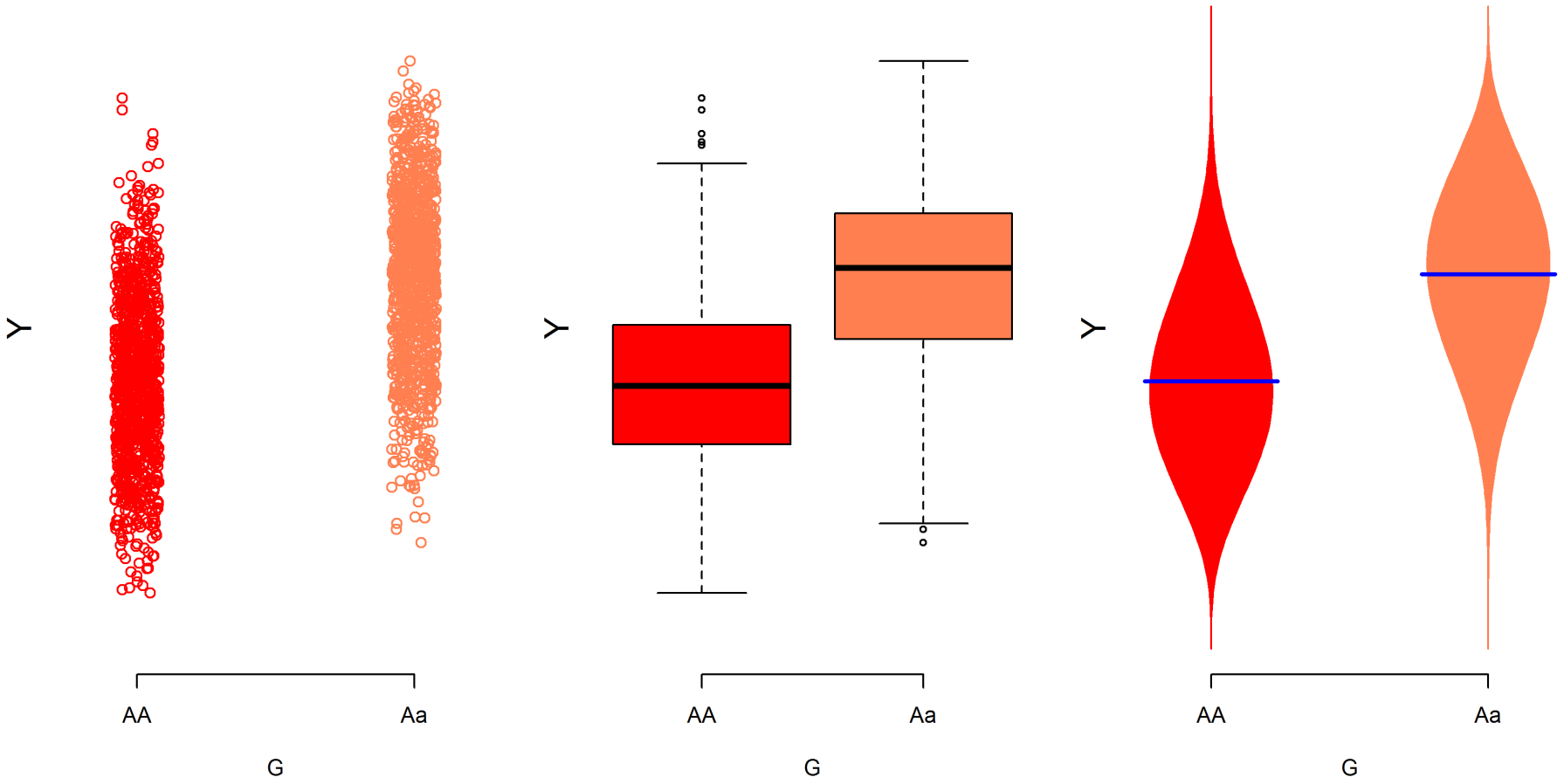
How single variant tests work

Formally, the *parameter* being estimated is 'what we'd get with infinite data' – for example, differences in means;



How single variant tests work

Particularly with binary genotypes (common with rare variants) using the difference in mean phenotype as a parameter is natural;



How single variant tests work

To summarize the relationship between quantitative phenotype Y and genotype G, we use parameter β_1, the slope of the *least squares line*, that minimizes

$$(Y - \beta_0 - \beta_1 G)^2,$$

averaged over the infinite 'population'.

- β_1 is an 'average slope', describing the **whole population**
- ... β_1 describes how much higher the **average** phenotype is, per 1-unit difference in G
- For binary G, β_1 is just the difference in means, comparing the parts of the population with $G = 1$ to $G = 0$

This 'averaging' is important; on its own, β_1 says **nothing** about **particular, individual** values of Y or G.

But if Y and G are totally unrelated, we know $\beta_1 = 0$, i.e. the line is flat – and this connects the methods to testing.

Inference for single variant tests

Of course, we never see the whole population – n is finite.

But it *is* reasonable to think that our data are one random sample (size n) from the population of interest.

We have no reason to think our size-n sample is 'special', so;

- Computing the population parameter β_1 using just our data $Y_1, Y_2, ...Y_n$ and $G_1, G_2, ...G_n$ provides a sensible **estimate** of β_1 – which we will call $\widehat{\beta}_1$. This *empirical* estimate is (of course) not the truth – it averages over our observed data, not the whole population.
- As we'll see a little later, this *empirical* approach can also tell us how much $\widehat{\beta}_1$ varies due to chance in samples of size n – and from this we obtain tests, i.e. p-values

Inference for single variant tests

For estimation, the empirical version of β_0 and β_1 are estimates $\widehat{\beta}_0$ and $\widehat{\beta}_1$, that minimize

$$\sum_{i=1}^{n} (Y_i - \beta_0 - \beta_1 G_i)^2.$$

With no covariates or relatedness, this means solving

$$\sum_{i=1}^{n} Y_i = \sum_{i=1}^{n} \beta_0 + \beta_1 G_i \qquad \sum_{i=1}^{n} Y_i G_i = \sum_{i=1}^{n} G_i(\beta_0 + \beta_1 G_i),$$

and for β_1 – the term of interest – gives

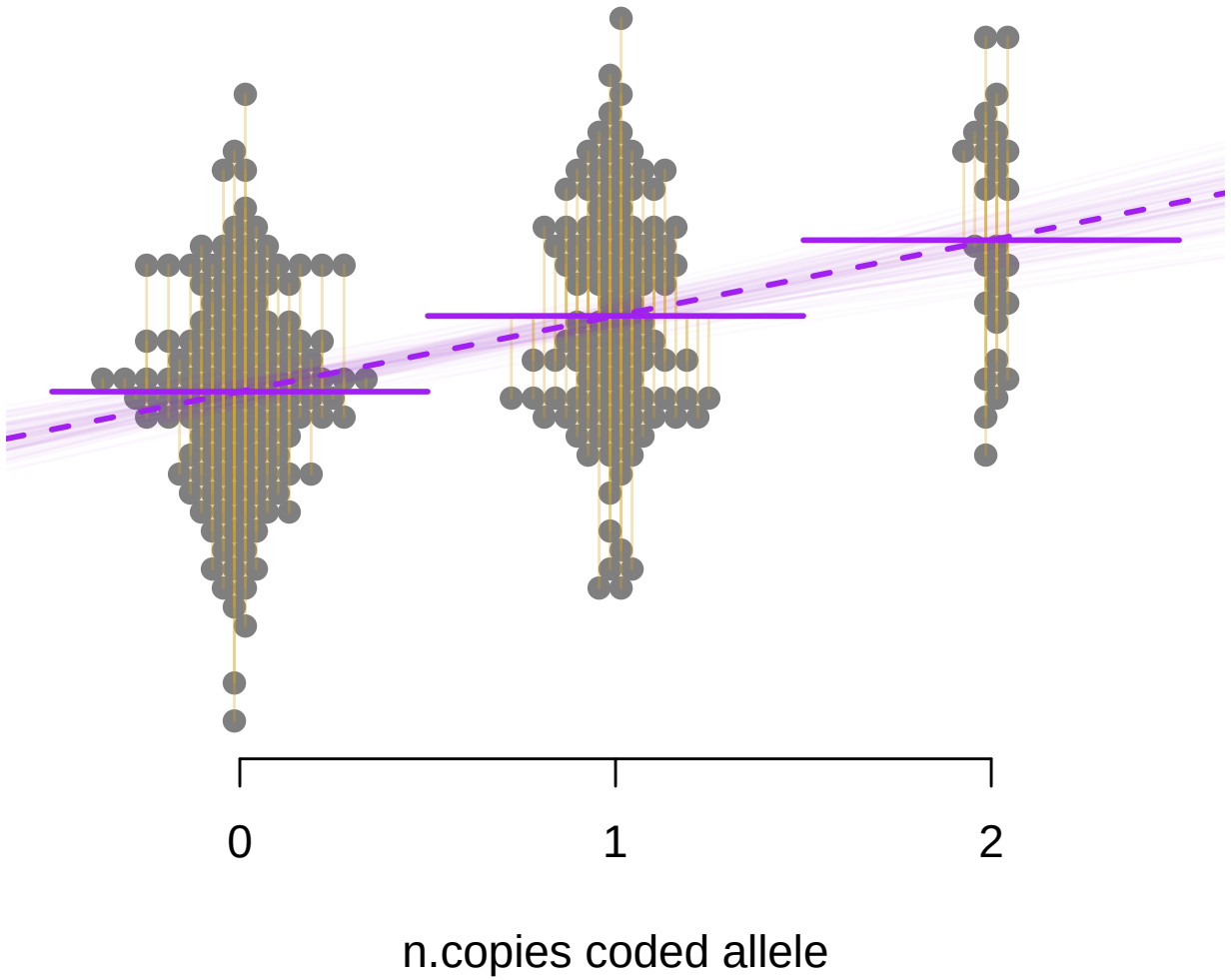
$$\widehat{\beta}_1 = \frac{\text{Cov}(\mathbf{Y}, \boldsymbol{G})}{\text{Var}(\boldsymbol{G})} = \frac{\sum_{i=1}^{n}(Y_i - \bar{Y})(G_i - \bar{G})}{\sum_{i=1}^{n}(G_i - \bar{G})^2} = \frac{\sum_{i=1}^{n} \frac{Y_i - \bar{Y}}{G_i - \bar{G}}(G_i - \bar{G})^2}{\sum_{i=1}^{n}(G_i - \bar{G})^2}.$$

- $\widehat{\beta}_1$ is a **weighted average** of ratios; how extreme each outcome is relative to how extreme its genotype is
- Weights are greater for extreme genotypes – i.e. heterozygotes, for rare alleles

Inference for single variant tests

But how noisy is estimate $\widehat{\beta}_1$?

To approximate $\widehat{\beta}_1$'s behavior in replicate studies, we could use each point's *actual* residual – the vertical lines – as the empirical std deviation of their outcomes;



In replicate studies, slope estimate $\widehat{\beta}_1$ would vary (pale purple lines) but contributions from each Y_i are not all equally noisy.

Inference for single variant tests

In unrelateds, this 'robust' approach gives three main terms;

$$\widehat{\text{StdErr}}(\hat{\beta}_1) = \frac{1}{\sqrt{n-2}} \frac{1}{\text{StdDev}(\boldsymbol{G})} \text{StdDev}\left(\frac{\boldsymbol{G} - \bar{\boldsymbol{G}}}{\text{StdDev}(\boldsymbol{G})} (\mathbf{Y} - \hat{\beta}_0 - \hat{\beta}_1 \boldsymbol{G}) \right).$$

... sample size, spread of G, and (weighted) spread of residuals.

Because of the Central Limit Theorem*, the distribution of $\hat{\beta}_1$ in replicate studies is approximately Normal, with

$$\hat{\beta}_1 \overset{approx}{\sim} N\left(\beta_1, \widehat{\text{StdErr}}(\hat{\beta}_1)^2 \right).$$

The corresponding *Wald test* of the null that $\beta_1 = 0$ compares

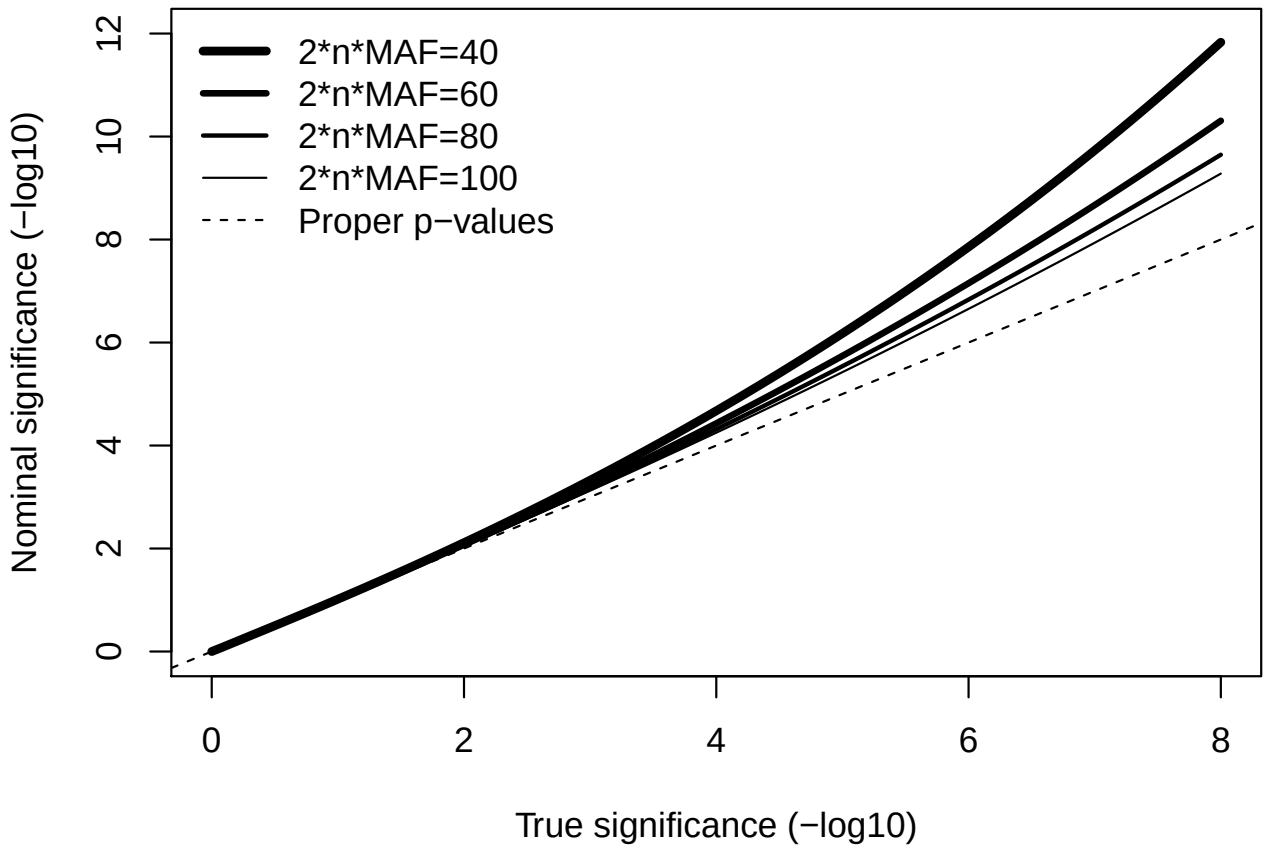
$$Z = \frac{\hat{\beta}_1}{\text{EstStdErr}(\hat{\beta}_1)} \text{ to } N(0,1).$$

The tail areas beyond $\pm Z$ give the two-sided p-value. Equivalently, compare Z^2 to χ^2_1, where the tail above Z^2 i s the p-value.

* ... the CLT tells us approximate Normality holds for essentially any sum or average of random variables.

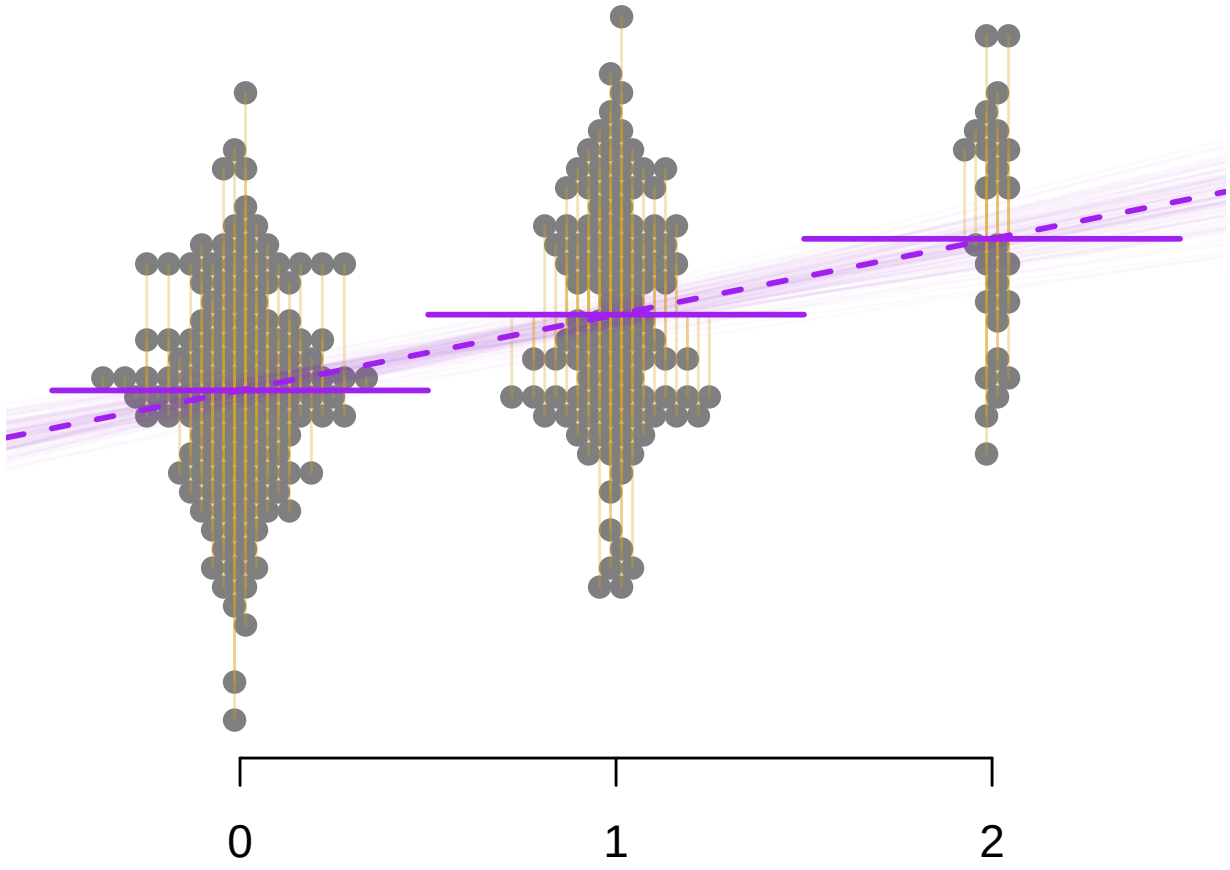
Inference for single variant tests

But the CLT's approximate methods, for these 'robust' standard errors don't work well with small n – when minor allele count is low and/or tiny p-values are needed;



- Some smarter approximations are available, but hard to make them work with family data
- Instead, we use a **simplifying assumption**; we assume residual variance is identical for all genotypes

Inference for single variant tests



We assume the residual variance is the same for all observations, regardless of genotype (or covariates)

Inference for single variant tests

Using the stronger assumptions (in particular, the constant variance assumption) we can motivate different estimates of standard error;

$$\widehat{\mathsf{StdErr}}(\hat{\beta}_1) = \frac{1}{\sqrt{n-2}} \frac{1}{\mathsf{StdDev}(\boldsymbol{G})} \mathsf{StdDev}\left((\mathbf{Y} - \hat{\beta}_0 - \hat{\beta}_1 \boldsymbol{G}) \right)$$

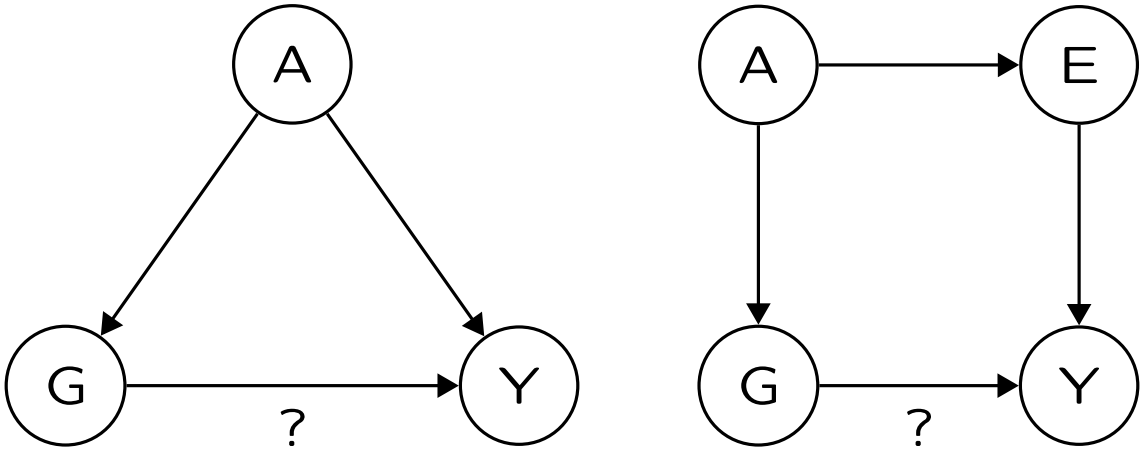
– compare to the robust version;

$$\widehat{\mathsf{StdErr}}(\hat{\beta}_1) = \frac{1}{\sqrt{n-2}} \frac{1}{\mathsf{StdDev}(\boldsymbol{G})} \mathsf{StdDev}\left(\frac{\boldsymbol{G} - \bar{\boldsymbol{G}}}{\mathsf{StdDev}(\boldsymbol{G})} (\mathbf{Y} - \hat{\beta}_0 - \hat{\beta}_1 \boldsymbol{G}) \right).$$

- Tests follow as before, comparing $\pm Z$ to $N(0,1)$ distribution – or slightly heavier-tailed t_{n-2}, though it rarely matters
- Known as *model-based* standard errors, and tests
- These **happen** to be exact tests if the residuals are independent $N(0, \sigma^2)$ for some σ^2, but are large-sample okay under just *homoskedasticity* – constant residual variance

Adjusting for covariates

Suppose you want to adjust for Ancestry, or (perhaps) some environmental variable;



Adjusting for covariates

As mentioned in Part 1, typically 'adjust' for confounders by multiple regression;

$$\text{Mean}(Y) = \beta_0 + \beta_1 G + \gamma_1 PC_1 + \gamma_2 PC_2 + \gamma_3 PC_3 + ...$$

... so β_1 estimates the difference in mean outcome, in those with identical PC values. (Treat other covariates like PCs)

Two equivalent* ways to estimate $\widehat{\beta}_1$;

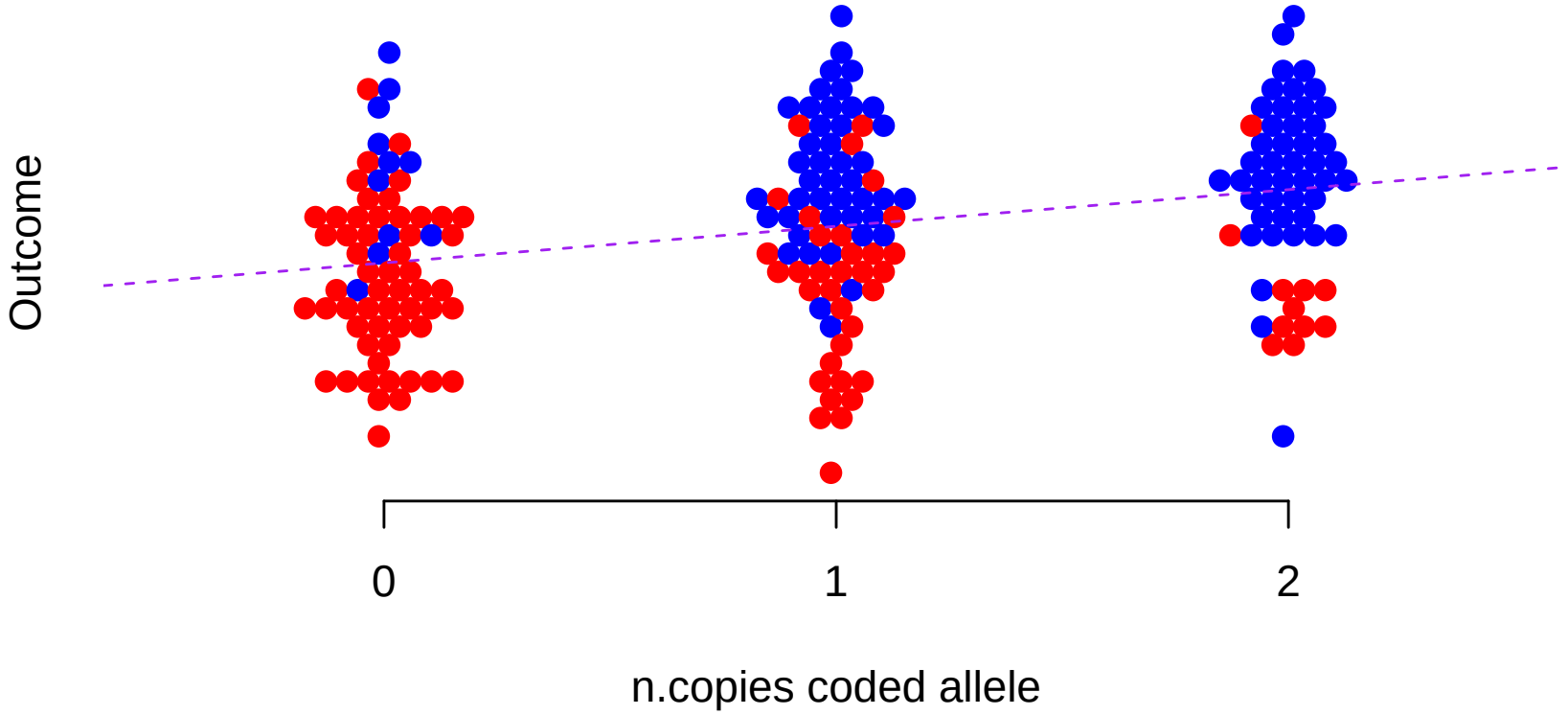
1. Minimize $\sum_{i=1}^{n}(Y_i - \widehat{\beta}_0 - \widehat{\beta}_1 G_i - \widehat{\gamma}_1 PC_{1i} - \widehat{\gamma}_2 PC_{2i})^2$
2. Take residuals regressing Y on PCs, regress them on residuals from regressing G on PCs; slope is $\widehat{\beta}_1$

