GREML: Heritability Estimation Using Genomic Data

Rob Kirkpatrick & Mike Hunter

March 5th, 2020

(Some slides courtesy of Matt Keller)

Overview

I. Regression Estimates of V_A.

II. Genomic Relatedness Matrices.

III. GREML.

IV. Combining GREML & SEM.

V. mxGREML Design.

VI. mxGREML Implementation.

Using genetic similarity at SNPs to estimate V_A

- Determine extent to which genetic similarity at SNPs is related to phenotypic similarity
- Multiple approaches to derive unbiased estimate of V_A captured by measured (common) SNPs
  - Regression (Haseman-Elston)
  - Mixed effects models (GREML)
  - Bayesian (e.g., Bayes-R)
  - LD-score regression

Regression estimates of h^2

$$\theta_{ij} = Z_i Z_j$$

product of centered scores (here, z-scores)

$$E[\theta_{ij}] = COV(Z_i, Z_j)$$

$$E[\theta_{ij} \mid \hat{\pi}_{ij}] = \hat{\beta}_0 + \hat{\beta}_1 \hat{\pi}_{ij}$$

$$\hat{\beta}_1 = \hat{h}^2$$

(the slope of the regression is an estimate of h^2)

Regression estimates of h^2

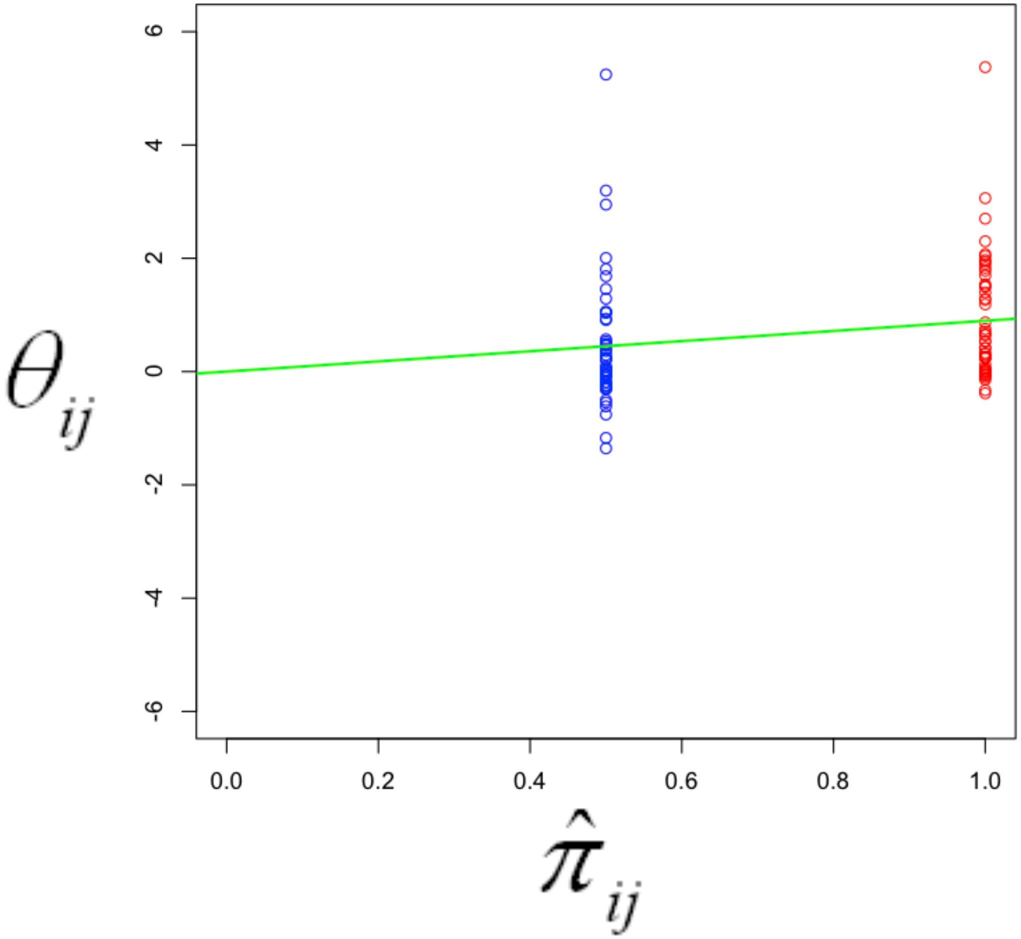
$$\theta_{ij} = Z_i Z_j$$

$$E[\theta_{ij}] = COV(Z_i, Z_j)$$

$$E[\theta_{ij} \mid \hat{\pi}_{ij}] = \hat{\beta}_0 + \hat{\beta}_1 \hat{\pi}_{ij}$$

$$\hat{\beta}_1 = \hat{h}^2$$

(the slope of the regression is an estimate of h^2)



Regression estimates of h^2

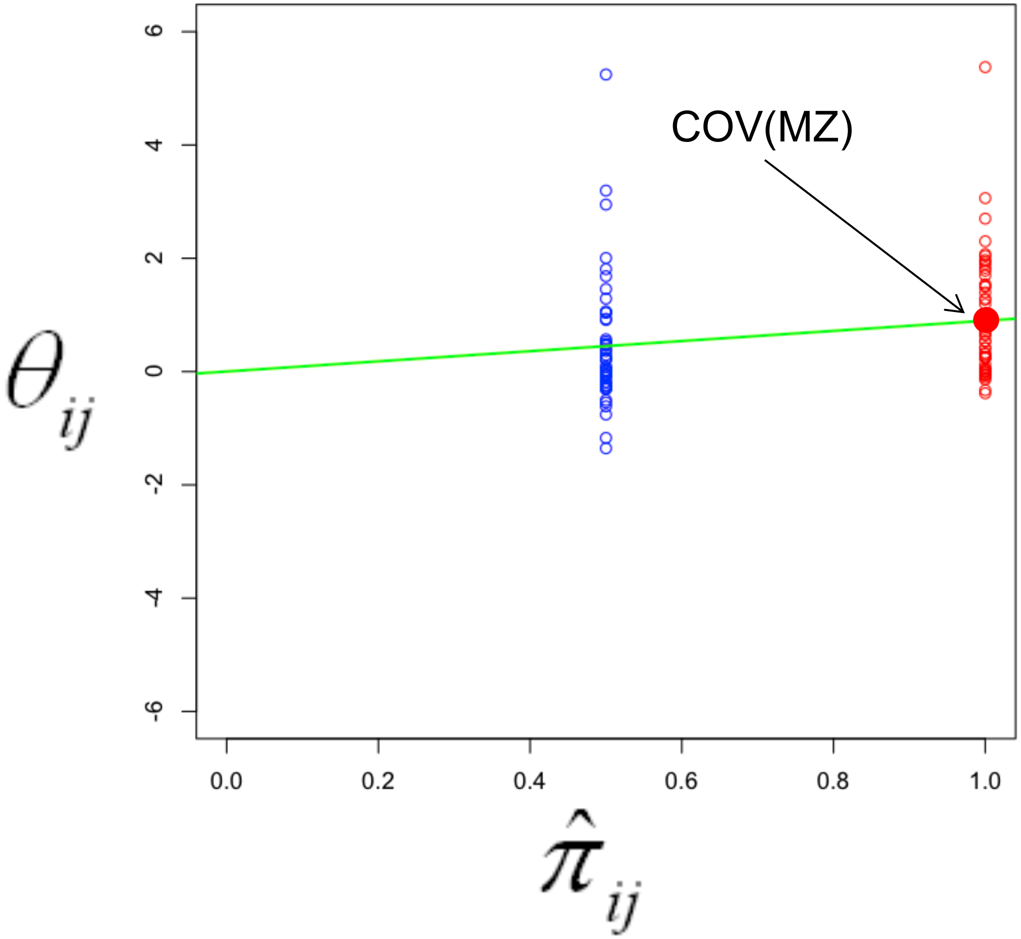
$$\theta_{ij} = Z_i Z_j$$

$$E[\theta_{ij}] = COV(Z_i, Z_j)$$

$$E[\theta_{ij} \mid \hat{\pi}_{ij}] = \hat{\beta}_0 + \hat{\beta}_1 \hat{\pi}_{ij}$$

$$\hat{\beta}_1 = \hat{h}^2$$

(the slope of the regression is an estimate of h^2)