- Either way, we learn about the association of Y with G after accounting for linear association of Y with the PCs.
- Average slope of Y with G, after accounting for linear effects of confounding

* This equivalence is the Frisch-Waugh-Lovell theorem

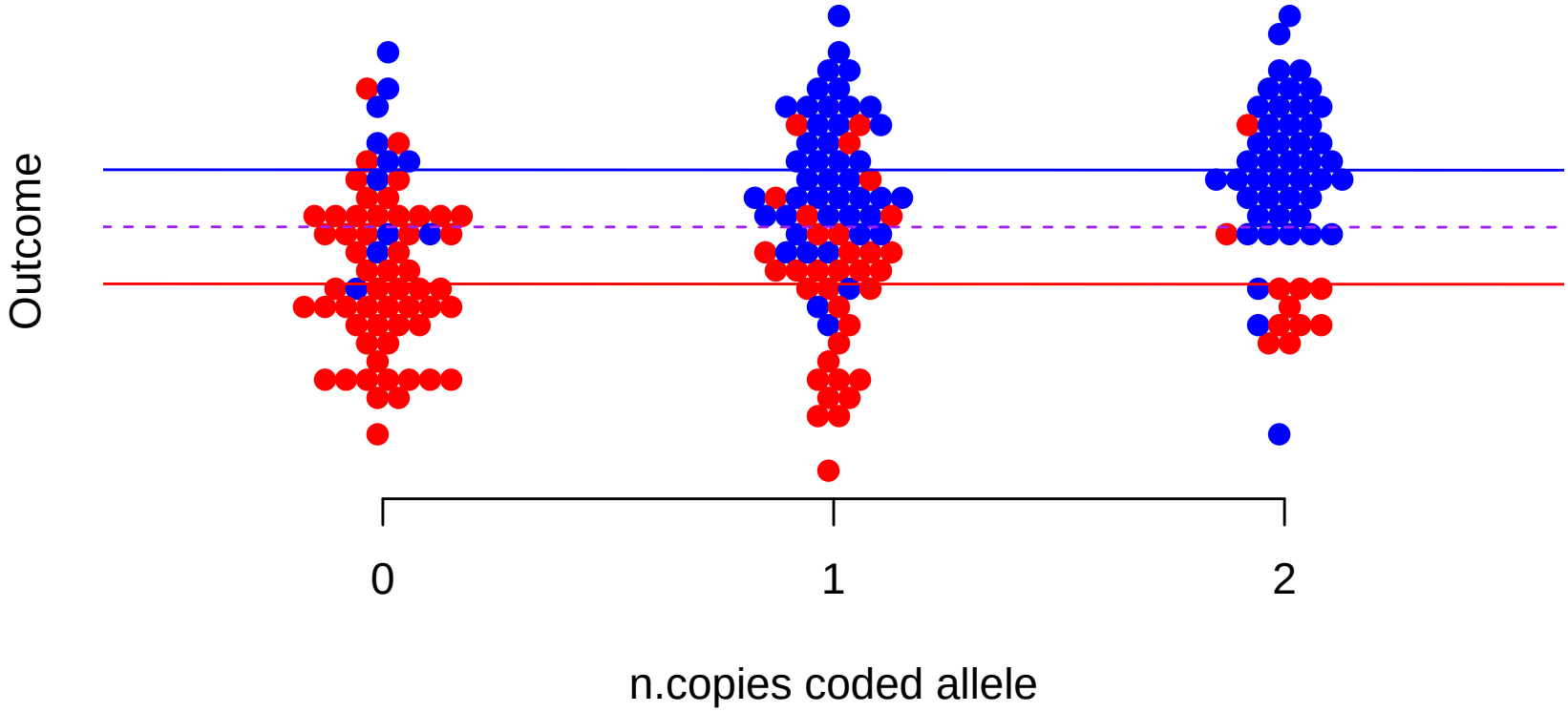
Adjusting for covariates

We often 'adjust for study'. What does this mean? For a purple signal confounded by red/blue-ness...



Adjusting for covariates

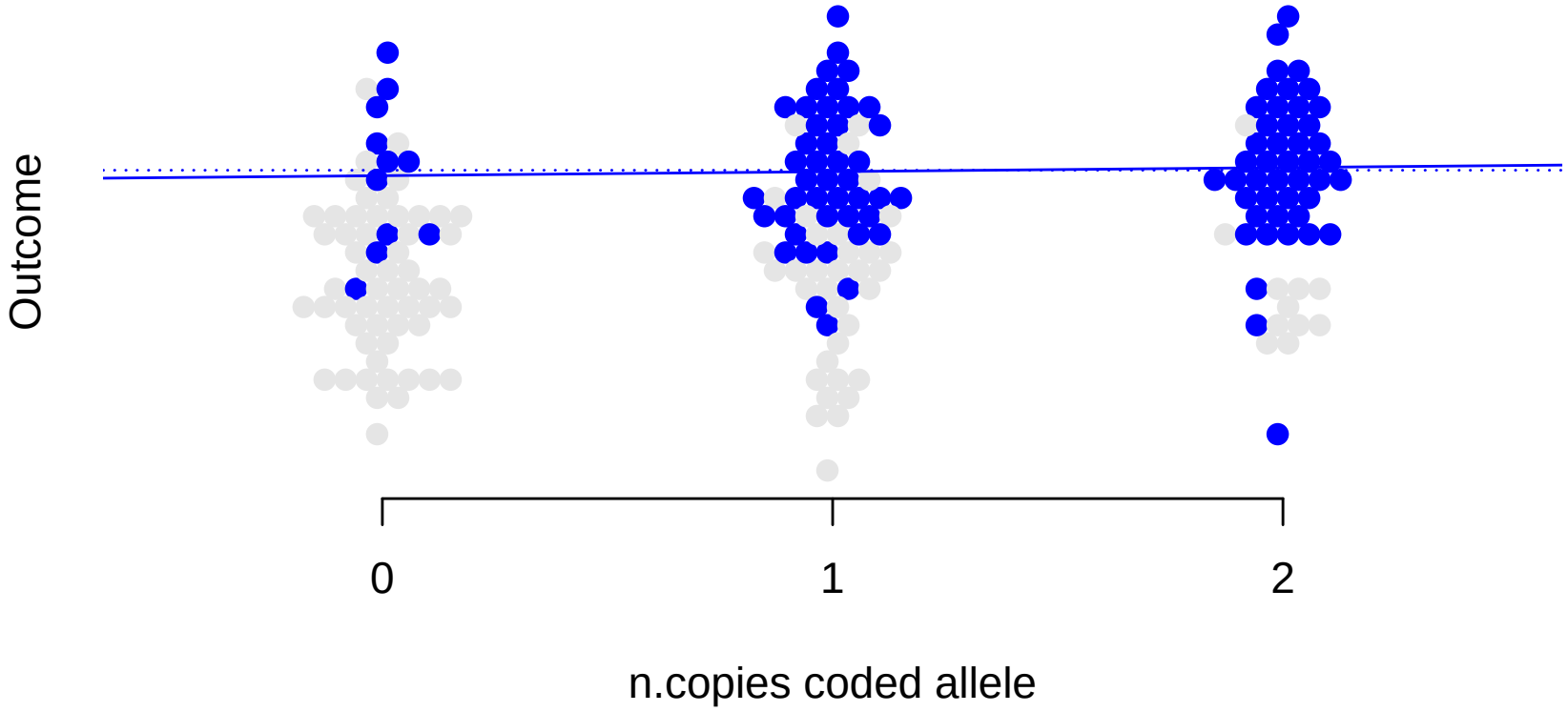
We often 'adjust for study'. What does this mean?
One way: $\widehat{\beta}_1$ is the slope of two parallel lines;



Adjusting for covariates

We often 'adjust for study'. What does this mean?

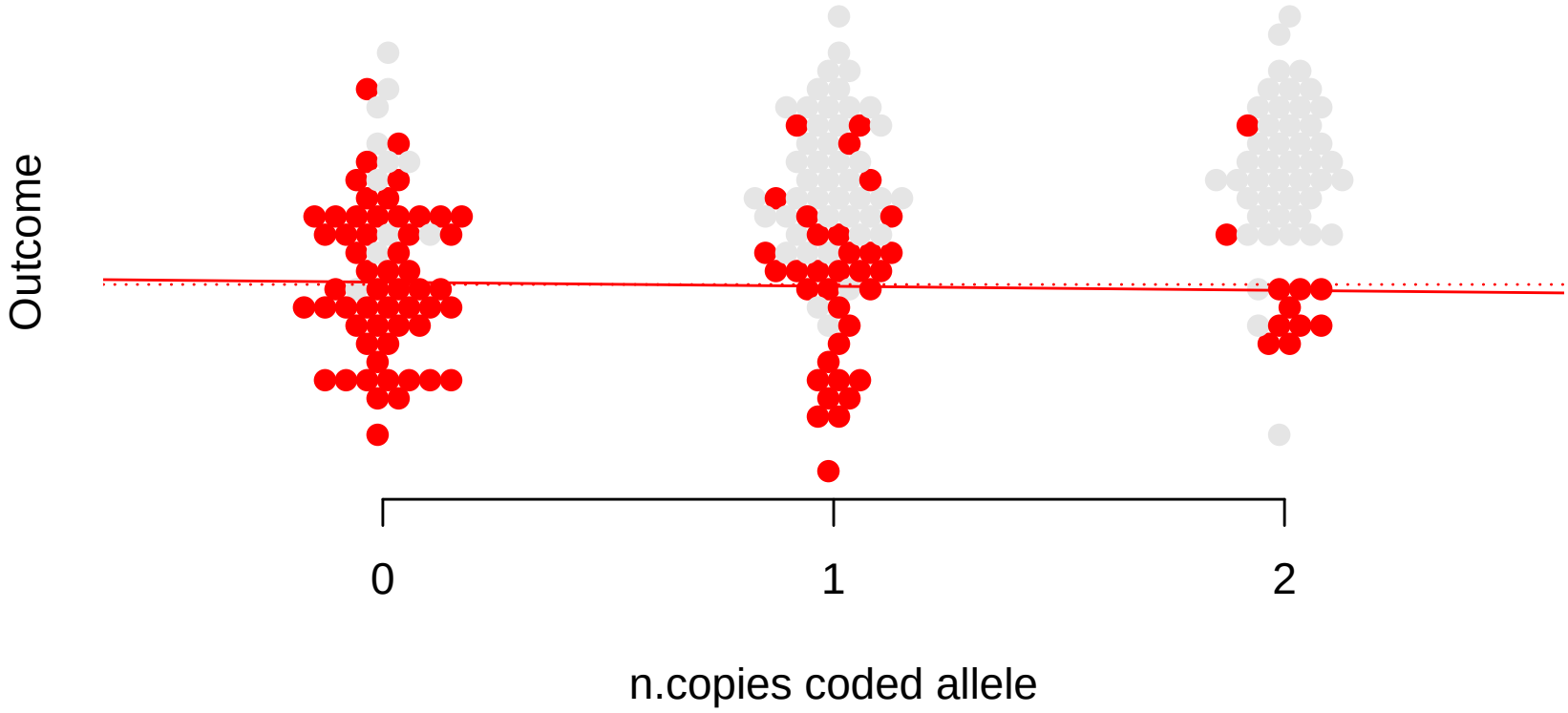
Or: residualize out the mean of one study, then regress on G;



Adjusting for covariates

We often ‘adjust for study’. What does this mean?

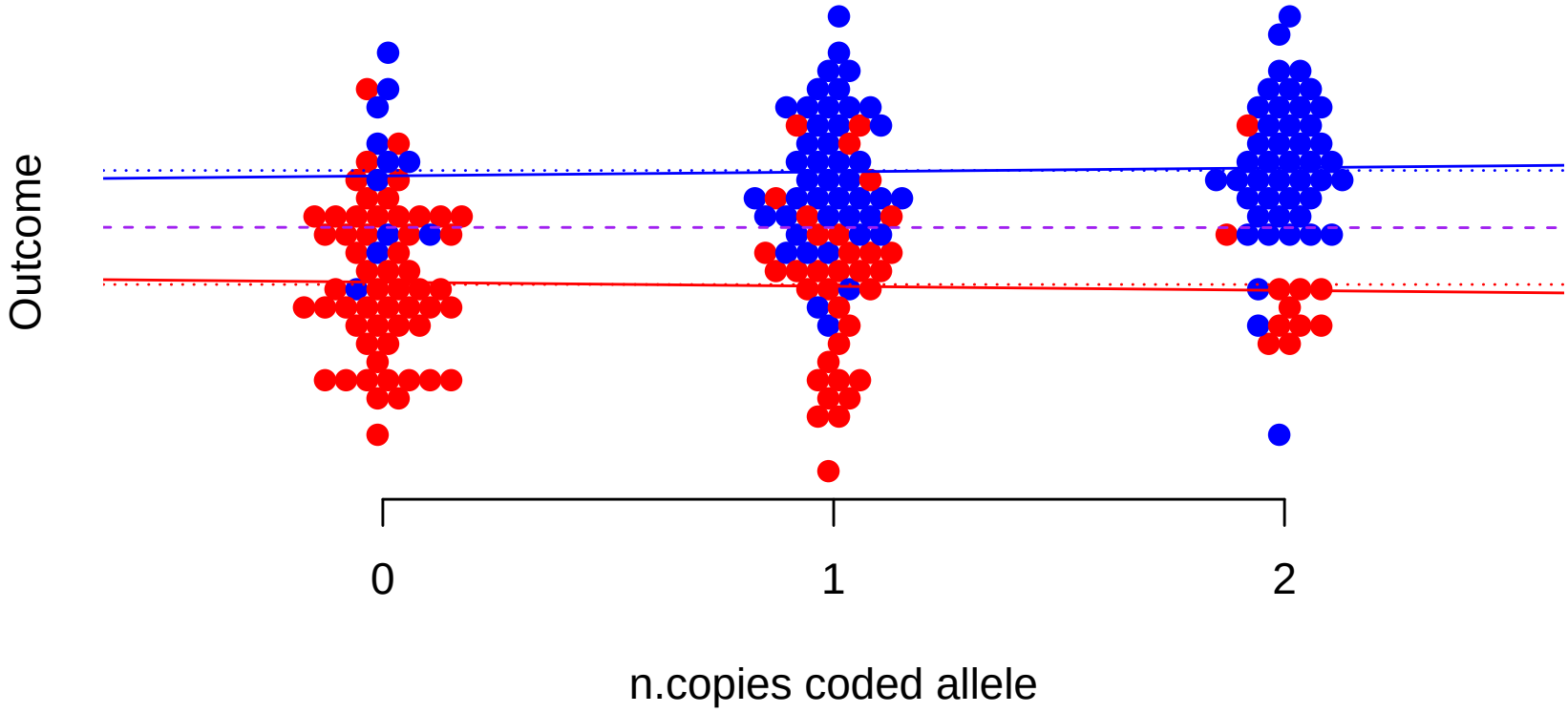
Or: rinse and repeat for the other study;



Adjusting for covariates

We often 'adjust for study'. What does this mean?

Or: average the two study-specific slopes;



Adjusting for covariates

Notes:

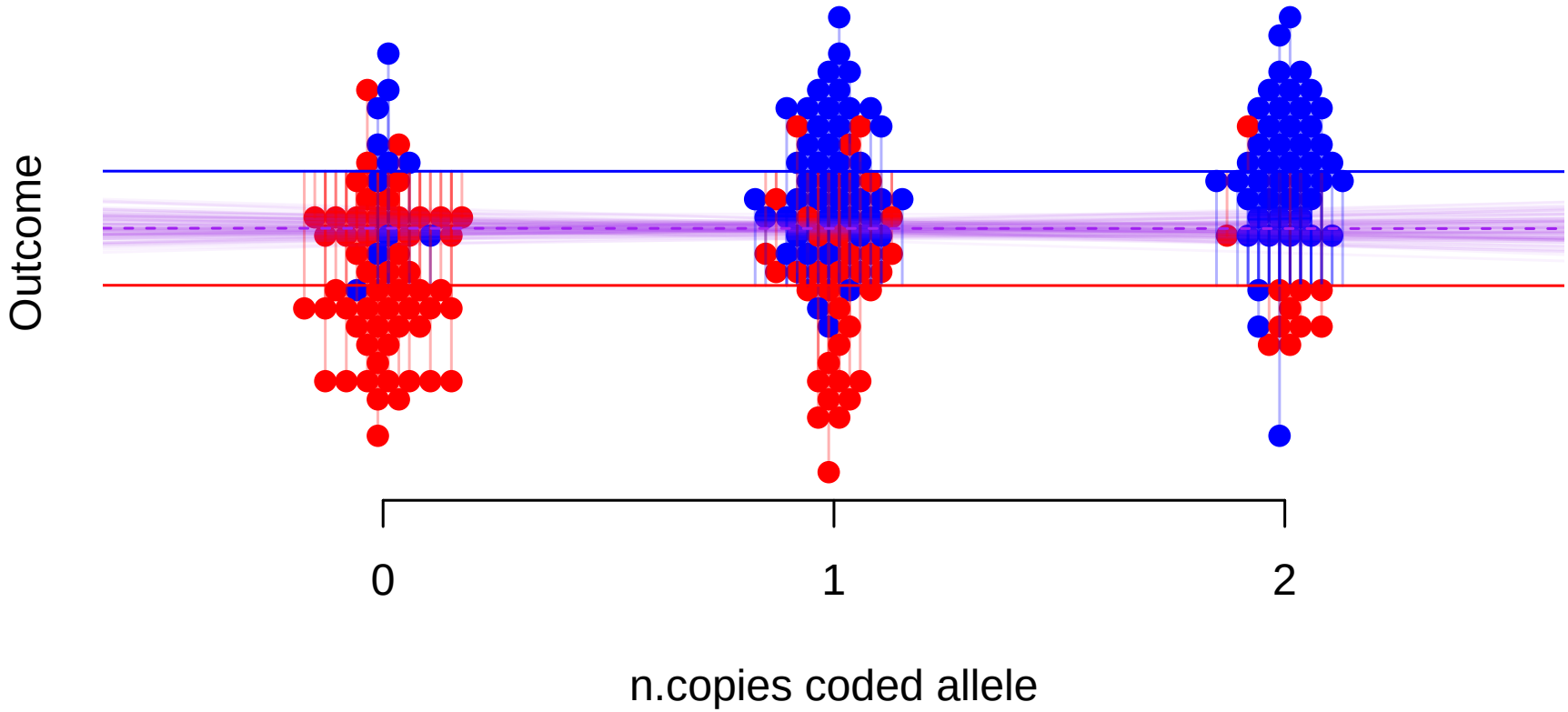
- Adjusting for study only removes trends in the mean outcome. It does not 'fix' any issues with variances
- Different 'slopes' (genetic effects) in different studies don't matter for testing; under the null all the slopes are zero
- The average slope across all the studies is a weighted average – the same precision-weighted average used in fixed-effects meta-analysis, common in GWAS

The two-step approach to adjustment is used, extensively, in our pipeline code – because fitting the null is the same for all variants.

Note: variance estimates ignoring the first step don't work well, even though the $\widehat{\beta}$ estimates are okay (Sofer *et al*, submitted)

Adjusting for covariates

The math for standard error estimates gets spicy (🌶🌶) but;



- Residuals (vertical lines) used empirically as std dev'ns, again
- Homoskedasticity: residual variance is the same for outcomes, regardless of study (or PC values, etc)
- For testing, can estimate residual standard deviation with G effect fixed to zero, or fitted

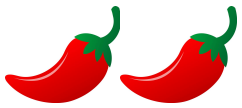
Allowing for relatedness

With relateds, linear regression minimizes a different criterion;

Unrelated	$\sum_{i=1}^{n} (Y_i - \beta_0 - \beta_1 G_i - ...)^2$
Related	$\sum_{i,j=1}^{n} R_{ij}^{-1} (Y_i - \beta_0 - \beta_1 G_i - ...)(Y_j - \beta_0 - \beta_1 G_j - ...)$

where R is a *relatedness matrix* – as seen on Wednesday.

- If we think two outcomes (e.g. twins) are highly positively correlated, should pay less attention to them than two unrelateds
- But those in affected families may carry more causal variants, which provides power
- Residual variance assumed to follow pattern given in R – though minor deviations from it don't matter



Math-speak for this criteria is $(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T R^{-1} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T$, and the estimate is $\hat{\boldsymbol{\beta}} = (\mathbf{X}^T R^{-1} \mathbf{X})^{-1} \mathbf{X}^T R^{-1} \mathbf{Y}$. The resulting $\hat{\beta}_1$ generalizes what we saw before – and is not hard to calculate, with standard software.

Allowing for relatedness

If $\sigma^2 R$ is the (co)variance of all the outcomes, the variance of the estimates is

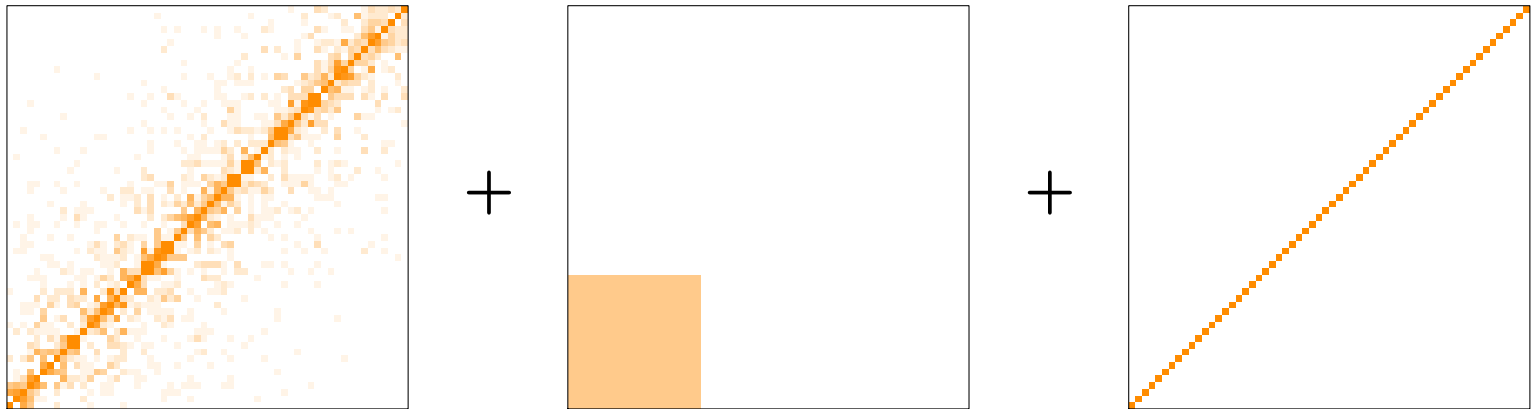
$$\widehat{\text{Cov}}(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}^T R^{-1}\mathbf{X})^{-1}$$

This too doesn't present a massive computational burden – but we skip the details here.

- Regardless of chili-pepper math, tests of $\beta_1 = 0$ can be constructed in a similar way to what we saw before
- Can allow for more complex variance/covariance than just R scaled – see next slide. To allow for estimating σ^2 terms, recommend using REML when fitting terms in variance, a.k.a. fitting a *linear mixed model* (LMM)
- Use of LMMs is model-based and not 'robust'; we do assume the form of the variance/covariance of outcomes
- Robust versions are available, but work poorly at small n

Allowing for relatedness

R can be made up of multiple *variance components*. For each pair of observations it can be calculated with terms like...

$R =$  $+$ $+$

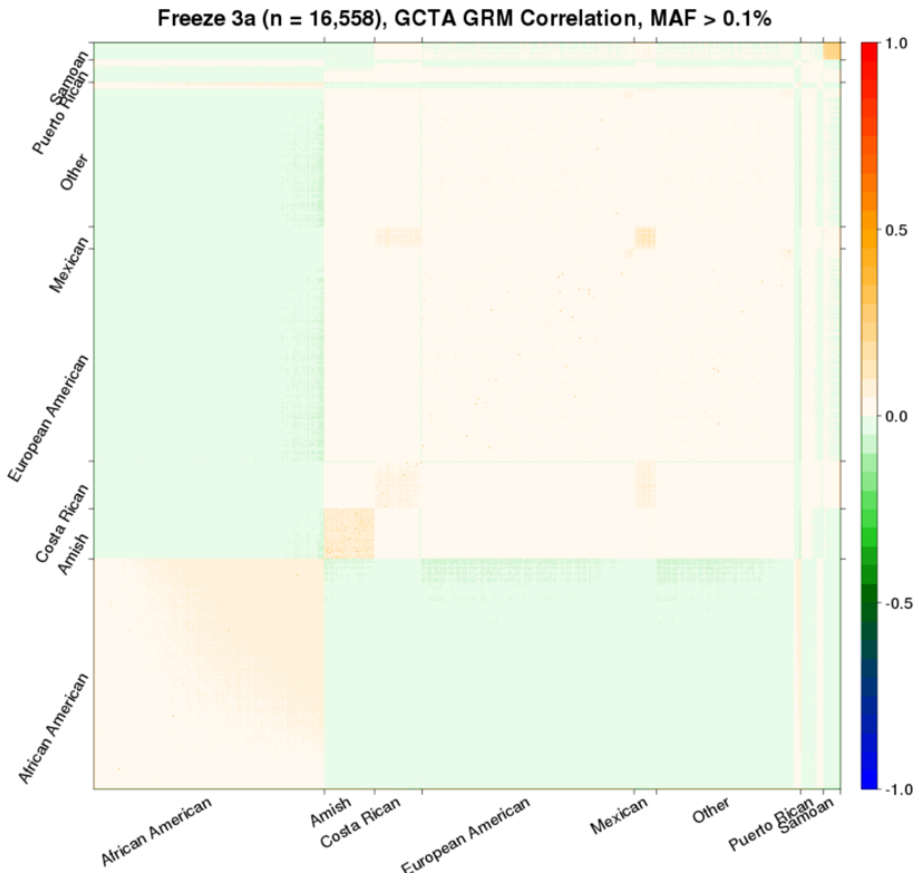
$= \quad \sigma^2_G \times$ Kinship $\quad + \quad \sigma^2_{\text{Cluster}} \quad + \quad \sigma^2_\epsilon$ (noise).

- The σ^2 terms – for familial relatedness, 'random effects' for clusters such as race and/or study, and a 'nugget' for individual noise – are fit using the data
- Can get $\hat{\sigma}^2 = 0$; do check this is sensible with your data

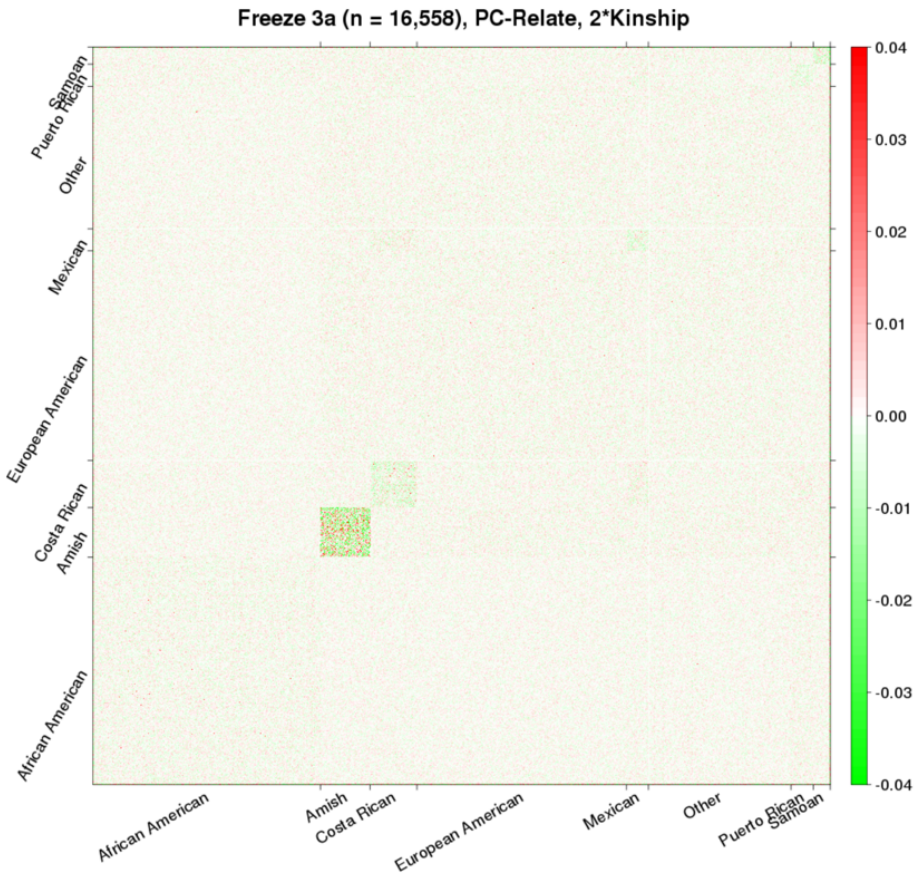
Allowing for relatedness

Note: using GRM $\approx$ adding random effects – blocks in R – for AA/non-AA. Kinship matrices don't do this;

Scaled GRM (correlation)

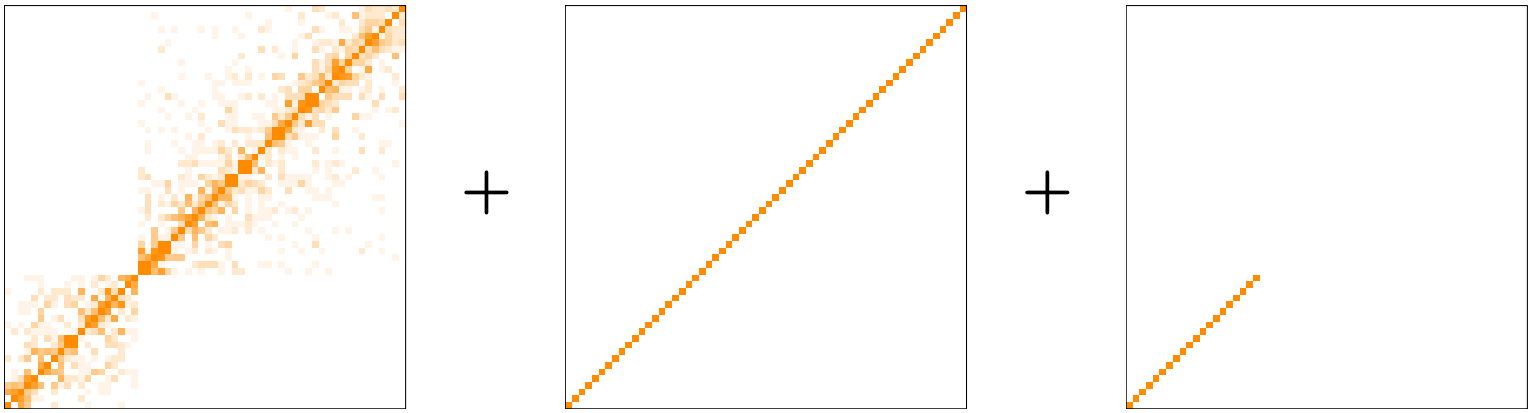
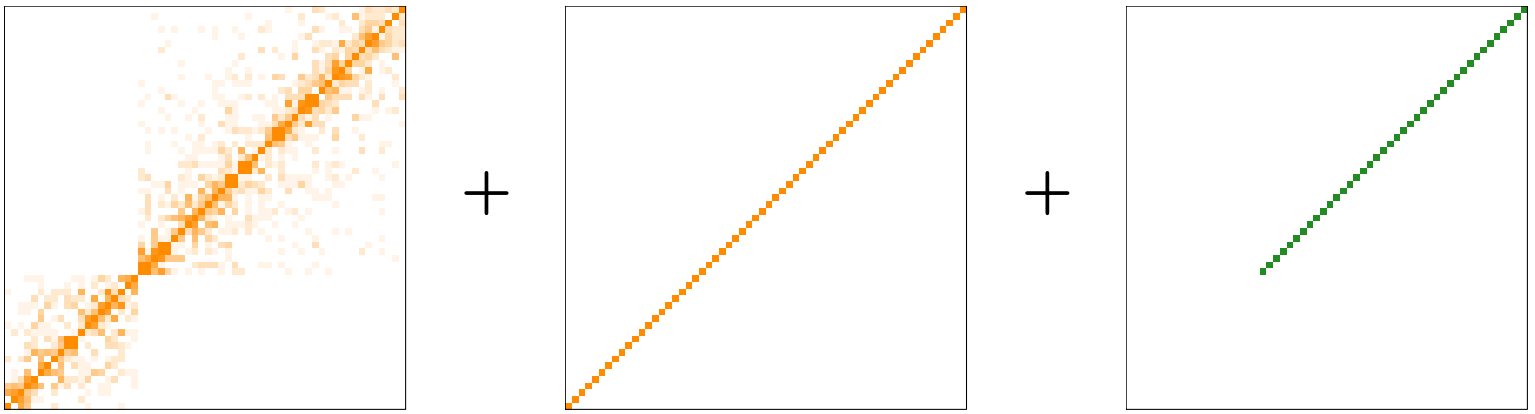


kinship



Allowing for non-constant variance

Measurement systems differ, so trait variances may vary by study (or other group). To allow for this – in studies A and B;

$R_A =$  $R_B =$ 

$= \sigma^2_G \times$ Kinship $+ \sigma^2_\epsilon +$ σ^2_{Study} (noise).

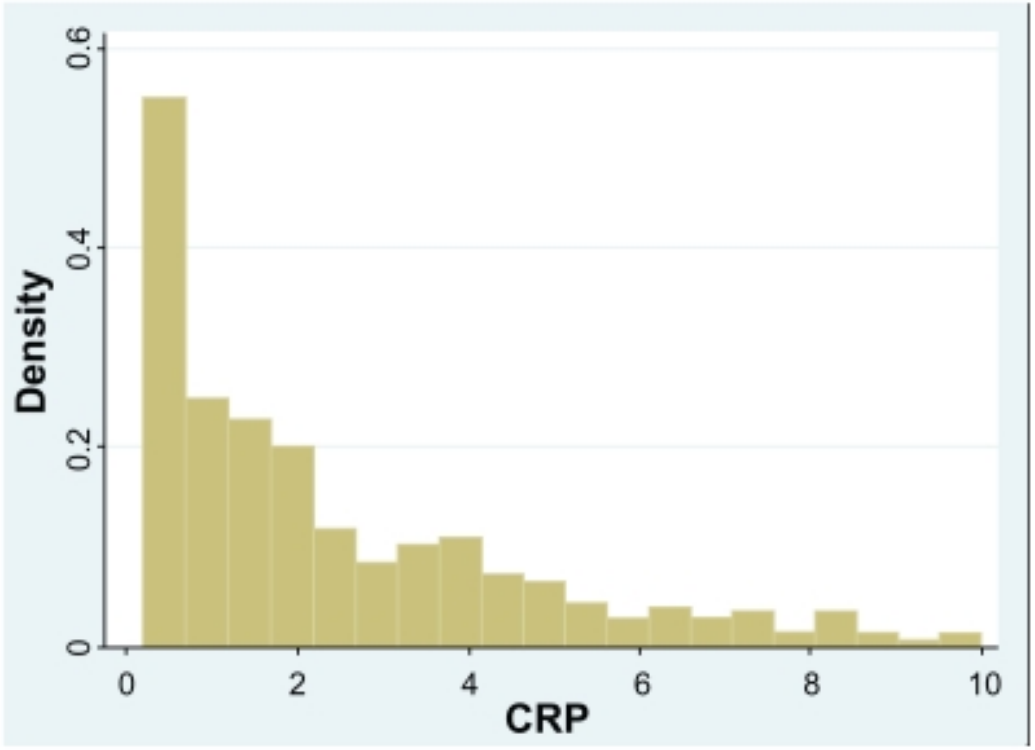
Allowing for non-constant variance

Notes:

- Non-constant variance may be unfamiliar; in GWAS meta-analysis it was automatic that each study allowed for its own 'noise'
- Ignoring it is dangerous – can lead to false positives, and lack of efficiency
- We've found it plays an important role in TOPMed analyses – see Rice et al 2017, Sofer et al, 2018, and upcoming applied papers
- Examine study- or ancestry-specific residual variances from a model with confounders only, and see if they differ. (Example coming in the hands-on material)

Trait transformations

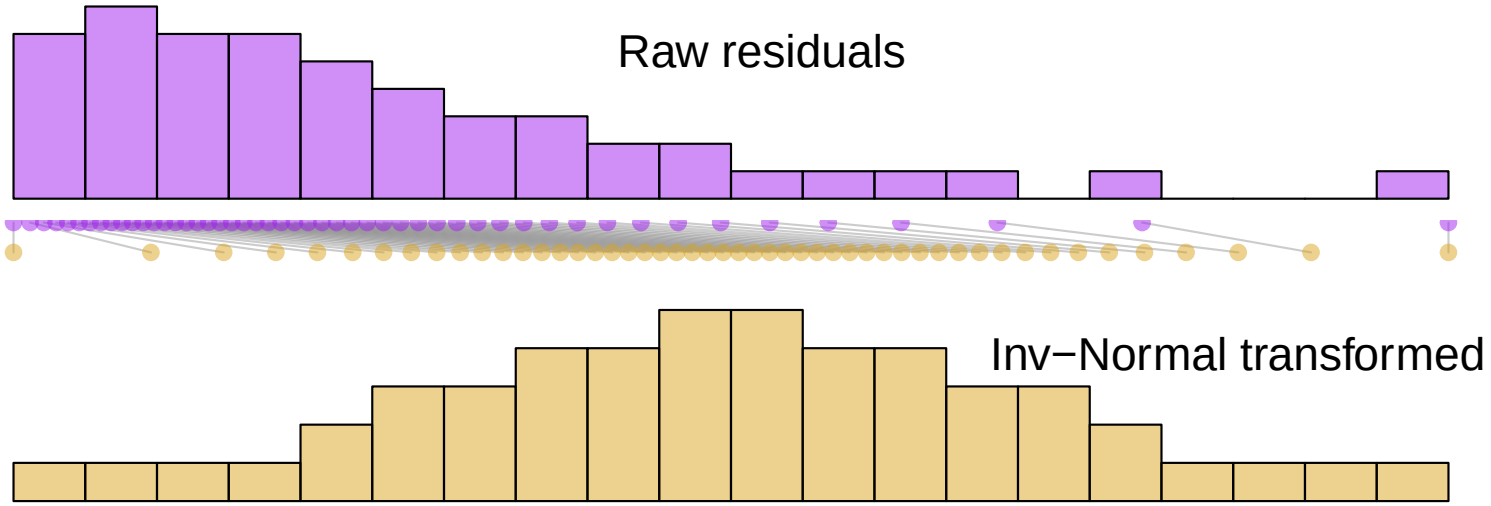
Some traits have **heavy tails** – this is, extreme observations that make the approximations used in p-value calculations less accurate.



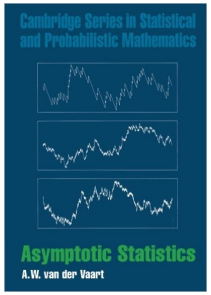
- Positive skew here – heavy right tail
- Here, the genotype of the individuals with CRP≈10 will affect $\hat{\beta}_1$ much more than those with CRP≈1
- Effectively averaging over fewer people, so approximations work less well

Trait transformations

For positive skews, log-transformations are often helpful – and already standard, so Working Groups will suggest them anyway. Another popular approach (after adjusting out confounders) is *inverse-Normal transformation* of residuals;

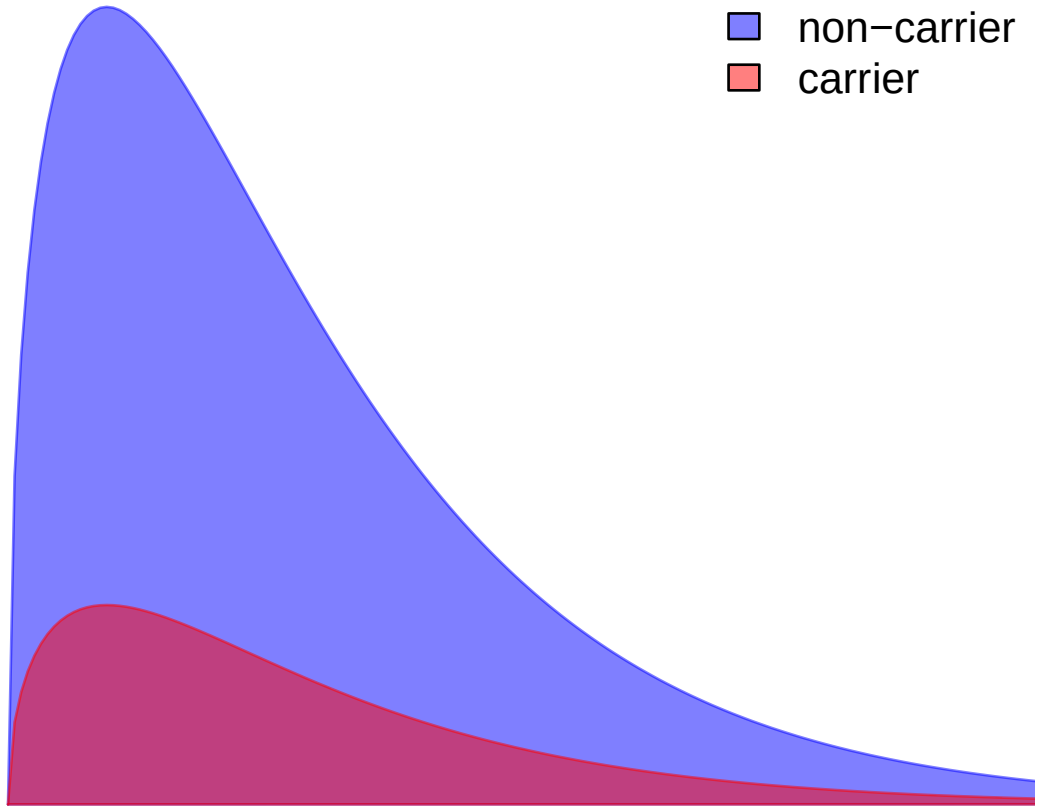


The **basic idea** is that this is as close as we can get to having the approximations work perfectly, at any sample size. For details (🌶🌶🌶🌶🌶) see Chapter 13...



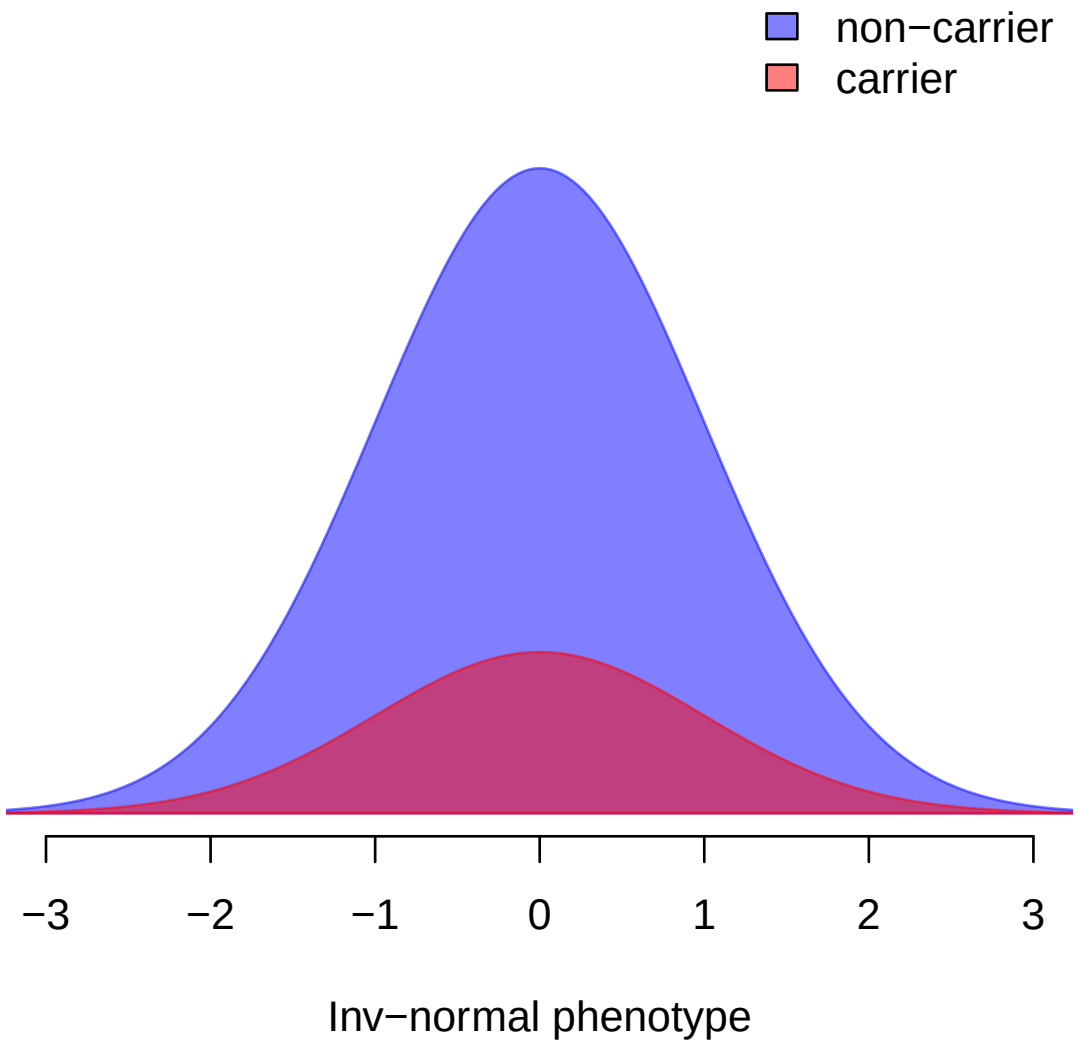
Trait transformations

Under a strong null hypothesis, inv-N works well **for testing**;



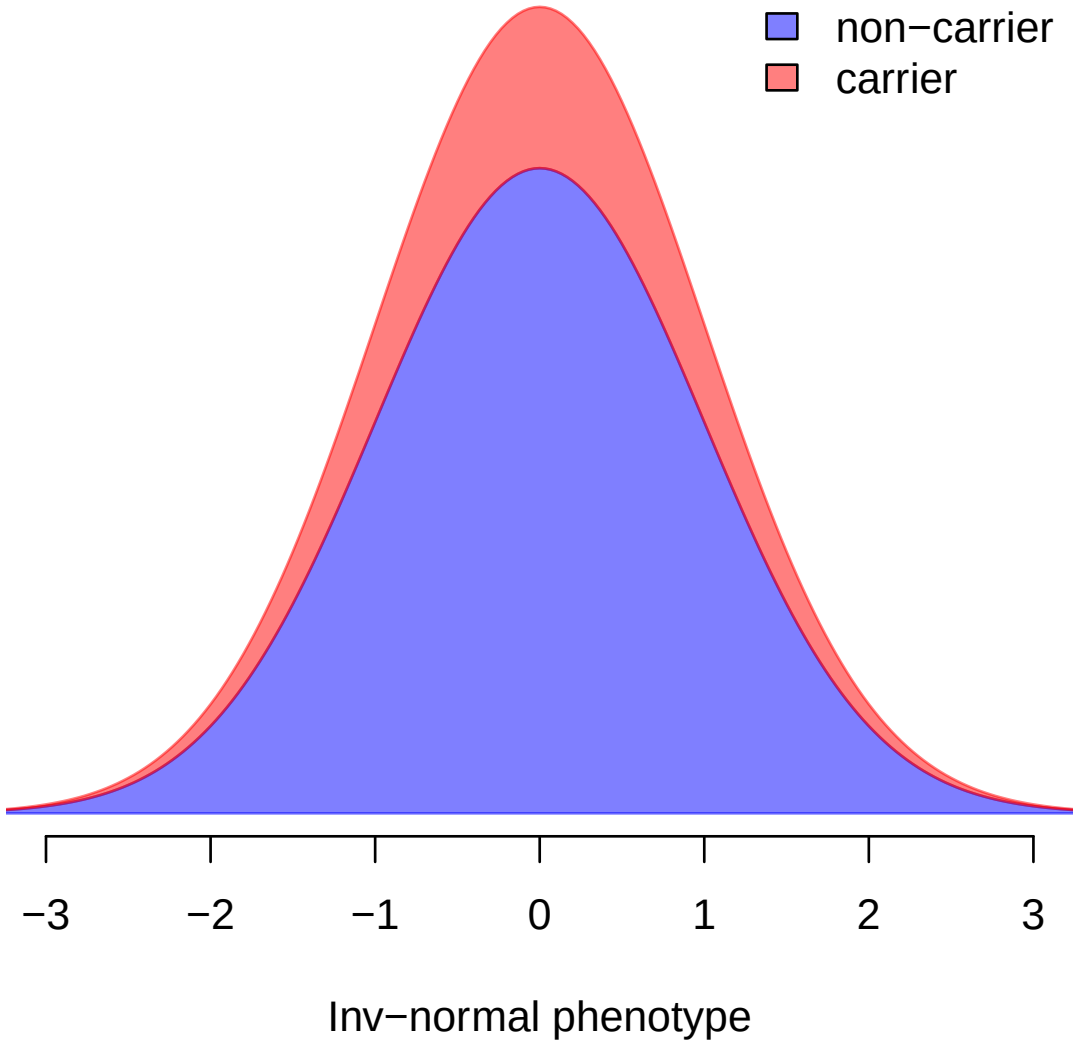
Trait transformations

Under a strong null hypothesis, inv-N works well **for testing**;



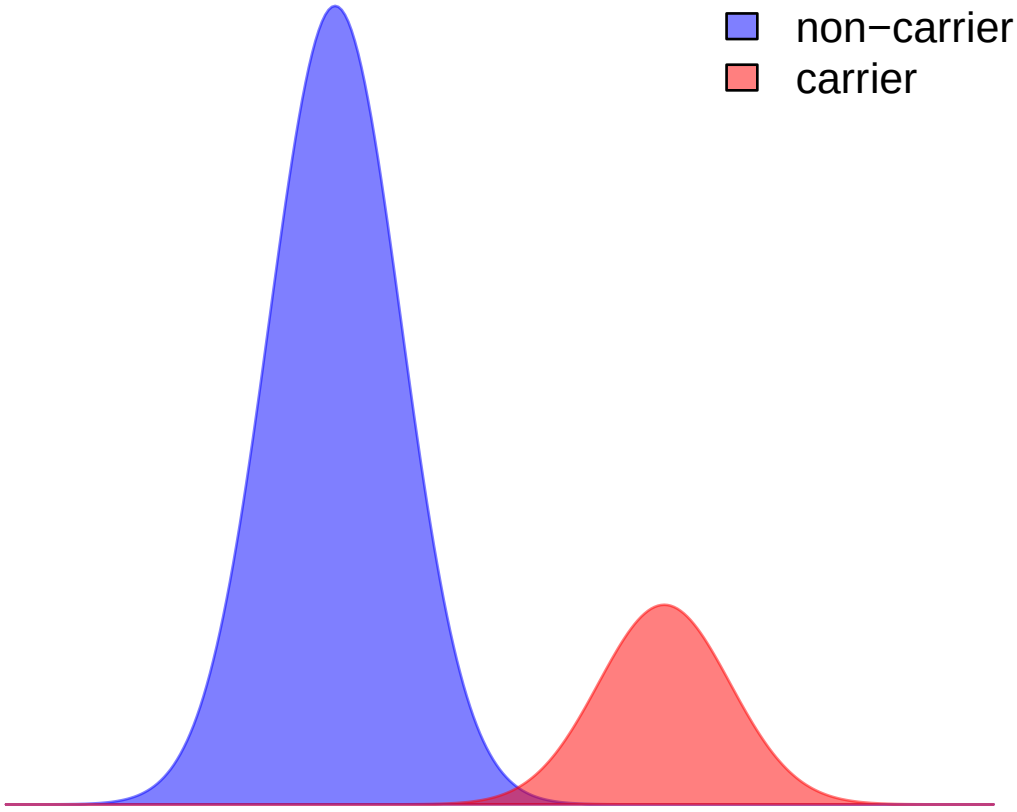
Trait transformations

Under a strong null hypothesis, inv-N works well **for testing**;



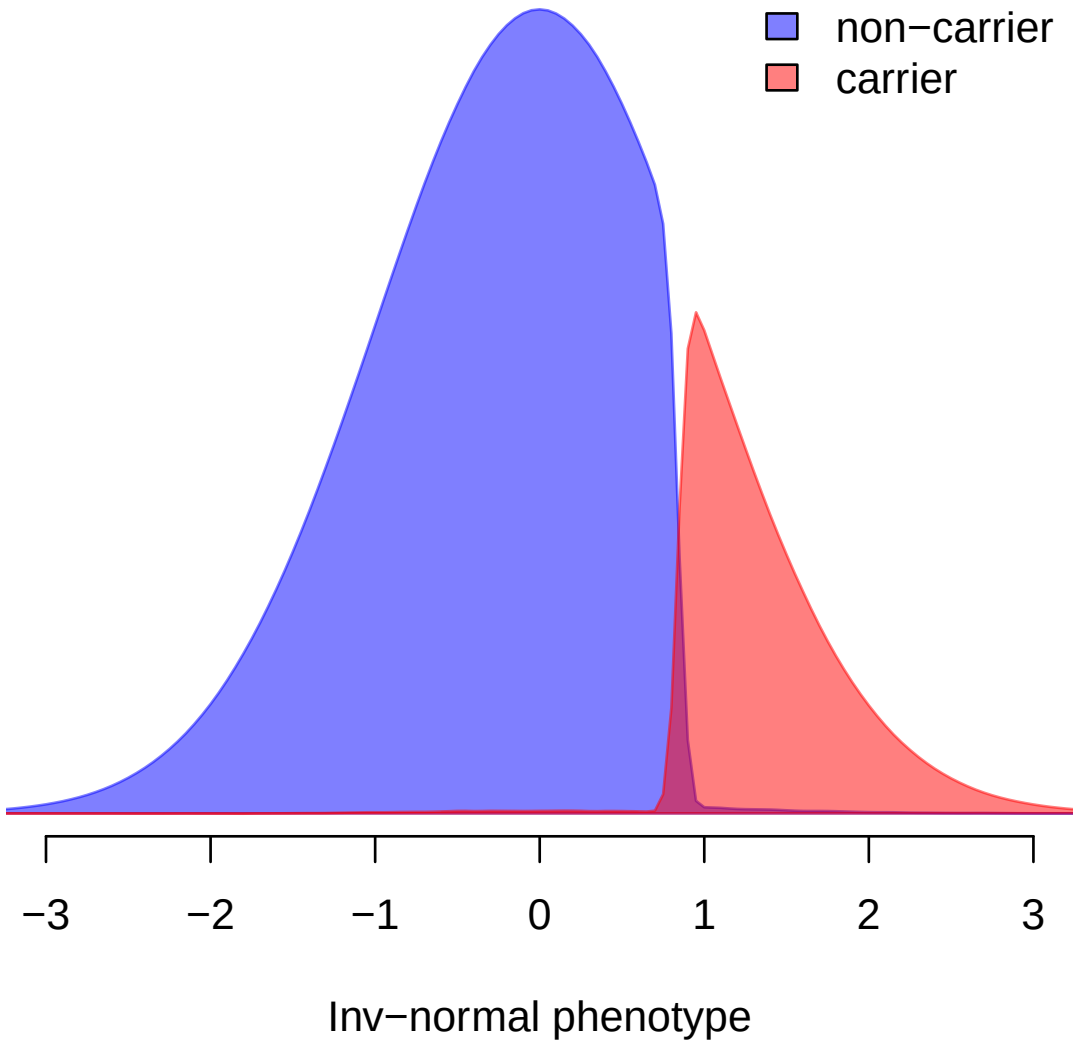
Trait transformations

But it can cost you power;



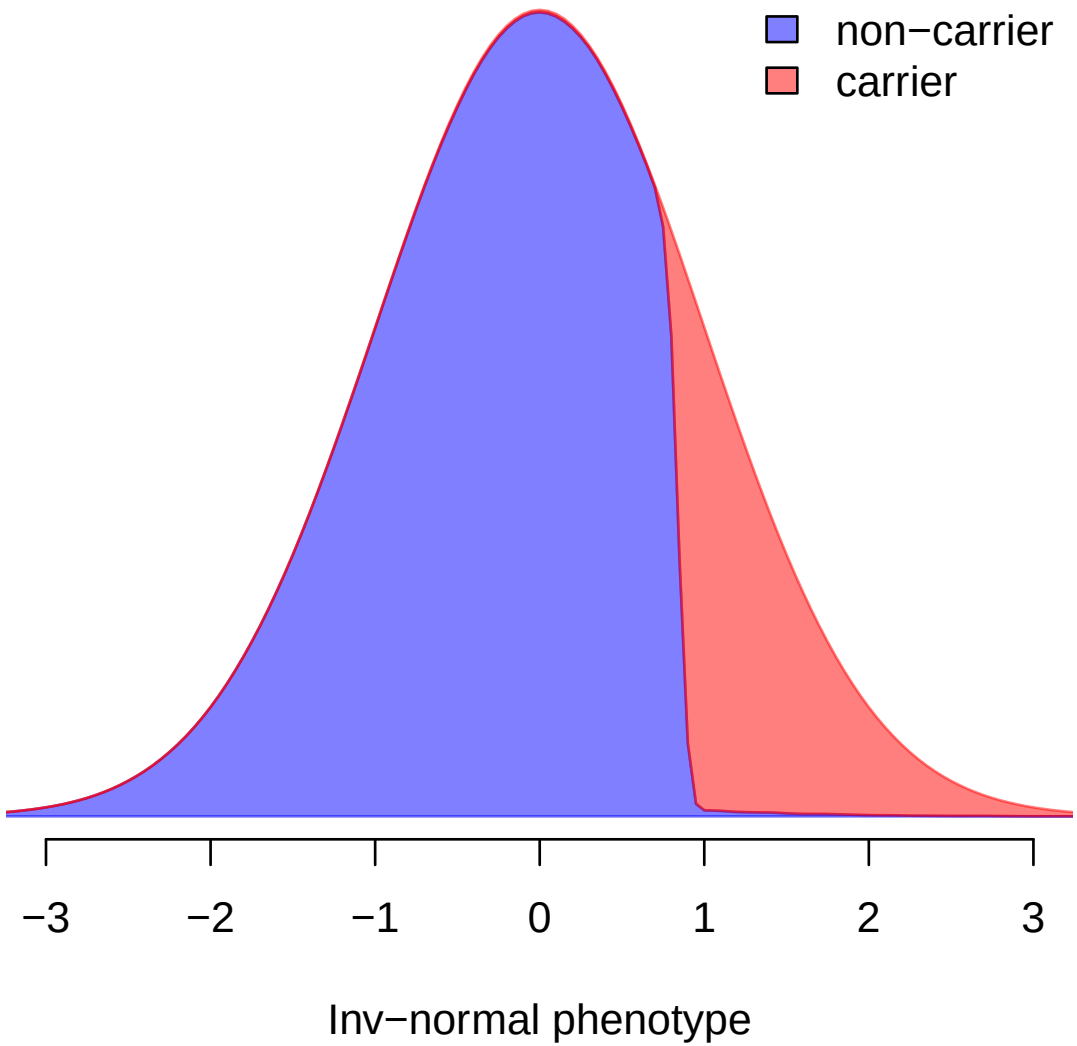
Trait transformations

But it can cost you power;



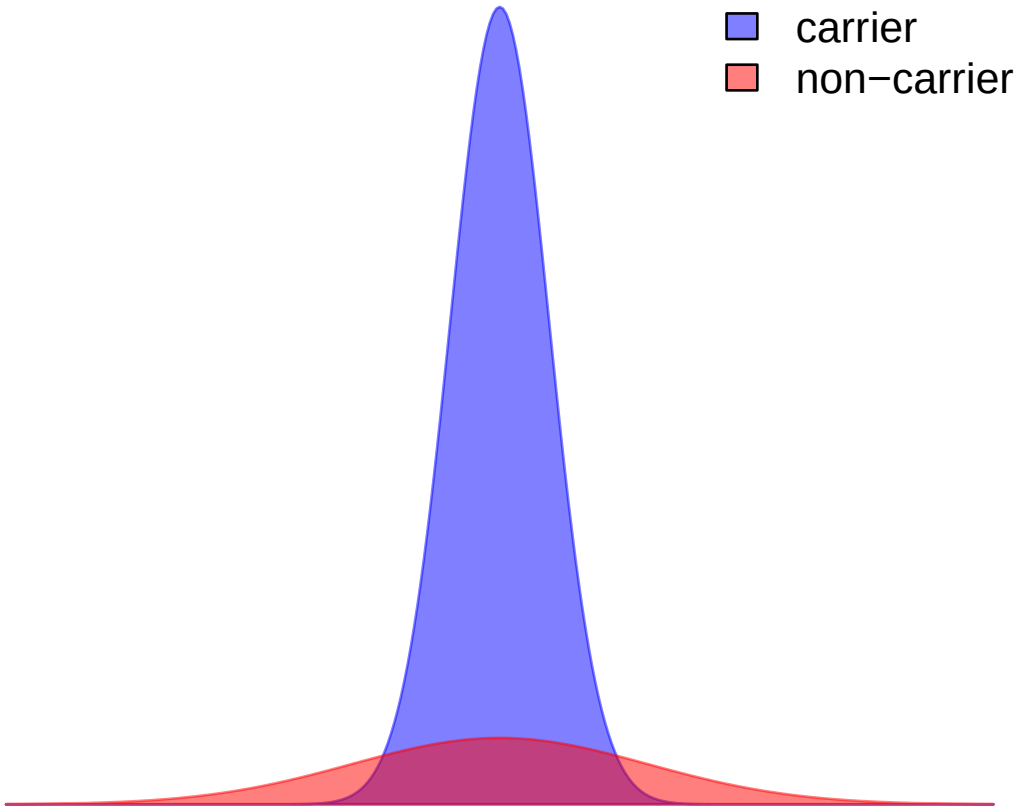
Trait transformations

But it can cost you power;



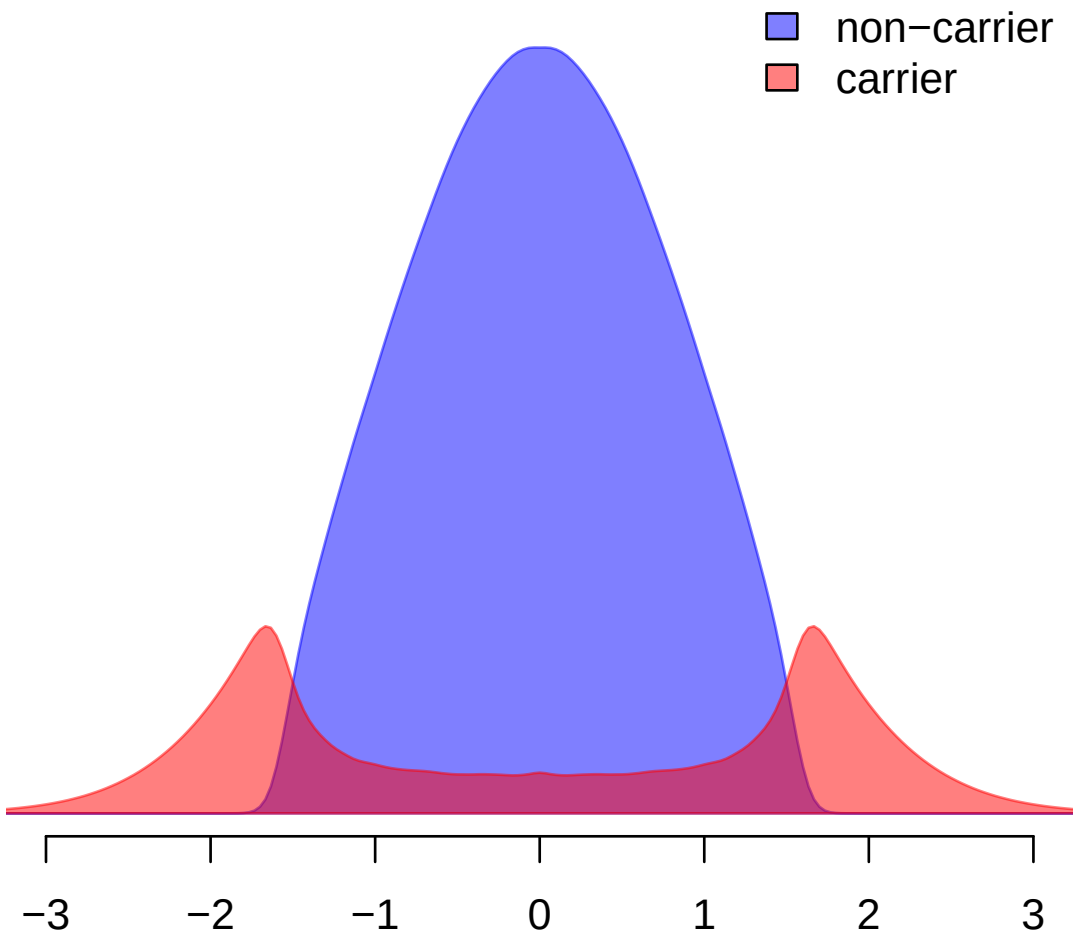
Trait transformations

Under weak nulls, it can even make things worse;



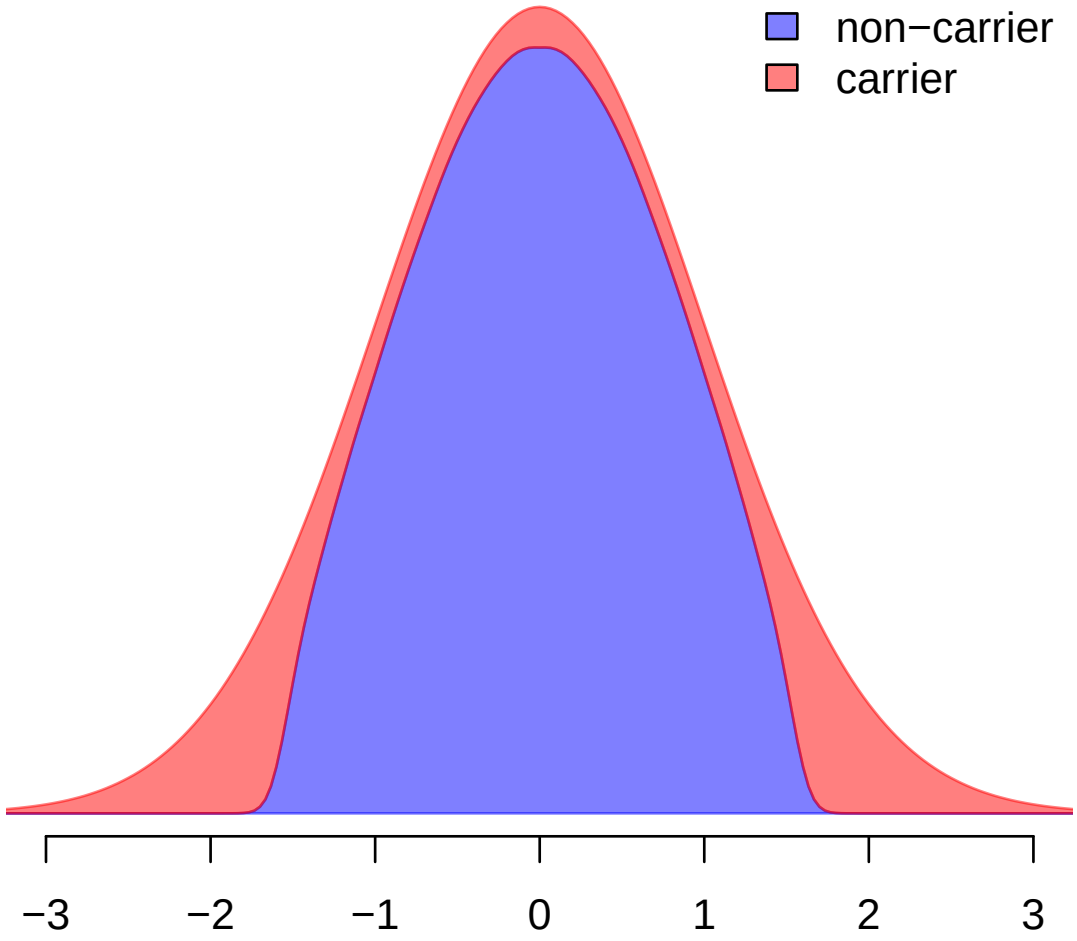
Trait transformations

Under weak nulls, it can even make things worse;



Trait transformations

Under weak nulls, it can even make things worse;



Trait transformations

What we do in practice, when combining samples from groups with different variances for a trait (e.g., study or ancestry)

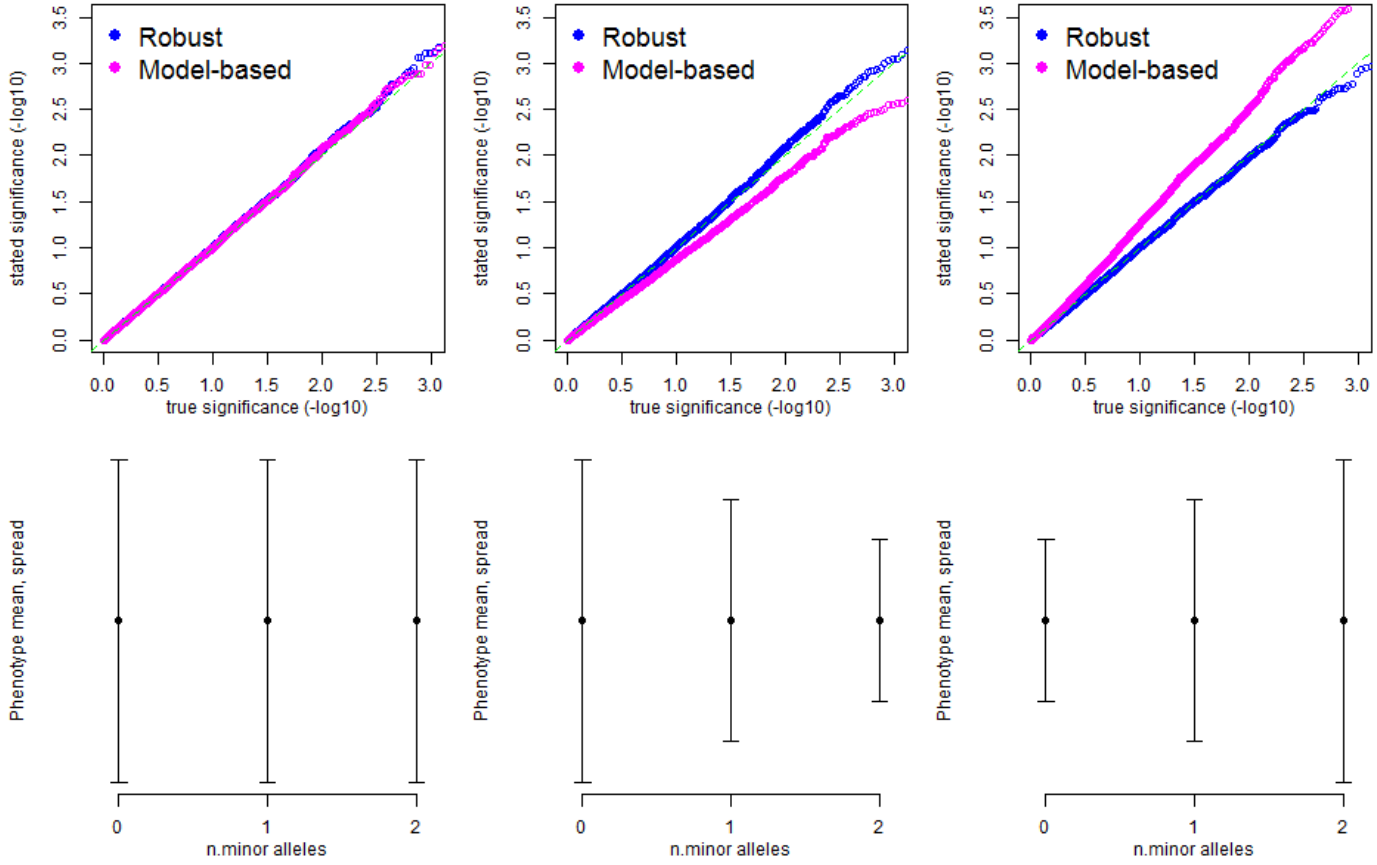
- Fit null model; effects for confounders, kinship/GRM terms in R, allowing for heterogeneous variances by group
- Inv-N transform the (marginal) residuals
- Within-group, rescale transformed residual so variances match those of untransformed residuals
- Fit null model (again!) – effects for confounders, kinship/GRM terms, but now allowing for heterogeneous variances by group

The analysis pipeline implements this procedure – examples follow in the hands-on session.

Why adjust twice? Adjustment only removes linear terms in confounding signals – and inv-Normal transformation can be strongly non-linear

When things go wrong

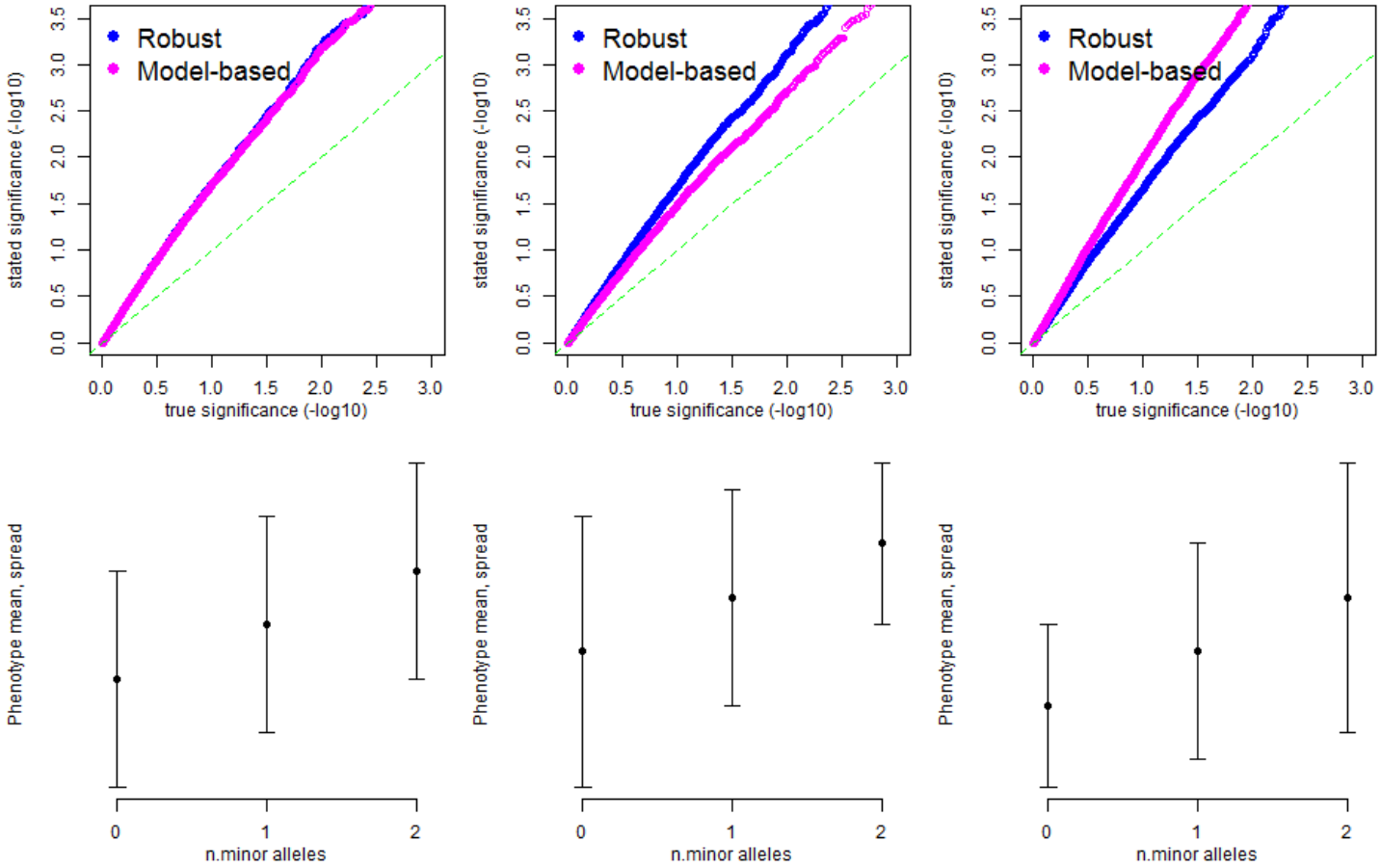
In linear regressions, we assumed constant variance of outcome across all values of G (and any other confounders). This may seem harsh: SNPs affecting variance are interesting! But...



Looks just like stratification – and is hard to replicate.

When things go wrong

Even when there is an undeniable signal, the efficiency depends on how the 'noise' behaves;



When things go wrong

'Try it both ways and see' doesn't always help;

- We don't know the truth, and have few 'positive controls' (if any – we want to find new signals!)
- We have fairly crude tools for assessing WGS results – just QQ plots, genomic control λ
- With one problem fixed, others may still remain

Planning your analysis really does help;

- What known patterns in the trait should we be aware of? (What was the design of the studies? How were there measurements taken?)
- What confounding might be present, if any?
- What replication resources do we have? How to ensure our discoveries are likely to also show up there?

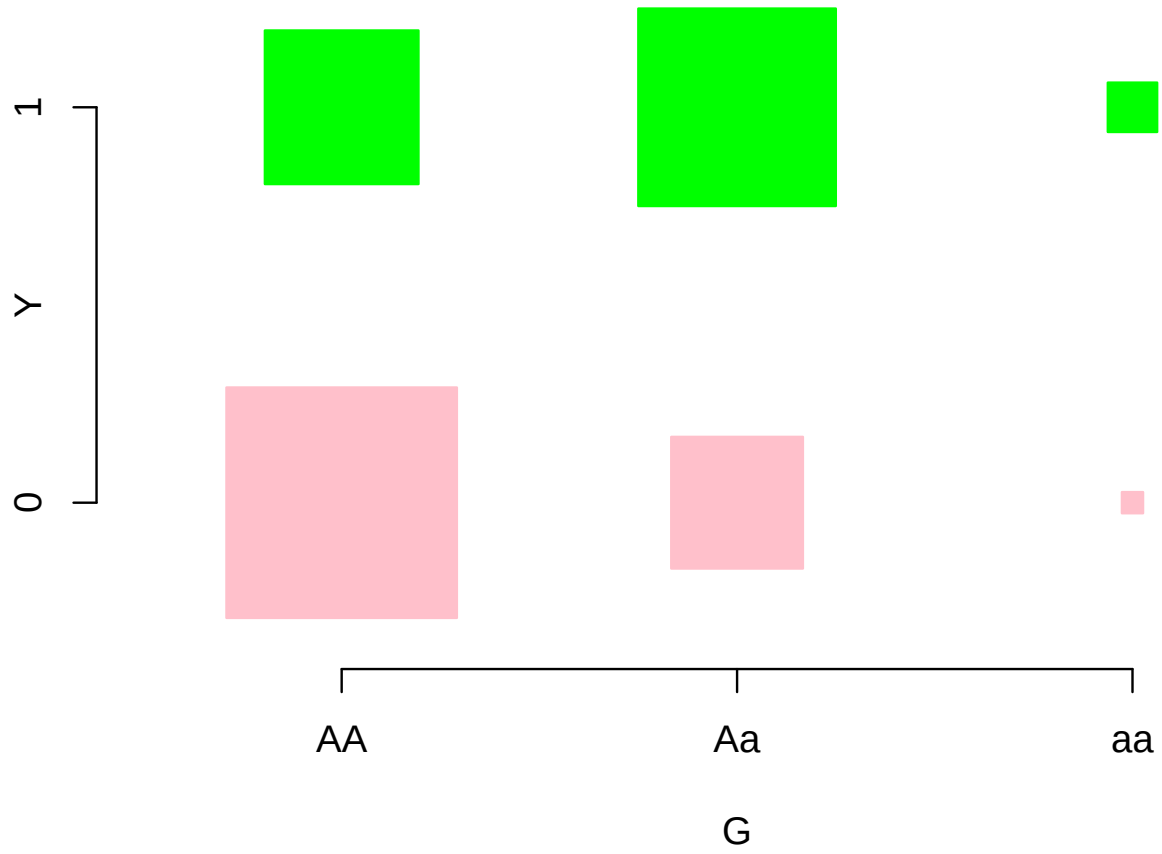
If problems persist, seek help! You can ask the DCC, IRC, Analysis Committee, etc

Binary outcomes

Cardiovascular researchers *love* binary outcomes!

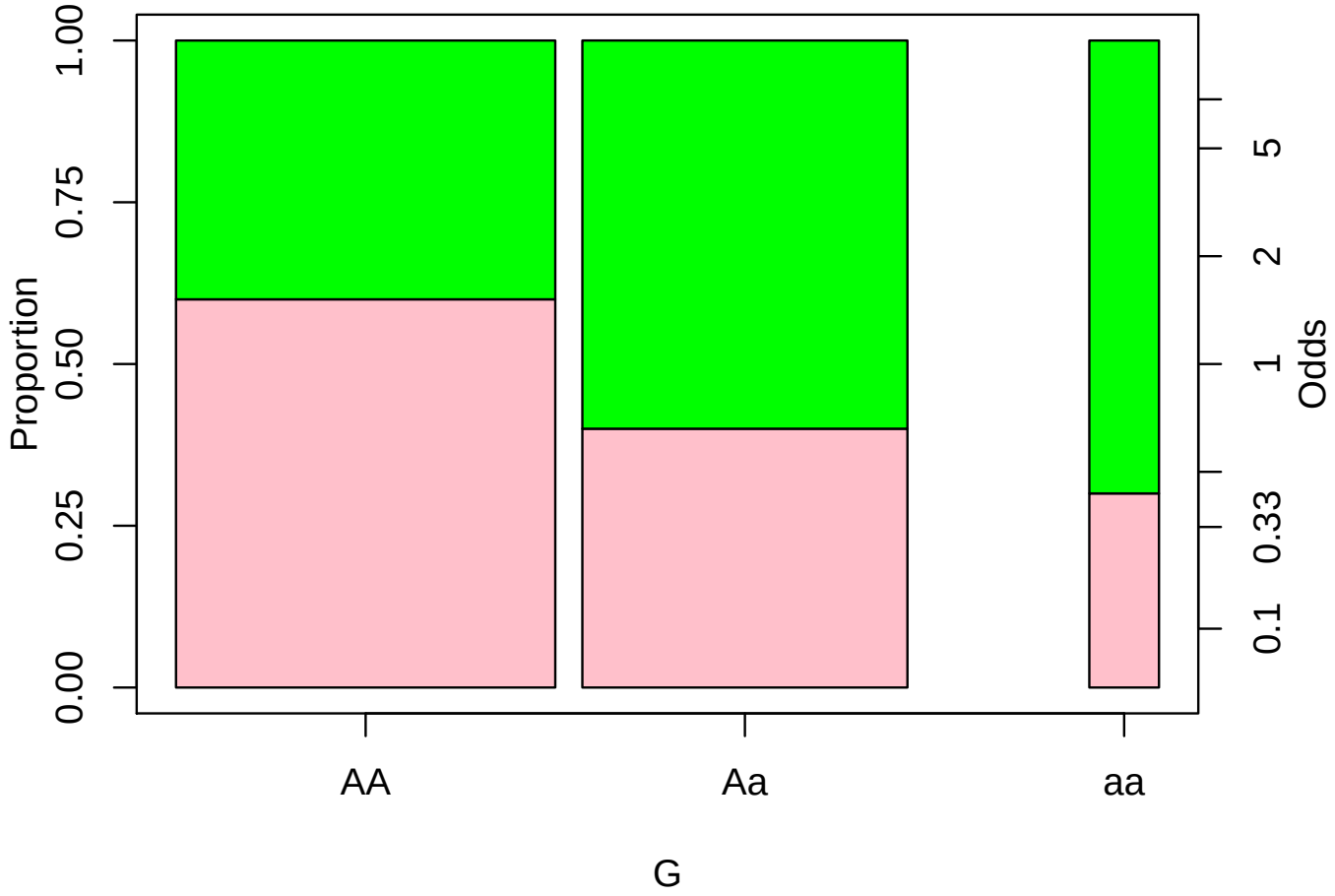
Binary outcomes

Counts of genotype, for people with pink/green outcomes;



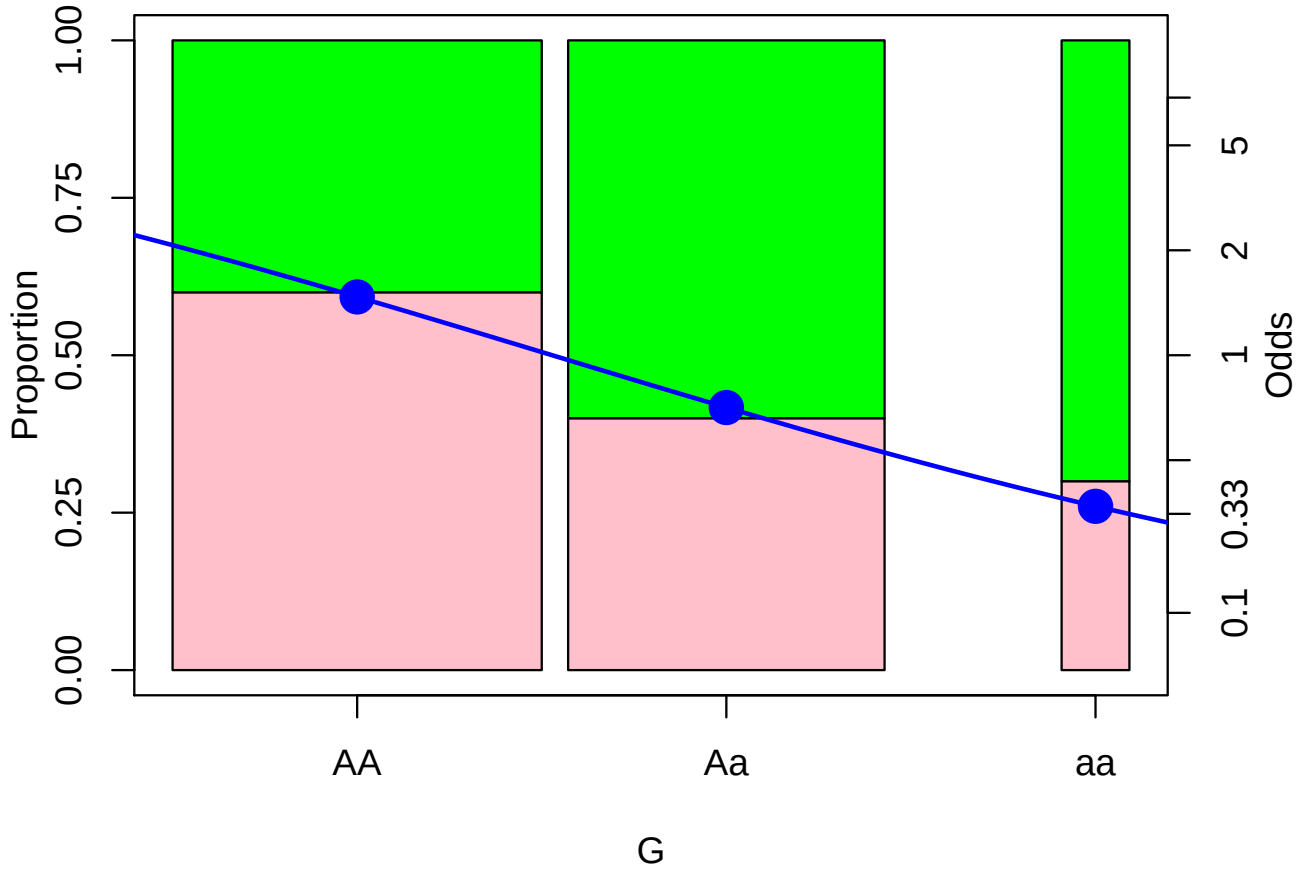
Binary outcomes

Coding the outcome as 0/1, 'mean outcome' just means 'proportion with 1', i.e. 'proportion with disease';



Binary outcomes

Logistic regression estimates an **odds ratio**, describing (averaged over the whole population) how the odds of the event are **multiplied**, per 1-unit difference in G;



With unrelated Y, G, this slope is flat; $OR = 1, \log(OR) = 0$.

Binary outcomes

The math of that *logistic-linear* curve;

$$\mathbb{P}[\,Y = 1|G = g\,] = \frac{e^{\beta_0+\beta_1 g}}{1 + e^{\beta_0+\beta_1 g}}$$

or equivalently

$$\begin{aligned}
\text{logit}\left(\mathbb{P}[\,Y = 1|G = g\,]\right) &= \log\left(\text{Odds}[\,Y = 1|G = g\,]\right) \\
&= \log\left(\frac{\mathbb{P}[\,Y = 1|G = g\,]}{1 - \mathbb{P}[\,Y = 1|G = g\,]}\right) \\
&= \beta_0 + \beta_1 g.
\end{aligned}$$

When adjusting for covariates, we are interested in the β_1 term;

$$\begin{aligned}
\mathbb{P}[\,Y = 1|G, PC_1, PC_2 ...\,] &= \frac{e^{\beta_0+\beta_1 g+\gamma_1 pc_1+\gamma_2 pc_2+...}}{1 + e^{\beta_0+\beta_1 g+\gamma_1 pc_1+\gamma_2 pc_2+...}} \\
\text{logit}(\mathbb{P}[\,Y = 1|G, PC_1, PC_2 ...\,]) &= \beta_0 + \beta_1 g + \gamma_1 pc_1 + \gamma_2 pc_2 + ...
\end{aligned}$$

Binary outcomes

Logistic regression is the standard method for fitting that curve – and testing whether it is flat. Unlike linear regression, it doesn't minimize squared discrepancies, i.e.

$$\sum_{i=1}^{n} (Y_i - \widehat{p}_i)^2$$

but instead minimizes the *deviance*

$$2 \sum_{i=1}^{n} \underbrace{Y_i \log(1/\widehat{p}_i)}_{\text{Large if } Y=1 \text{ and } \widehat{p} \text{ small}} + \underbrace{(1 - Y_i) \log(1/(1 - \widehat{p}_i))}_{\text{Large if } Y=0 \text{ and } \widehat{p} \text{ large}},$$

– which is actually quite similar.

To get tests, we don't calculate $\widehat{\beta}_1$ and its standard error, as in Wald tests. Instead, we calculate the slope of the deviance under the null with $\beta_1 = 0$. This slope is called the **score** and comparing it to zero is called a **score test**.

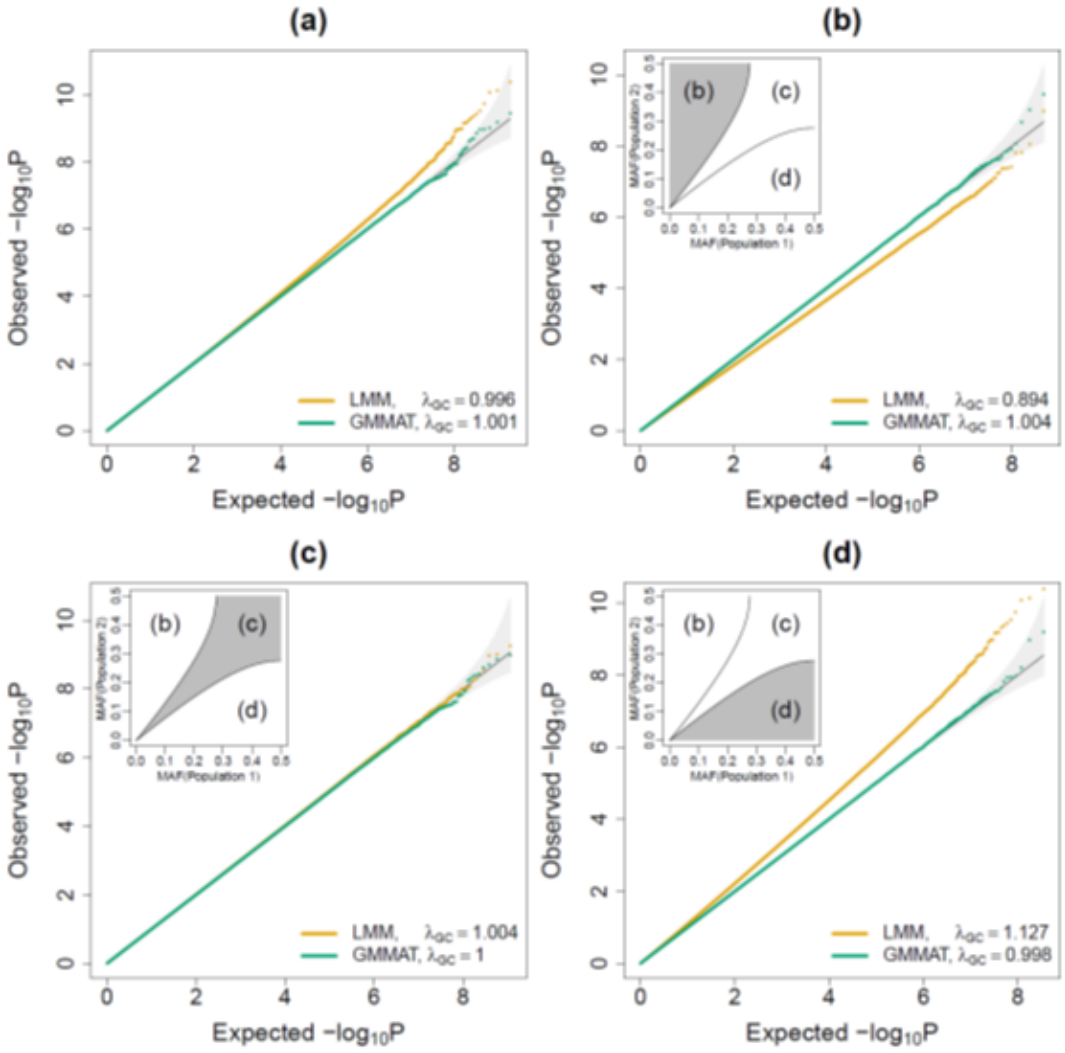
Binary outcomes

Notes on score tests:

- Score tests make allowing variance components for family structure **much** faster. The GMMAT software provides this, and is part of our pipeline
- Score tests only fit the null model, which is also faster than fitting each $\widehat{\beta}_1$
- Score tests produce no $\widehat{\beta}_1$. But if you do need an estimate – say for a 'hit' – just go back and re-fit that variant
- Like Wald tests, score tests rely on approximations; the score test approximations are *generally* at least no worse than the Wald test ones
- For rare outcomes (very *unbalanced* case-control studies) Wald test are terrible; score tests are still not good. For independent outcomes the Firth version of logistic regression is better. Likelihood ratio tests are also worth trying

Binary outcomes

Putting binary outcomes into LMMs (assuming constant variance) tends to work badly – because binary outcomes don't have constant variance, if confounders are present. How badly depends on MAF across studies...

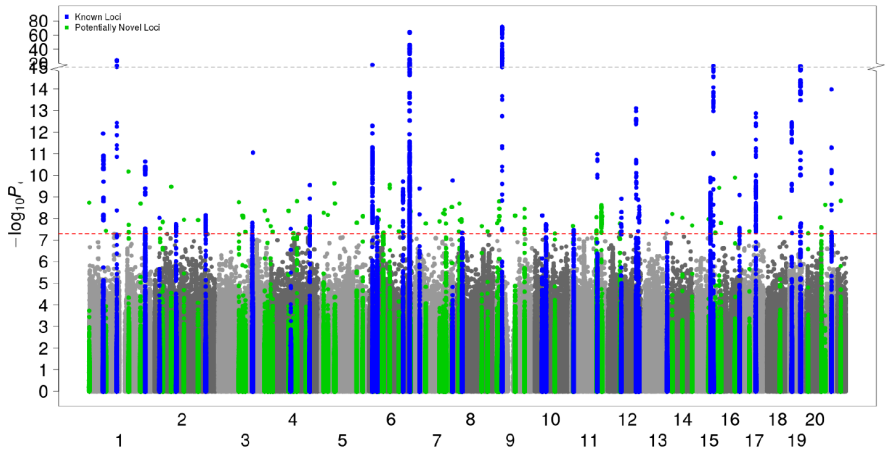


... but please don't do it. For more see the GMMAT paper.

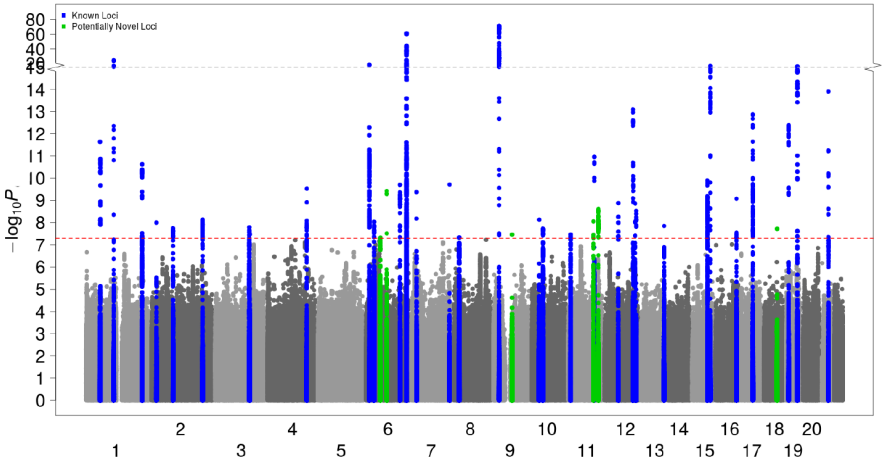
Binary outcomes

Still-more sophisticated *saddlepoint approximations* give better approximations of the reference distribution, used to get the p-values. They are an old method (1954) but new to genetics: (here for CAD in Biobank, $n \approx$400k, 1:12 matching)

SAIGE-NoSPA (GMMAT)



SAIGE



- Zhou et al, in press, Nature Genetics
- Biggest differences are for rarest variants
- SAIGE coming very soon in GENESIS

Part 2: Summary

Single variant analyses in TOPMed...

- Have a heavy testing focus – sensibly – but use tools from linear and logistic regression
- Must account for relatedness
- Should account for study-specific trait variances, where these differ
- Must be fast!
- Do face challenges, particularly for very rare variants and binary outcomes. We may give up interpretability ($\widehat{\beta}_1$'s 'meaning') in order to get more accurate p-values

Pt 3: Multiple Variant Association Tests

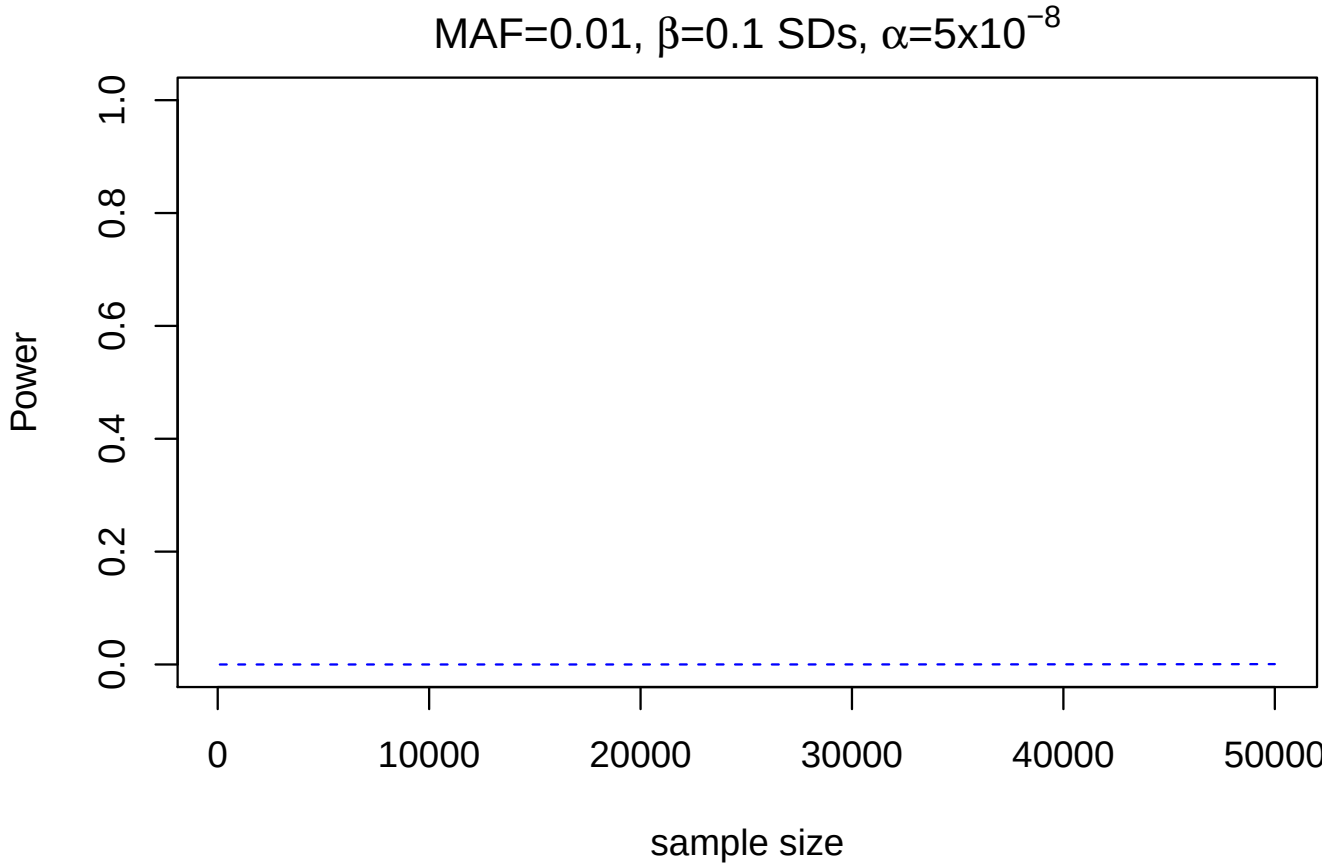
Overview

- Burden tests, as an extension of single-variant methods
- Introduction to SKAT and why it's useful
- Some issues/extensions of SKAT
- Simple calculations for how much annotation helps

– then another hands-on session, with discussion

Why test multiple variants together?

Most GWAS work focused on analysis of single variants that were **common** – MAF of 0.05 and up. For rare variants, power is a major problem;

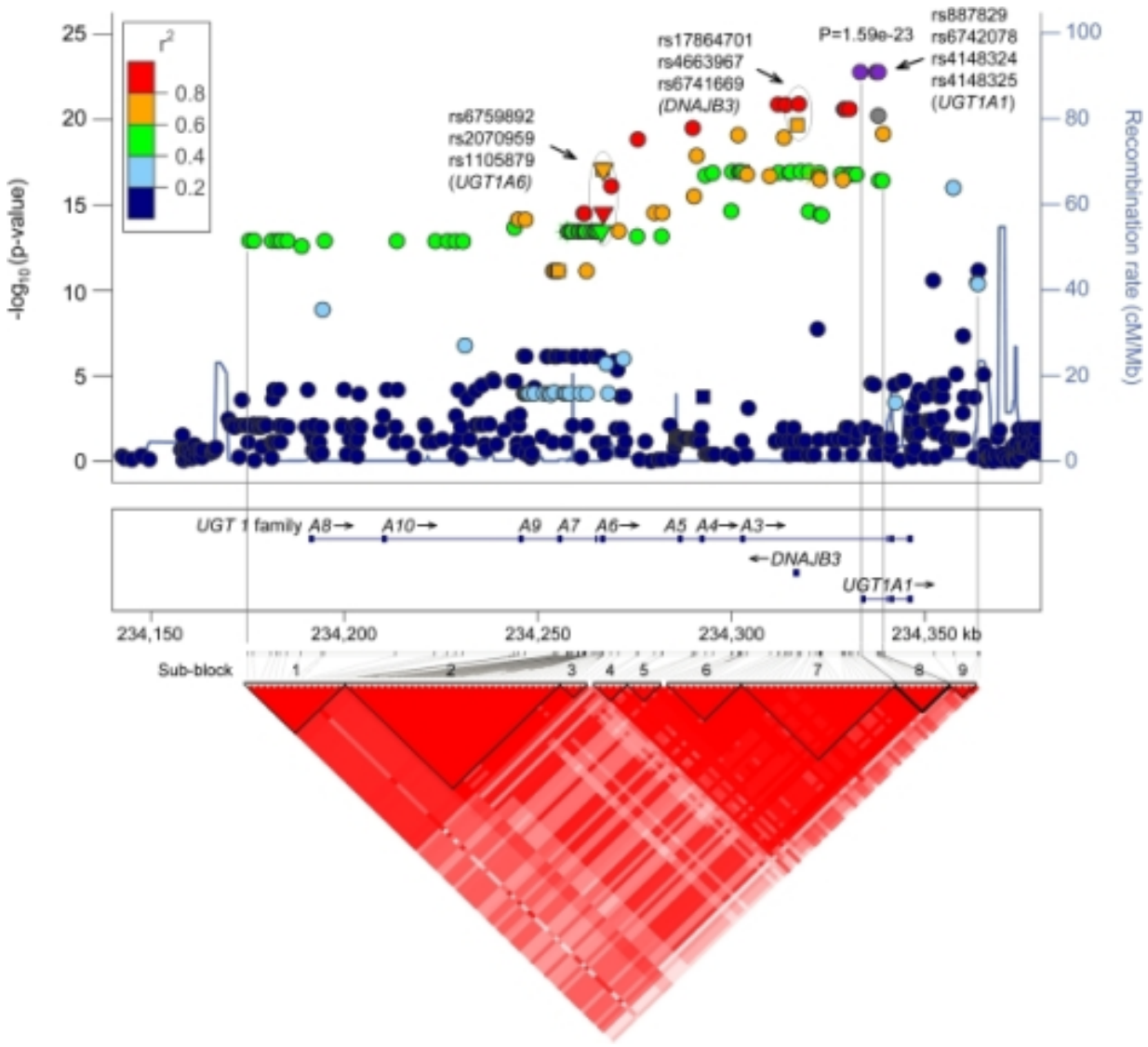


(Even worse for binary outcomes!)

Why test multiple variants together?

Increasing n is (very!) expensive, so instead consider a slightly different question.

Which question? Well...



We care most about the regions – and genes – not exact variants

Burden tests

This motivates *burden* tests; (a.k.a. *collapsing* tests)

1. For SNPs in a given region, construct a 'burden' or 'risk score', from the SNP genotypes for each participant

2. Perform regression of participant outcomes on burdens

3. Regression p-values assess **strong null** hypothesis that all SNP effects are zero

- Two statistical gains: better signal-to-noise for tests and fewer tests to do
- 'Hits' suggest the region contains at least one associated SNP, and give strong hints about involved gene(s)
- ... but don't specify variants, or what their exact association is – which can slow some later 'downstream' work

Burden tests: counting minor alleles

A simple first example; calculate the burden by adding together the number of coded* alleles. So for person i, adding over m SNPs...

$$\text{Burden}_i = \sum_{j=1}^{m} G_{ij}$$

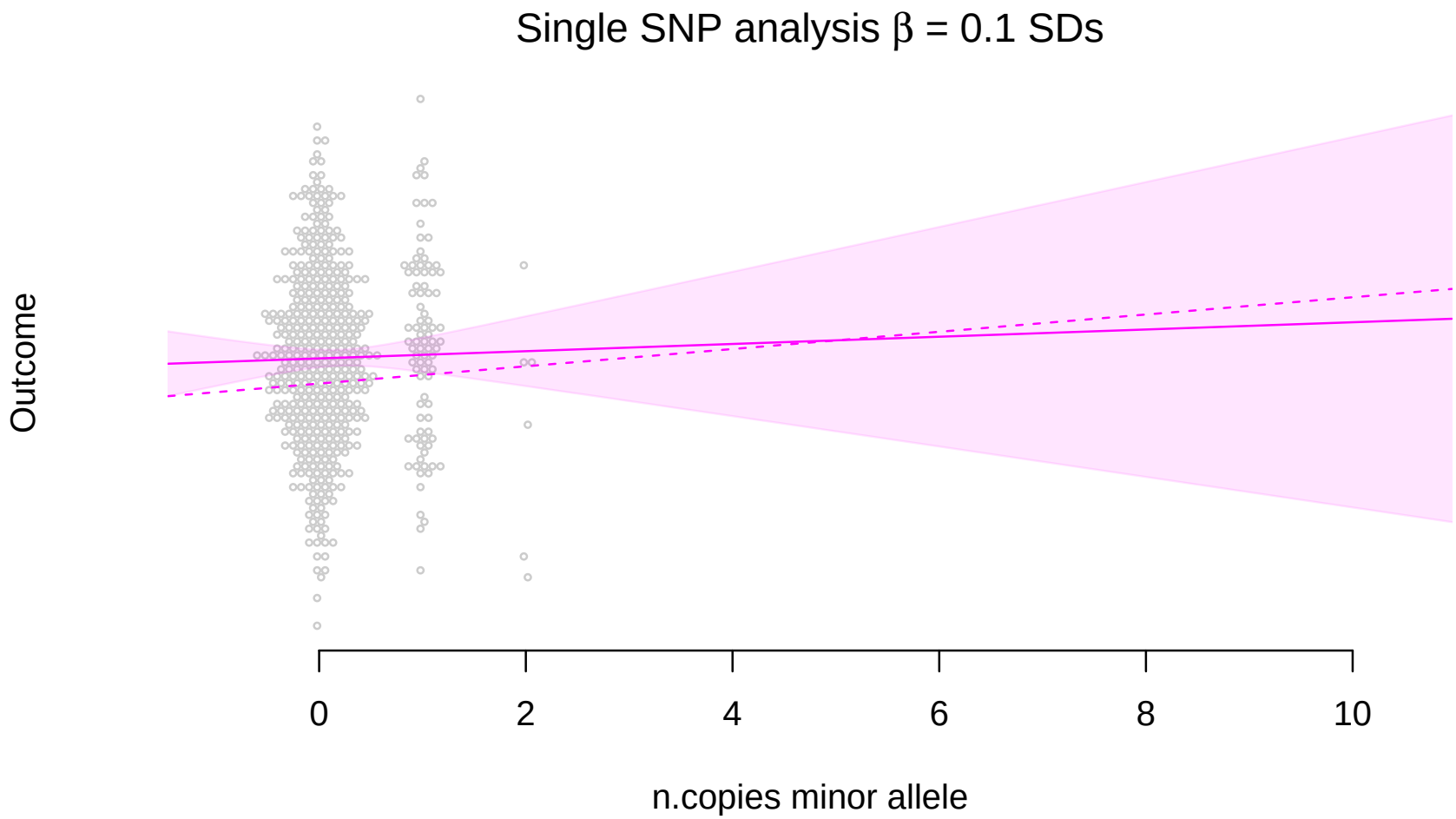
where each G_{ij} can be 0/1/2.

- Burden can take values $0, 1, ..., 2m$
- Regress outcome on Burden in familiar way – adjusting for PCs, allowing for relatedness, etc
- If many rare variants have deleterious effects, this approach makes those with extreme phenotypes 'stand out' more than in single SNP work

* ...usually the same as minor allele

Burden tests: counting minor alleles

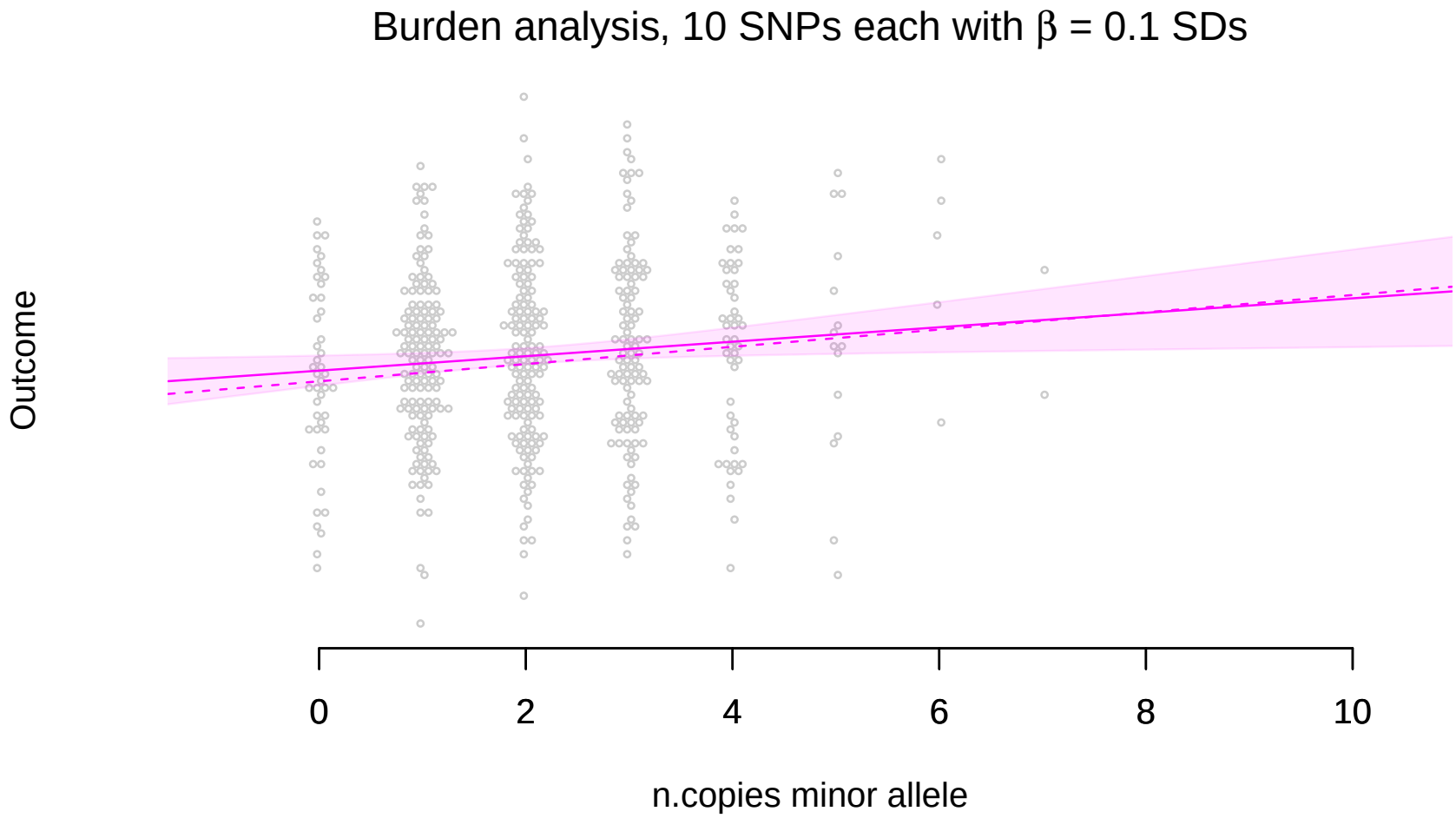
Why does it help? First a single SNP analysis; ($n = 500$, and we know the true effect is $\beta = 0.1$ per copy of the minor allele)



Few copies of the variant allele, so low precision; $p = 0.7$

Burden tests: counting minor alleles

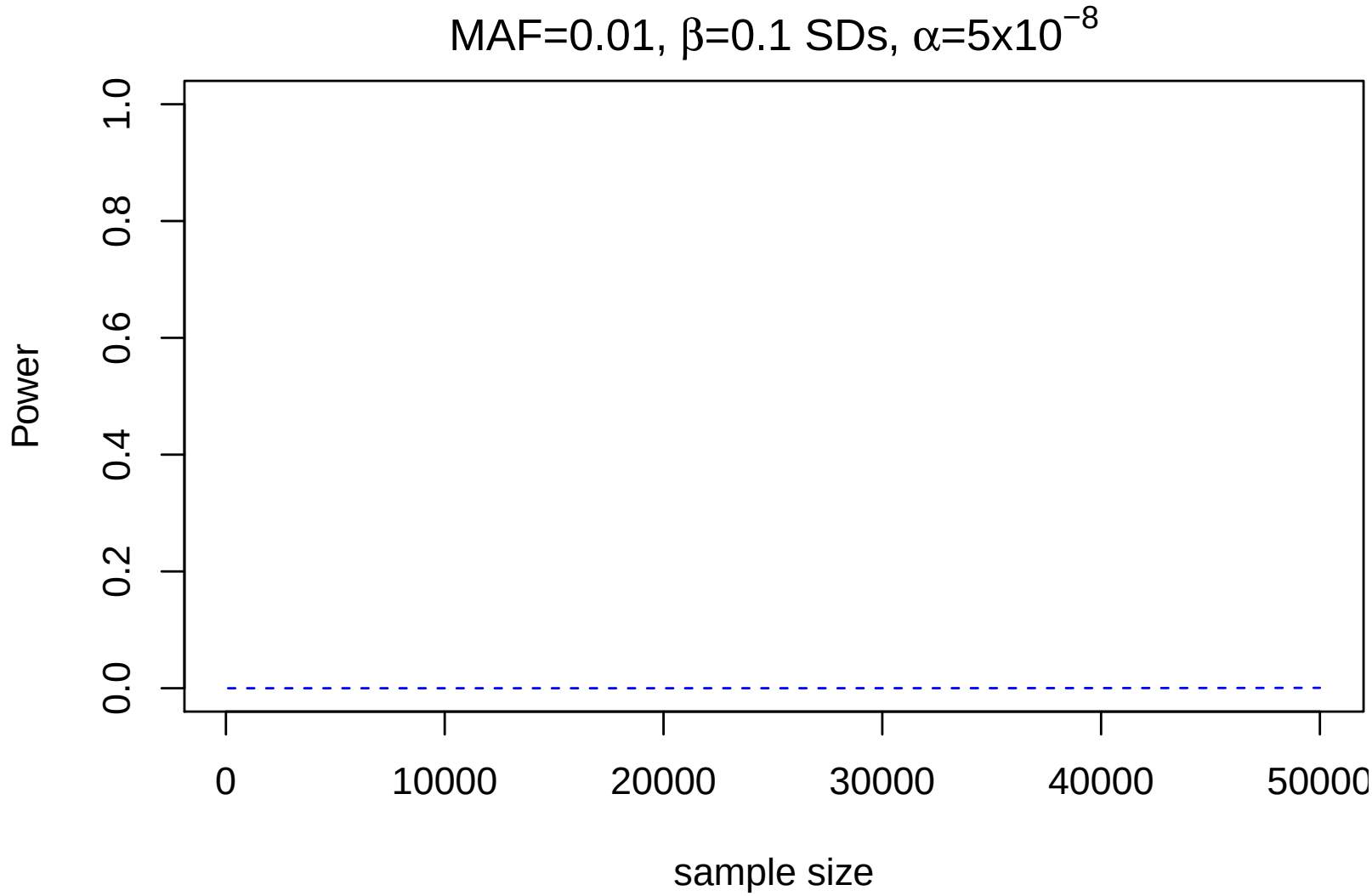
With same outcomes, analysis using a burden of 10 SNPs each with the same effect.



The signal size is identical – but with less noise get $p = 0.02$

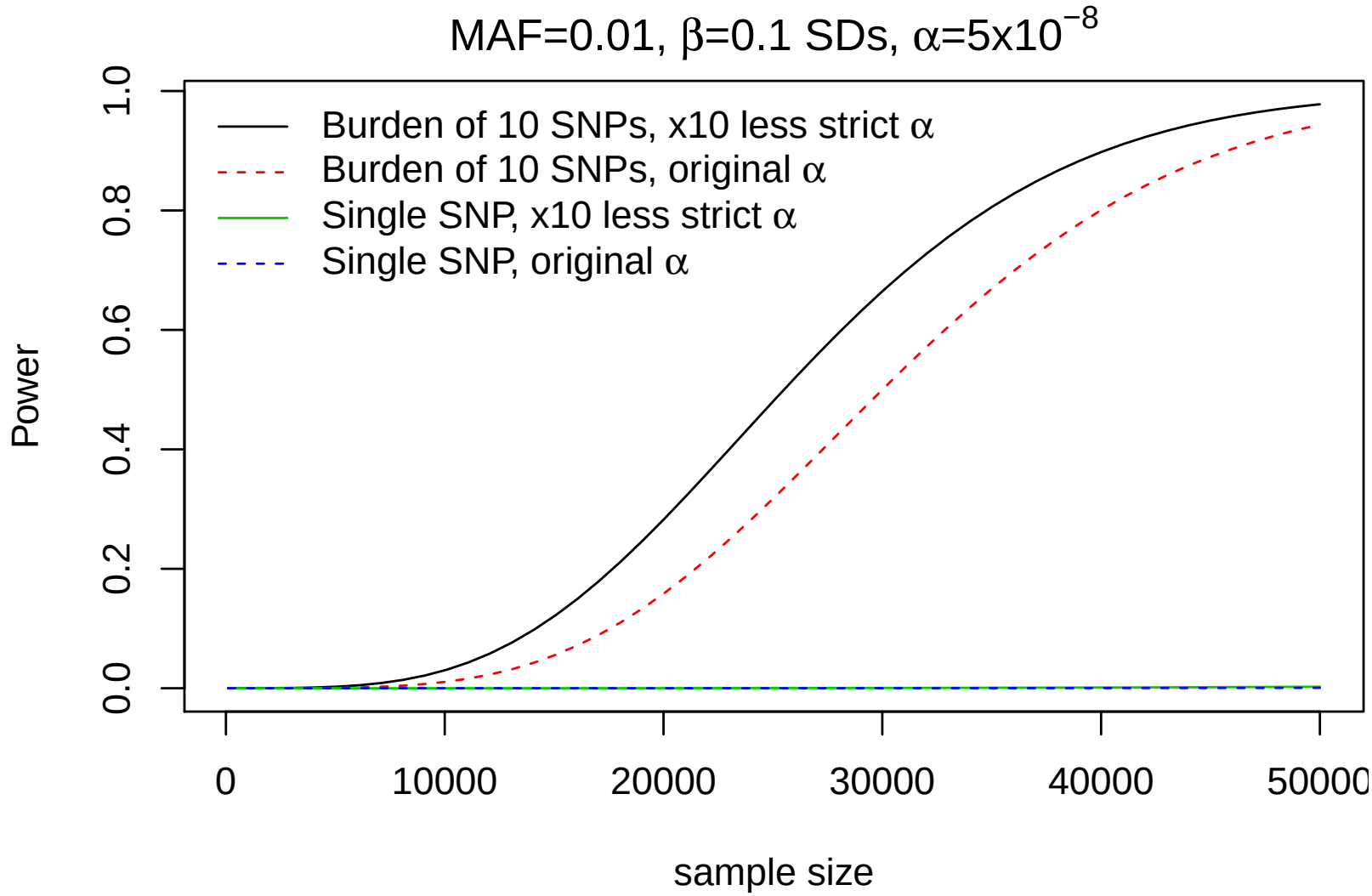
Burden tests: does it help power?

Back to our miserable single SNP power;



Burden tests: does it help power?

What happens with x10 less noise? x10 fewer tests?



Burden tests: does it help power?

The signal size β is unchanged. The extra power is **almost all** due to reduced noise in $\widehat{\beta}$, not fewer tests. To see this it may help to look at the sample size formula;

$$n_{\text{needed}} \geq \frac{(Z_{1-\alpha} + Z_{\text{Power}})^2}{\beta^2 \text{ Var(genotype)}}$$

- Using a burden increases the spread (variance) of the 'genotypes' by a factor of m, compared to 0/1/2 single SNPs
- Increasing α from 5×10^{-8} to 5×10^{-7} changes $Z_{1-\alpha}$ from ≈ 5.5 to 5.0 – which is helpful, but not **as** helpful

However, the assumption that all m SNPs have the same effect is implausible, even as an approximation. Power to find an 'average' effect is often low – when we average effects in both directions, and/or with many effects $\approx$ zero.

Burden tests: other burdens

Many other burdens are available; all are 'uni-directional'

- Cohort Allelic Sum Test (CAST) uses

$$\text{Burden}_i = \begin{cases} 1 & \text{if any rare variant present} \\ 0 & \text{otherwise} \end{cases}$$

  ... much like the 'dominant model' of inheritance
- Risk scores;

$$\text{Burden}_i = \sum_{j=1}^{m} \widehat{\beta}_j G_{ij}$$

  ... where, ideally, the $\widehat{\beta}_j$ come from external studies. (Can estimate them from data – see Lin & Tang 2011 – but getting p-values is non-trivial)
- Which variants to include at all can depend on MAF, putative function, and other factors – see annotation material

Which to choose? The one that best separates high/low phenotypes, at true signals – choosing is not easy.

Burden tests: other burdens

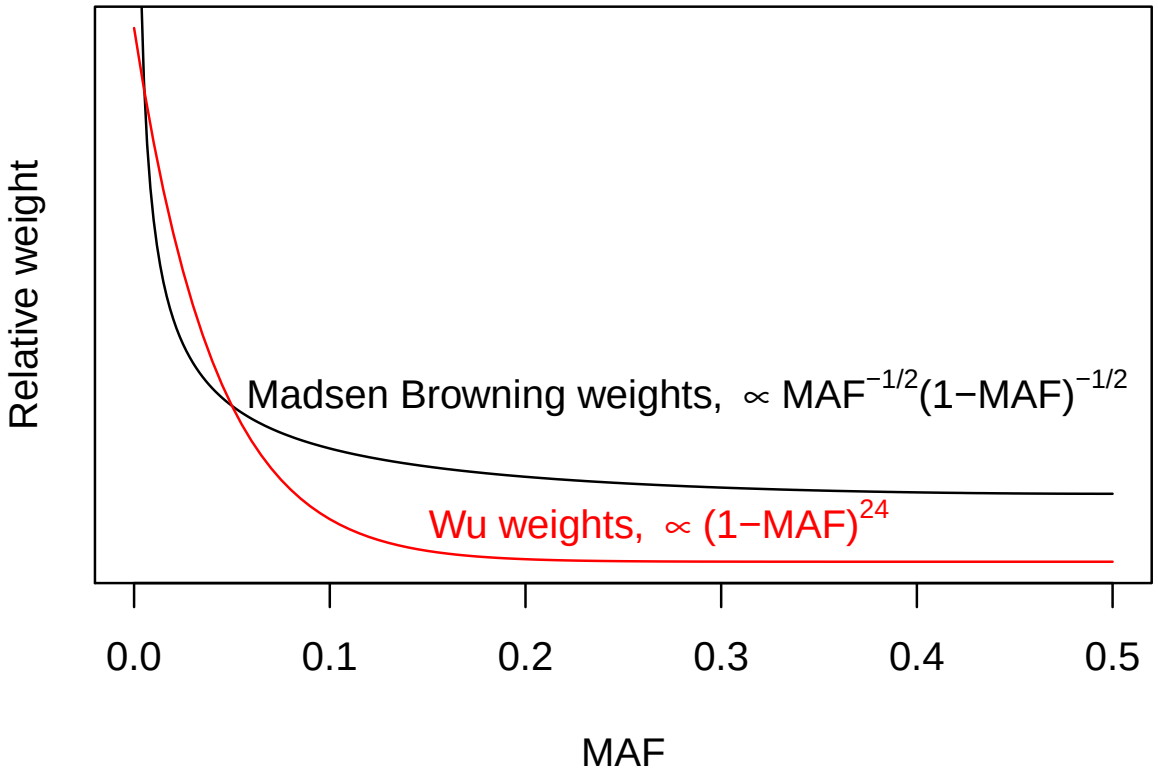
Weighting also generalizes the available burdens;

$$\text{Burden}_i = \sum_{j=1}^{m} w_j G_{ij}.$$

– genotype at variants with higher w_j affect total burden more strongly than with lower w_j.

Two simple & popular MAF-based weights; (right)

(Only *relative* weight matters)



SKAT

...and the Sequence Kernel Association Test

SKAT: recall score tests

We met score tests in Section 2. For linear or logistic regression, the score statistic for a single SNP analysis can be written;

$$\sum_{i=1}^{n} (Y_i - \widehat{Y}_i)(G_i - \bar{G})$$

- ... where $\widehat{Y}_i$ comes from fitting a null model, with no genetic variants of interest
- Values of the score further from zero indicate stronger association of outcome Y with genotype G
- Reference distribution for this test statistic involves variance of outcomes $\mathbf{Y}$, which may allow for relatedness

SKAT: scores^2 for many variants

With m SNPs, to combine the scores without the averaging-to-zero problem, the SKAT test squares them and adds them up. In other words, its test statistic is

$$Q = \sum_{i=1}^{n} \sum_{j=1}^{m} (Y_i - \hat{Y}_i)^2 G_{ij}^2 w_j^2$$

with variant-specific weights w_j, as in burden tests.

In math-speak this is

$$Q = (\mathbf{Y} - \hat{\mathbf{Y}})^T \mathbf{GWWG}^T (\mathbf{Y} - \hat{\mathbf{Y}})$$

where **G** is a matrix of genotypes and **W** a diagonal matrix of variant-specific weights.

Standard results about the distribution of many squared, added-up Normal distributions can be used to give a reference distribution for Q...

SKAT: scores^2 for many variants

It turns out the approximate distribution of Q is under usual assumptions (e.g. constant variance);

$$Q \sim \sum_{k=1}^{\min(m,n)} \lambda_k \chi_1^2$$

i.e. a weighted convolution of many independent χ_1^2 distributions, each weighted by λ_k. (The λ_k are eigenvalues of a matrix involving $\mathbf{G}, \mathbf{W}$, confounders, and variance of $\mathbf{Y}$ under a fitted null model.)

This approximate distribution can be calculated approximately (!) – and we get p-values as its tail areas, beyond Q.

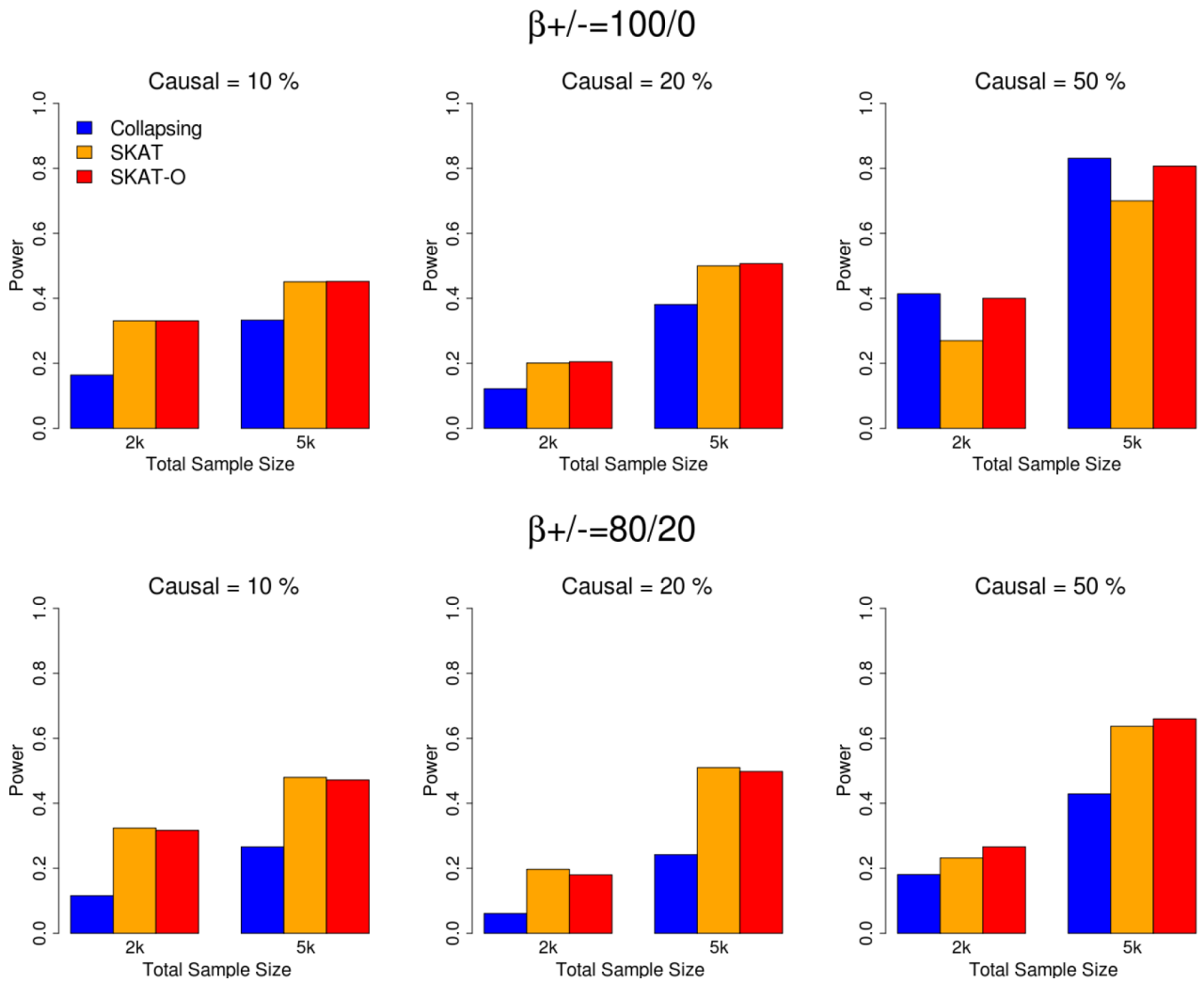
SKAT has quickly become a standard tool; see hands-on session for examples/implementation.

Comparing SKAT and Burden tests

- Burden tests are more powerful when a large percentage of variants are associated, with similar-sized effects in the same direction
- SKAT is more powerful when a small perentage of variants are associated and/or the effects have mixed directions
- Both scenarios can happen across the genome – best test to use depends on the underlying biology, if you have to pick one
- With tiny α a factor of 2 in multiple testing is not a bit deal – so generally okay to run both
- If you can't pick, can let the data help you: SKAT-O does this (for details see Lee et al 2012)

Comparing SKAT and Burden tests

Examples of power comparisons (for details see Lee et al 2012)



SKAT: other notes

- As with burden tests, how to group and weight variants can make a big difference – see annotation sessions
- Allowing for family structure in SKAT is straightforward; put kinship, non-constant variance etc into the variance of $\mathbf{Y}$.
- For heavy-tailed traits, inverse-Normal transforms will help – a lot!
- Various small-sample corrections are also available – but may not be quick. In particular for binary traits see Lee, Emond et al – coming soon to GENESIS
- When both m and n get large, constructing the reference distribution takes a lot of memory. But it can be (well!) approximated by using random methods; see fastSKAT – available now, brought into next version of GENESIS

How does annotation help?

Annotation-based filters are popular for WGS work. They can help by:

- Reducing the multiple testing problem for single variant work
- Concentrating on the true signals, for e.g. SKAT

Earlier we saw how simple burden tests can improve power over single-variant work – but **not**, primarily, by reducing the number of tests. This is non-intuitive – so we provide some "back of envelope" calculations about how power is affected, to help understand the impact of good/bad filters.

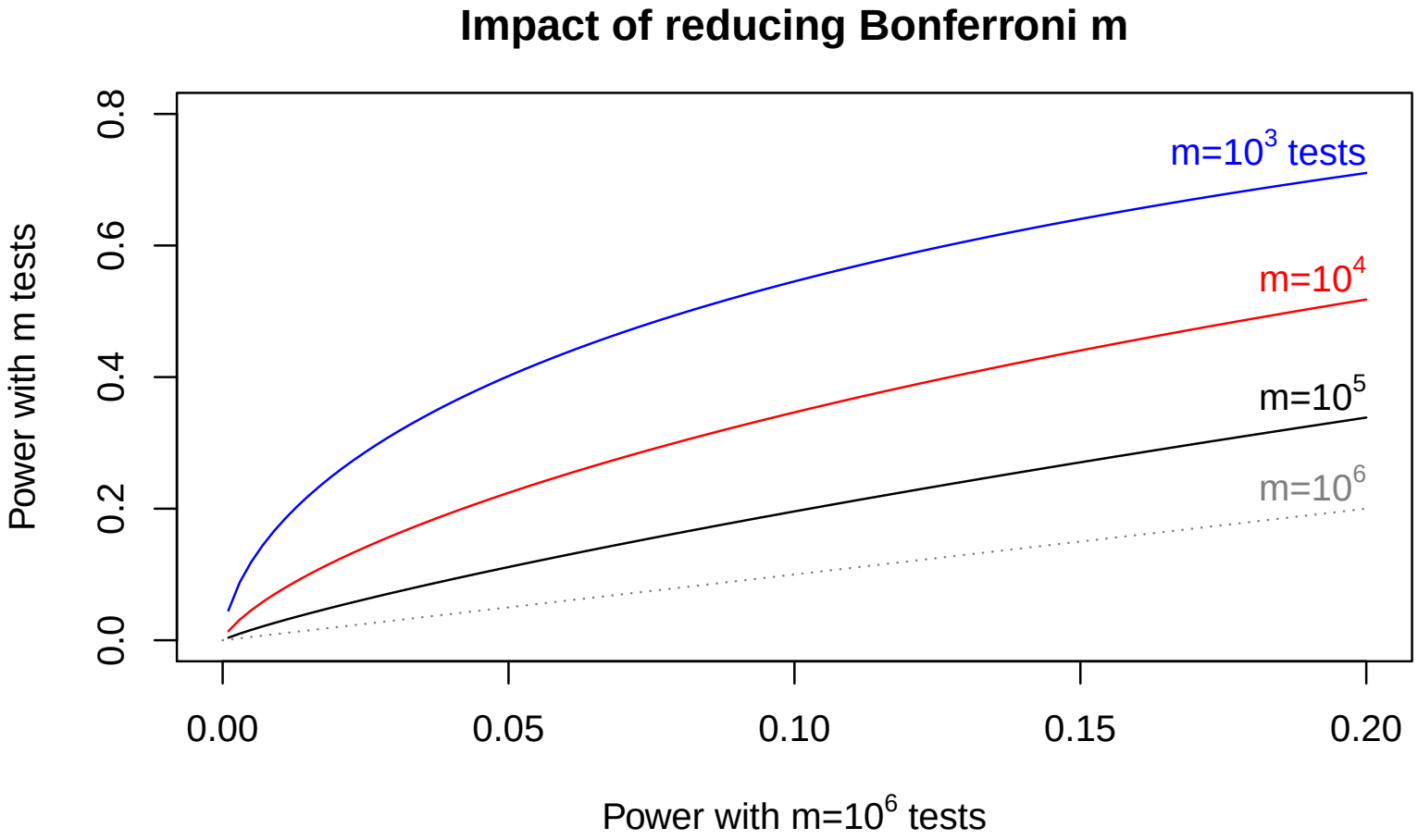
How does annotation help? Single var'ts

For these, if we know/assume **both**:

- Power for a single variant analysis with a true signal
- 'Unfiltered' significance threshold, i.e. $\alpha/m = 0.05/10^6 = 5 \times 10^{-8}$

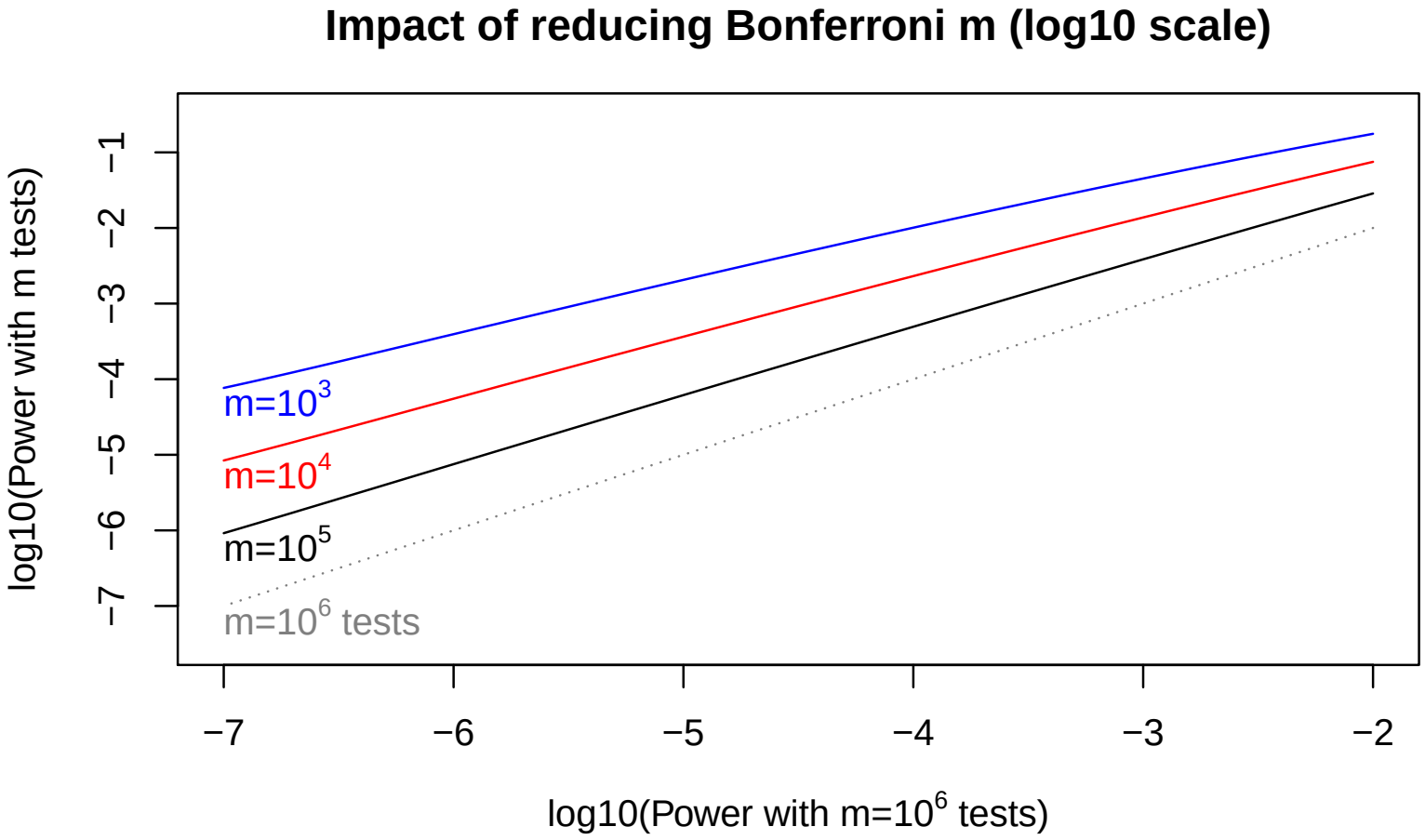
...then we can obtain power for that signal, at the lower value of m that filtering provides.

How does annotation help? Single var'ts



Need *at least* an order of magnitude smaller m to move from 'hopeless' to 'hopeful'.

How does annotation help? Single var'ts



For each $\times 10$ reduction in m, get between $\times 2$ and $\times 10$ increase in power. Need several of these to move from 'hopeless' to 'hopeful'.

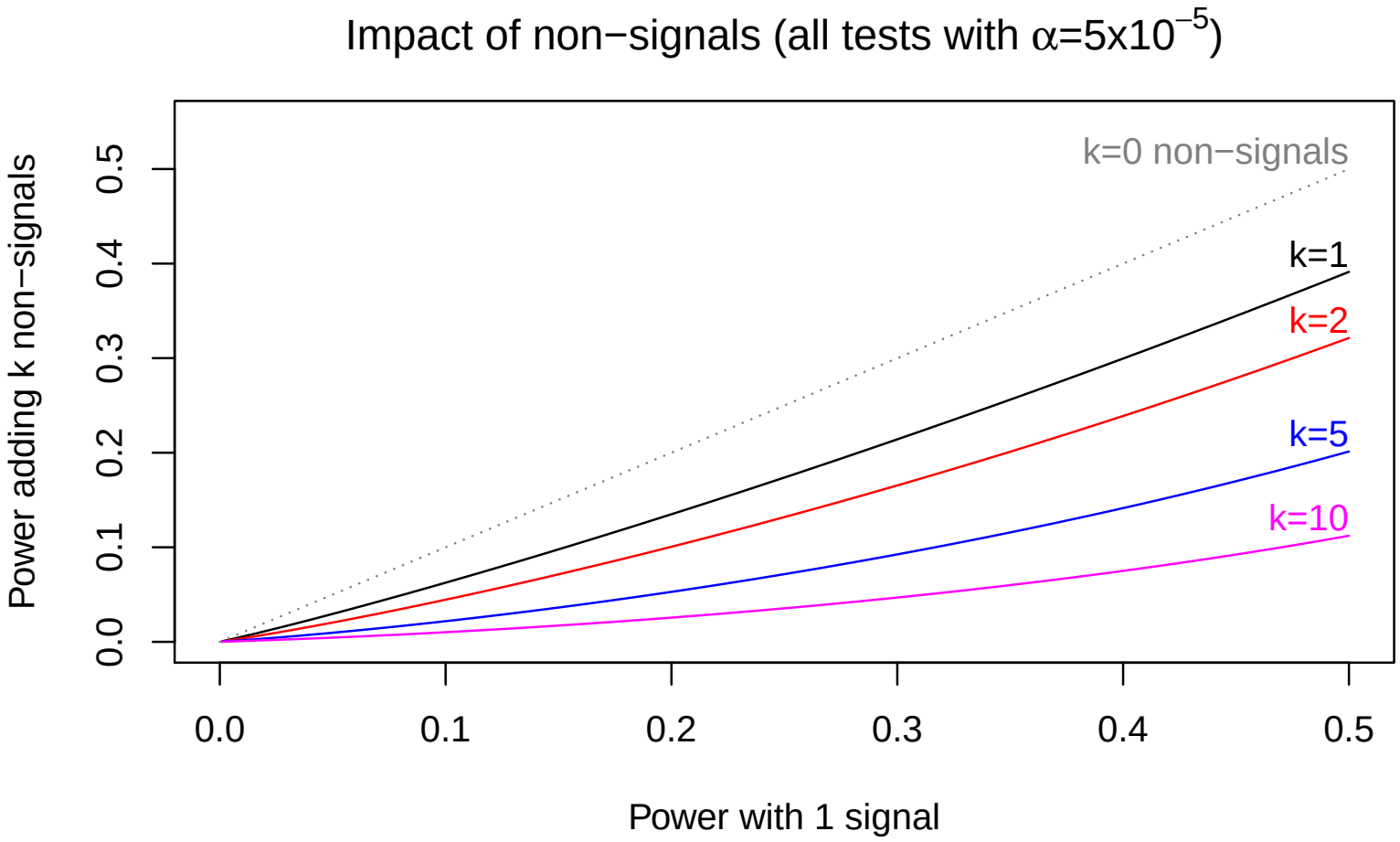
How does annotation help? SKAT

There are more 'moving parts' in SKAT, so here assume

- Testing 20,000 regions, using $\alpha = 5 \times 10^{-5}$
- Have one true (effective) signal in a region
- Any weights used do not up/down weight signal/non-signal variants and yes this is debatable

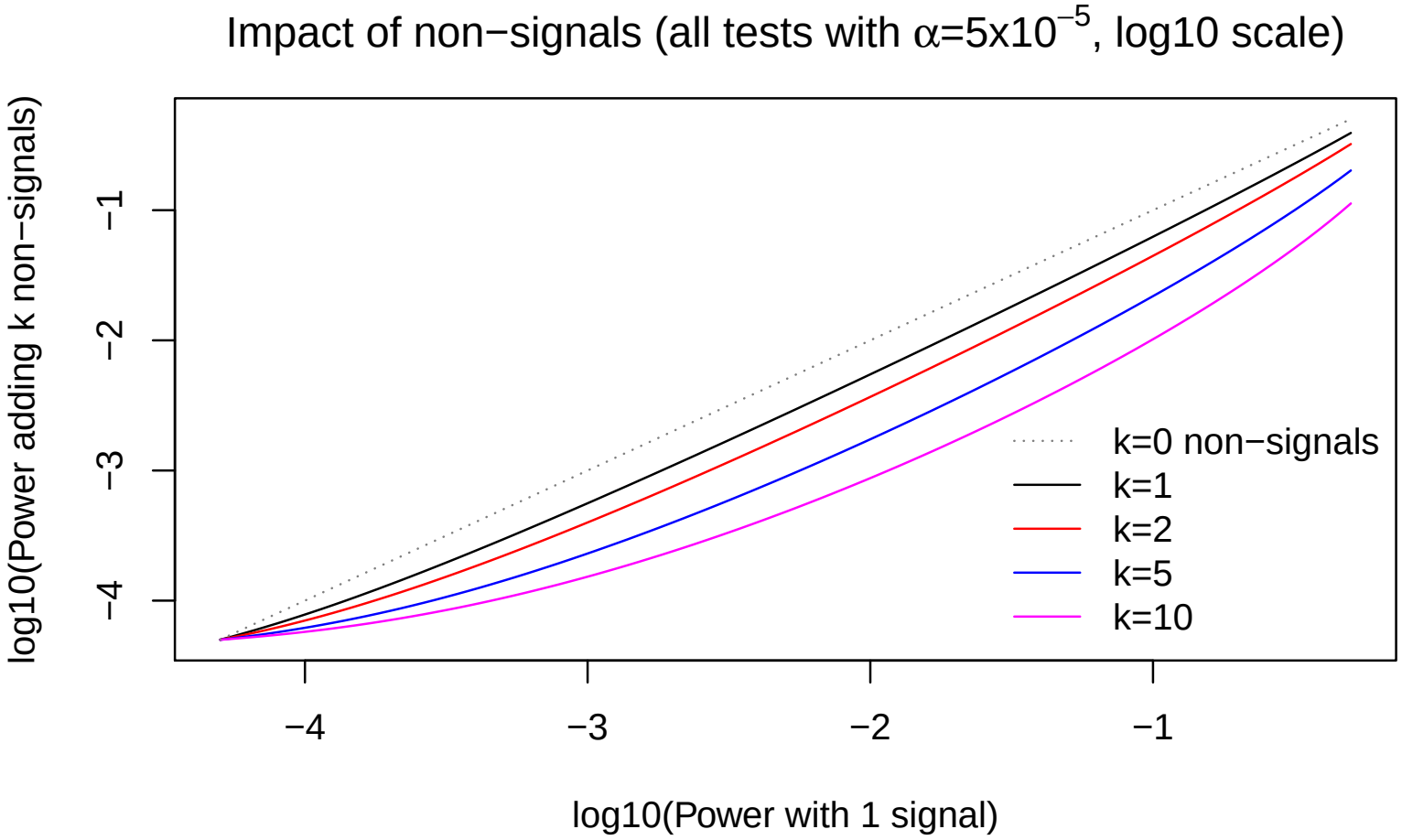
With these, we can obtain the power if k independent **non**-signals are added to a region where the true signal has a specified power.

How does annotation help? SKAT



Can also view this as effect of filtering out non-signals - so $k = 10$ shows impact filtering out noise, when 9/10 variants are just noise.

How does annotation help? SKAT



Reducing k by $\times 10$ provides up to $\times 10$ more power, but often less that that.

How does annotation help? SKAT

Summary of these calculations;

> Filtering needs to be **very aggressive** and **correct** to move analyses from hopeless to hopeful.

- 'Correct' means 'actually selecting the SNPs where there are true signals'
- Milder (non-aggressive) filtering doesn't hurt
- Incorrect filtering (omitting SNPs with true signals) **does hurt**
- Situations with multiple small signals in a region may behave somewhat better - not studied here

Part 3: Summary

- In WGS, power for single variant analyses can be very limited
- Multiple-variant tests enable us to answer region-specific questions of interest, not variant-specific – and power is better for them
- Burden/collapsing tests are conceptually simple, but only provide best power against very specific alternatives
- SKAT (and relatives) give better power for many *other* alternatives
- With very small signals, neither approach is guaranteed to give actually-good power. Accuracy of approximate p-values can also be an issues. Treat results with skepticism, and try to replicate

Part 4: Faster/Cheaper ways to compute

Motivation

The methods we've discussed are (good) defaults for WGS data. But...

- Fitting the null model, as currently implemented, now takes considerable computing resources
- For subsequent releases, we expect that the computational burden will become all but intolerable
- Need to fit null model faster and/or using less memory
- Actually running tests is also not cheap!

Chaoyu Yu (Biost PhD student) has been working on these issues, as an RA.

Jen Brody (and others) have trialed some 'fixes' on the Analysis Commons.

Why is this hard?

For outcomes $\mathbf{Y}$, covariates $\mathbf{X}$, fixed effects $\boldsymbol{\beta}$ and covariance matrix $\mathbf{V}$ we assume

$$\underbrace{\mathbf{Y}}_{n\times 1} \sim N\left(\underbrace{\mathbf{X}}_{n\times p}\underbrace{\boldsymbol{\beta}}_{p\times 1}, \underbrace{\mathbf{V}}_{n\times n}\right),$$

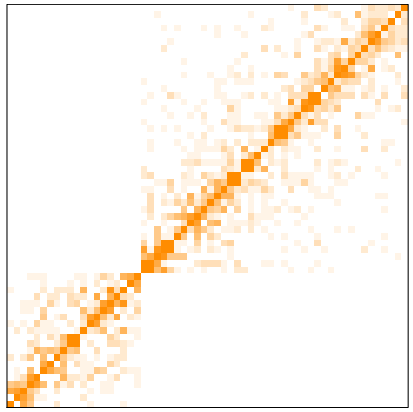
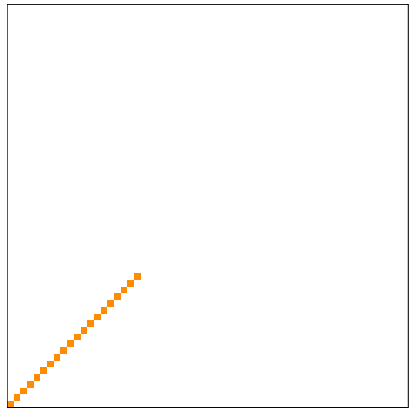
where $\mathbf{V}$ is the sum of several *variance components*.

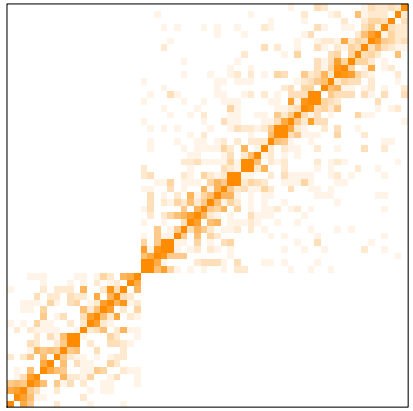
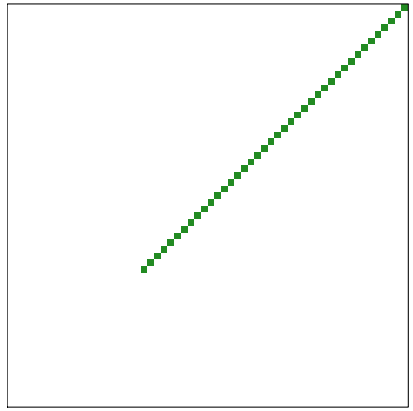
- In TOPMed's use of LMMs, we fit a *null model* – with no single SNPs in $\mathbf{X}$, say (although PCs may be present)
- The fitted null model's estimates $\hat{\boldsymbol{\beta}}$ and $\hat{\mathbf{V}}$ are then used to perform tests, for individual variants. In this second step, no extra fitting takes place

(Similar results hold for GLMMs, so are omitted here)

Why is this hard?

With two groups, these might be;

$\mathbf{V} \mid \text{in group } A =$  $+$ 

$\mathbf{V} \mid \text{in group } B =$  $+$ 

so we can write $\mathbf{V} = \sigma^2_G \times$ kinship (relatedness) $+ \sigma^2_{\text{group}}$ (noise)

... and we need to estimates $\hat{\sigma}^2_G$, $\hat{\sigma}^2_A$ and $\hat{\sigma}^2_B$.

Why is this hard?

The log-likelihood for the model specifies

$$l(\boldsymbol{\beta}, \mathbf{V}) = -\frac{n}{2} \log |\mathbf{V}| - \frac{1}{2}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T \mathbf{V}^{-1} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + p \log(2\pi)$$

Optimizing this with respect to $\boldsymbol{\beta}$, for a fixed $\mathbf{V}$ matrix, is conceptually simple; we want to optimize a weighted version of sum of square residuals, where the weight given to each pair of observations i, j is given by the i, j element of $\mathbf{V}^{-1}$.

To do the optimization for $\boldsymbol{\beta}$, we have to minimize a quadratic. This is easy to write; the resulting estimate is

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}\mathbf{V}^{-1}\mathbf{X})^T \mathbf{X}^T \mathbf{V}^{-1} \mathbf{Y},$$

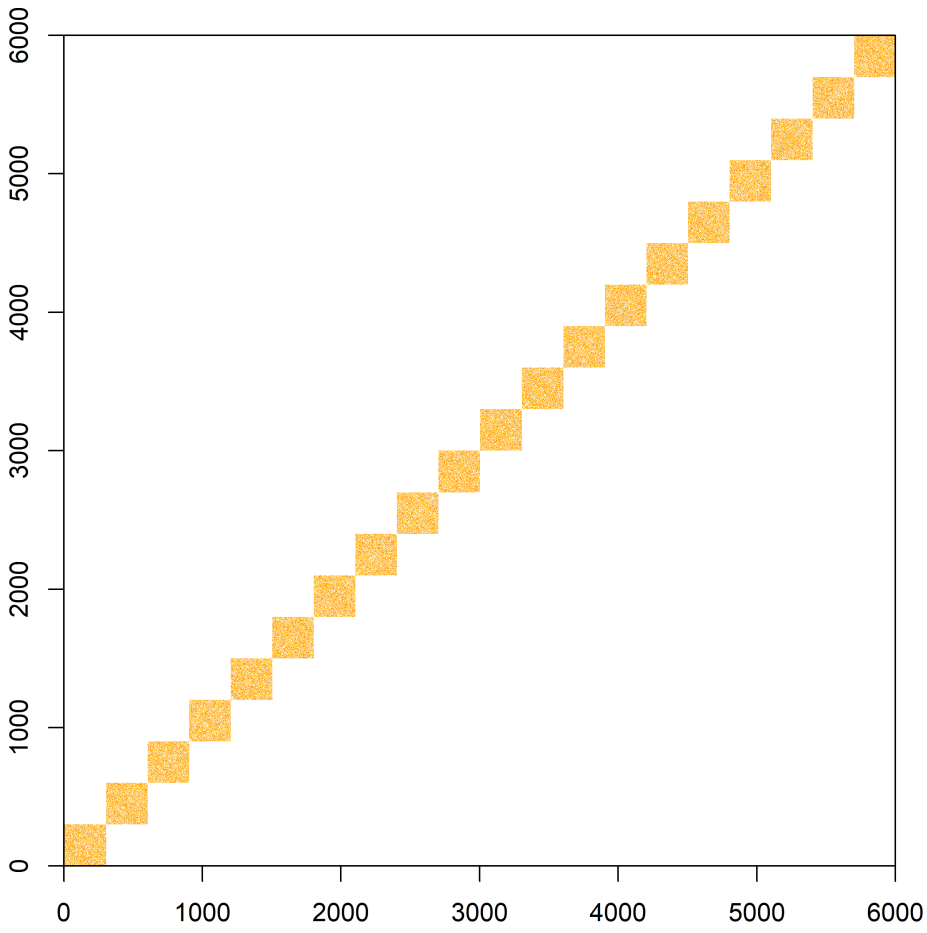
but calculating $\mathbf{V}^{-1}$ is slow, taking of the order of n^3 steps for general $\mathbf{V}$.

A fix: exploiting sparsity

Just **storing** general **V**, or its inverse, uses substantial memory; for $n = 6000$ it takes 274 Mb. For $n = 50,000$ this would be 18.6 Gb – it scales with n^2

But typically our variance matrices are roughly block-diagonal (kinship plus some diagonal elements).

For $n = 6000$ in 30 blocks of 200, kinship might look something like this;



A fix: exploiting sparsity

R's `Matrix` package provides matrix utilities (addition, multiplication, inversion, etc) that can exploit this *sparsity*.

- Storage is **much** cheaper; 20.7 Mb for $n = 6000$ in 30 blocks of 200 – and for $n = 50,000$ in blocks of 200, this scales linearly – so 20.7 Mb $\times$ 50/6 = 172.5 Mb
- Matrix arithmetic steps for block-diagonal matrices are also **much** faster – they grow with $n_{\text{largest block}}$, not overall n. This makes a big difference; 1000^3 is smaller than $50,000^3$ by a factor of 125,000.

In practice, this step **solves** the difficulty of $\mathbf{V}$ being $n \times n$, and slow to invert – provided we are willing to use sparse relatedness matrices.

MMAP uses the same idea, known as "adding over the pedigrees"

A fix: exploiting sparsity in practice

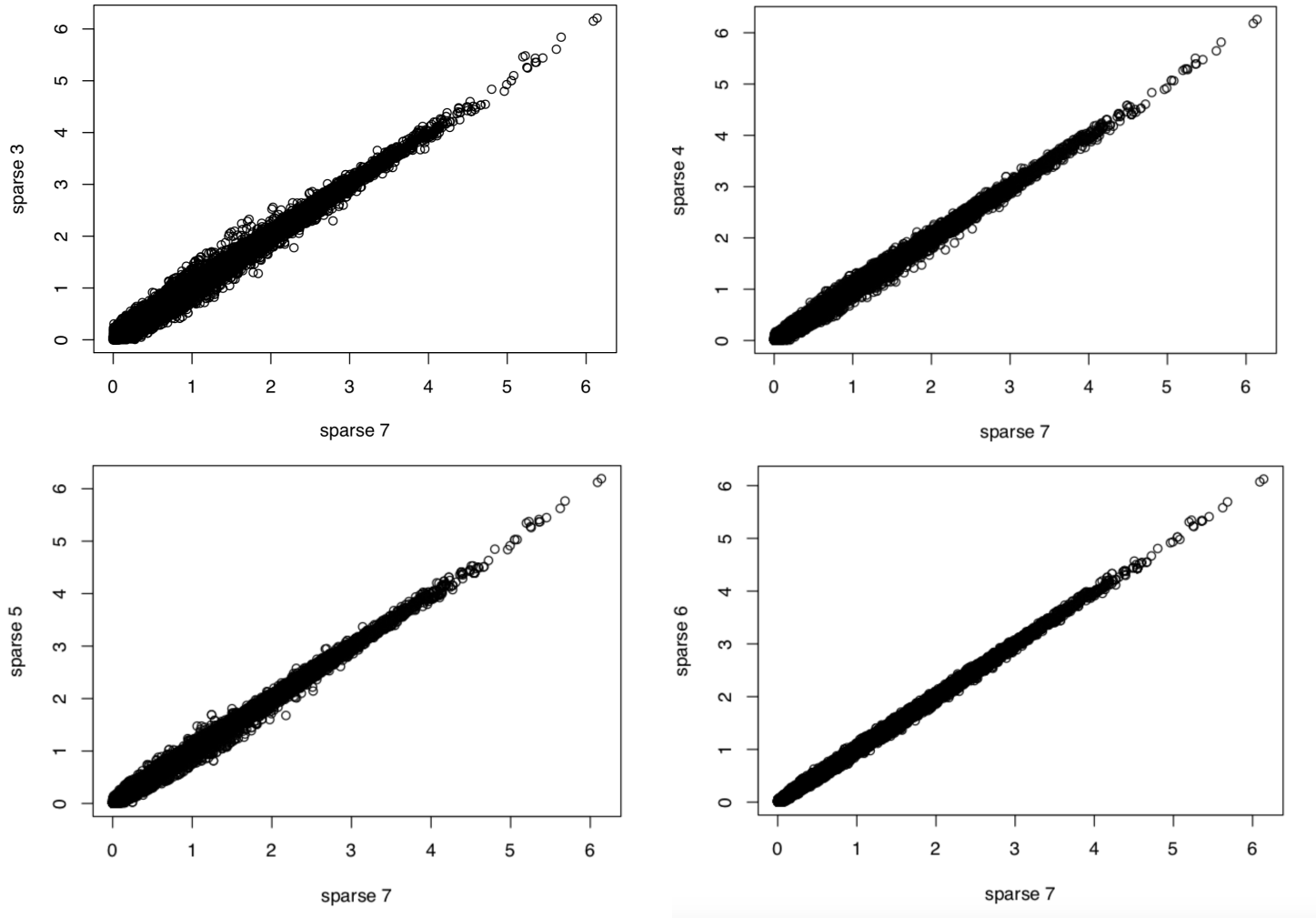
Taking this approach, we have to determine a threshold for 'unrelated' – when do we set kinships equal to zero, so we can have a block-diagonal $\mathbf{V}$?

Empirical results from analysis of $n = 25077$ TOPMed participants, setting 'unrelated' to be 'at least nth-degree relative'

Kinship	Sparsity degree	3	4	5	6	7
	Max block size	794	1007	1836	7717	20783
	# of blocks	19592	18503	17289	14219	4173
Null Model	Time (s)	67	91	156	7457	187266
	Max Memory (GB)	16	15	18	15	37
Test	Time (s)	7676	8189	8505	12891	43406
	Max Mem (GB)	2	2	2	3	8

A fix: exploiting sparsity in practice

And how do the results compare? Comparing -log10(p) from the blocks obtained zeroing out below 7th-degree:



A 2nd fix: hand-coding projections

For the σ^2 estimates, we use *restricted maximum likelihood* (REML) which estimates them using only the parts of $\mathbf{Y}$ orthogonal to (unaffected by) the fixed effects.

- There is no closed-form solution for the $\hat{\sigma}^2$ estimates; we have to solve equations iteratively
- The equations feature many uses of this matrix;

$$\mathbf{P} = \mathbf{V}^{-1} - \mathbf{V}^{-1}\mathbf{X}(\mathbf{X}^T\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{V}^{-1}$$

  ... and $n \times n$, non-sparse matrix. Our current code calculates it and stores it explicitly
- But we don't need to. For example

$$\underbrace{\mathbf{P}\mathbf{Y}}_{n\times 1} = \underbrace{\mathbf{V}^{-1}\mathbf{Y}}_{n\times 1} - \underbrace{\mathbf{V}^{-1}\mathbf{X}}_{n\times p}\underbrace{(\mathbf{X}^T\mathbf{V}^{-1}\mathbf{X})^{-1}}_{p\times p}\underbrace{\mathbf{X}^T\mathbf{V}^{-1}\mathbf{Y}}_{p\times 1}$$

  ... and the elements here use sparse $\mathbf{V}^{-1}$ that we're storing anyway. We get exactly the same arithmetic, using much less memory. (Keen people: for matrix traces we use other tricks)

A 2nd fix: hand-coding in practice

For the $n = 6000$ example, with 20 blocks of 300 people – and fitting non-constant variance for two groups (15 blocks:5 blocks) using data that takes 7 iterations to fit the null;

- $\sim$740 seconds (12m 20s) with the current code
- 11 seconds with the new version

Scaling these linearly ($\times 50/6 \approx 8.33$) fitting the null should present no major challenge.

Code is now available.

Related: fastSKAT

We discussed and used SKAT in Part 3. For m variants and n steps, the compute time to implement a SKAT test grows with $m \times n \times \min(m, n)$. For 100k people and 100k markers, this is a problem.

- Computing the test statistic is not the bottleneck, instead we must compute a specific convolution:

  $$\lambda_1\chi_1^2 + \lambda_2\chi_1^2 + \lambda_3\chi_1^2 + \lambda_4\chi_1^2 + \lambda_5\chi_1^2 + \ldots + \lambda_{\min(m,n)}\chi_1^2$$

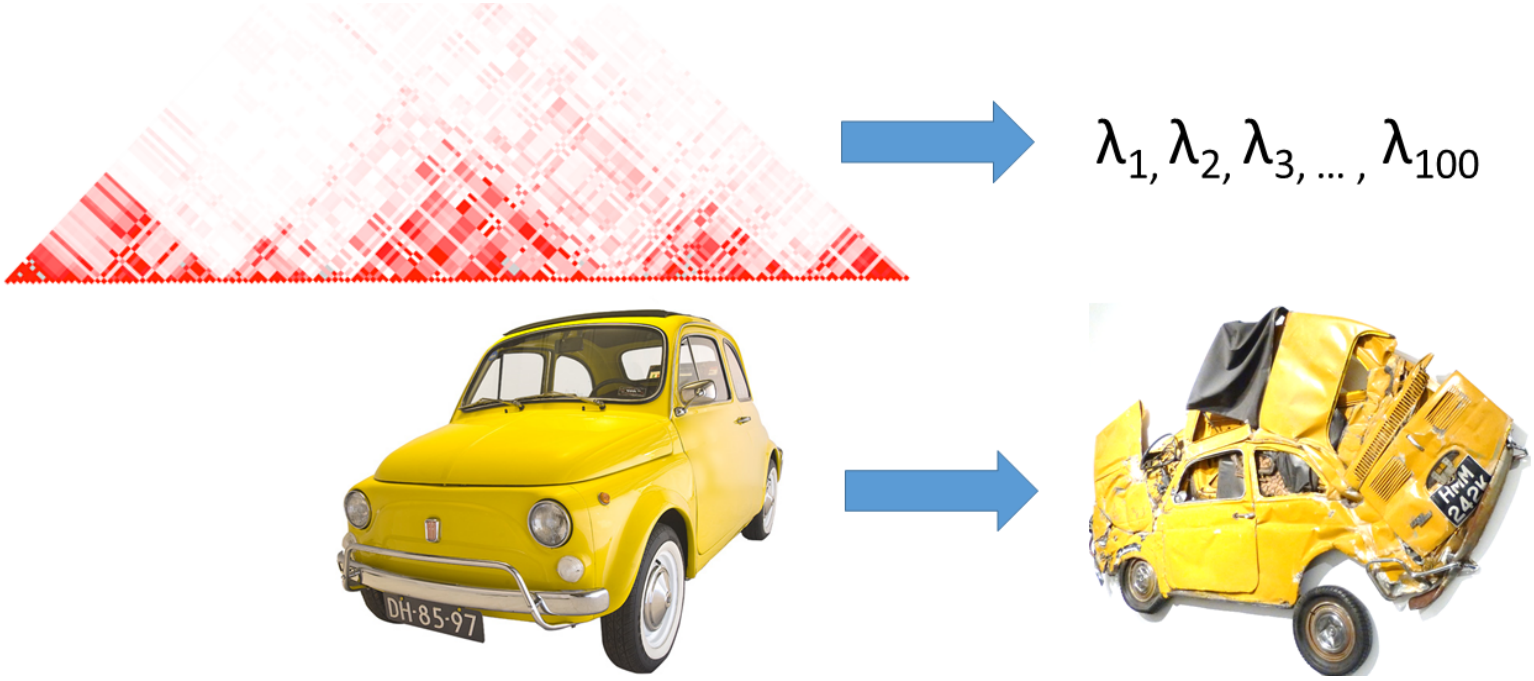
- fastSKAT approximates in two ways: first, use

  $$\lambda_1\chi_1^2 + \lambda_2\chi_1^2 + \lambda_3\chi_1^2 + \lambda_4\chi_1^2 + \ldots \lambda_{100}\chi_1^2 + \text{remainder term}$$

  ... where the remainder term comes from survey-sampling methods

Related: fastSKAT

fastSKAT's second speed-up is to approximate the large λ_j (eigenvalues of the LD matrix, basically) by Stochastic Singular Value Decomposition:



With these 'fixes', compute time grows with $m \times n$, so can be **several orders of magnitude faster** than standard methods.

Related: fastSKAT

Some examples of fastSKAT in practice:

- Group SNPs in SKAT by their Topologically Associating Domains (TADs) – taking 260 CPU days to 16 CPU hours
- Group SNPs by histone class across whole chromosomes – infeasible with standard methods, about 1 CPU day with fastSKAT

For details see Lumley et al (2018) – code available, also incorporated in next release of GENESIS.

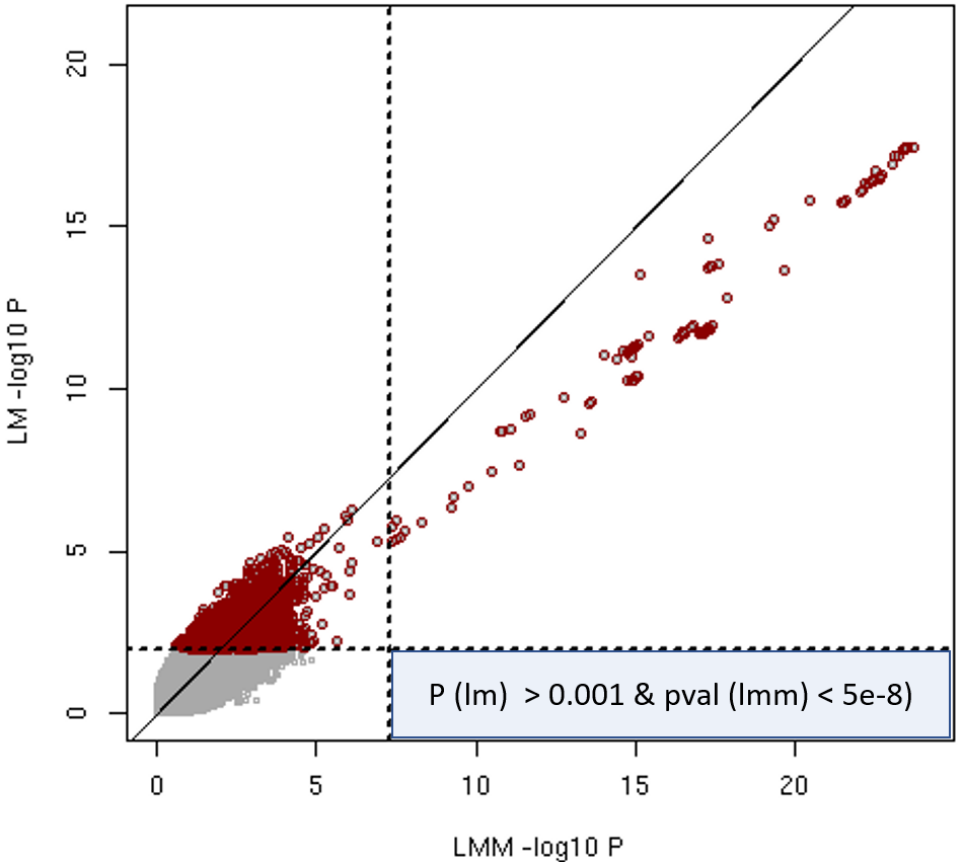
A 3rd fix: screen out uninteresting tests

We usually focus on getting the analysis 'right' with **very** high precision – for $p \approx 10^{-8}$ or smaller. But:

- Some imperfect analyses (e.g. ignoring relatedness, using LMMs not GLMMs, using meta-analysis, using very sparse kinships) **are good enough for screening**
- If results don't reach, say, $p = 0.01$ with these (true for $\approx 99\%$ of results) we can safely ignore them
- GENESIS does this for linear models, via `genesis_dscan()`
- Particularly useful for $n < 20,000$ and binary outcomes

A 3rd fix: screen out uninteresting tests

An example: screen with linear regression (LM) of GRM-adjusted residuals, if $p < 0.001$ run full LMM:



- Grey box shows (no) false negatives – or anything close
- Compute time $\times 10$ better – 7h 25 min LMM vs 43 min double-scan
- QQ plots can look odd – a minor issue

Bonus track: Cholesky decomposition

To invert **V**, we use *Cholesky decomposition*

- ... i.e. writing **V** as $\mathbf{L}\mathbf{L}^T$ for some lower-triangular **L** – analogous to taking a square root
- Our code saves and uses `chol(V)`
- This is easier to invert than **V** – using `chol2inv()` and also permits taking the inverse of a subset of **V**
- The subset property is very useful in GWAS, but may not be needed if everyone is included in analysis of every variant – as in some WGS work
- Eliminating the use of `chol2inv()` at each variant is a well-known trick for speeding up the association tests – hugely – as most of the CPU time is used running `chol2inv()`

GENESIS now offers both versions.

Bonus track: fewer iterations?

Compared to 'full' REML, the algorithm we use to fit the null has a few missing terms, in the *information matrix* – which determines how far to 'step' as we update the $\hat{\sigma}^2$ terms.

- This method is called AIREML. The missing steps average to zero (in a sense) and omitting them saves a lot of computing steps
- Omitting them is **not** critical – we just fit the null taking more iterations
- If we can do them cheaply, using full REML updates might lead to e.g. 4 iterative steps, not 7 – perhaps at the cost of using slightly more memory
- Worth revisiting if fitting null models takes considerable time

More generally: expect to have to hand-inspect null model fits, particularly if some $\hat{\sigma}^2$ terms go to zero.

Acknowledgments

Many thanks to:

- Jen, Stephanie, Tamar, Chaoyu
- Han Chen & Jeff O'Connell