Regression estimates of h^2

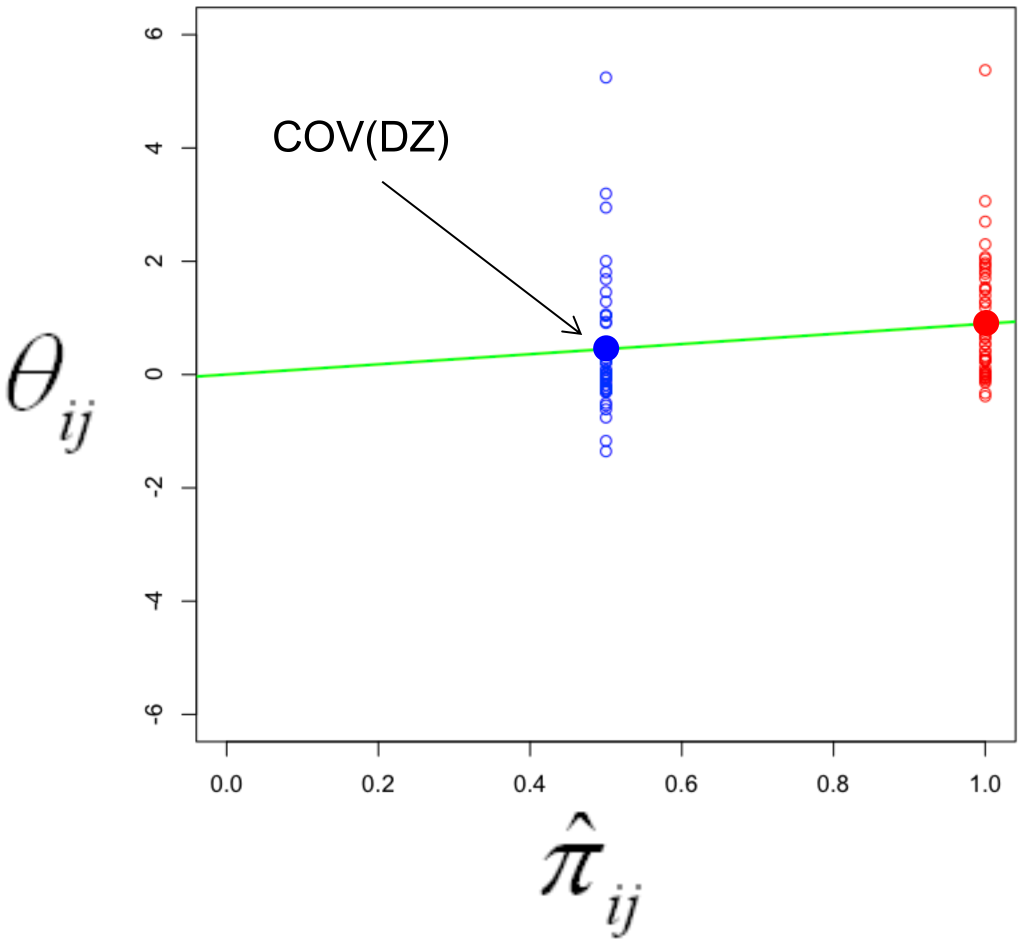
$$\theta_{ij} = Z_i Z_j$$

$$E[\theta_{ij}] = COV(Z_i, Z_j)$$

$$E[\theta_{ij} \mid \hat{\pi}_{ij}] = \hat{\beta}_0 + \hat{\beta}_1 \hat{\pi}_{ij}$$

$$\hat{\beta}_1 = \hat{h}^2$$

(the slope of the regression is an estimate of h^2)



Regression estimates of h^2

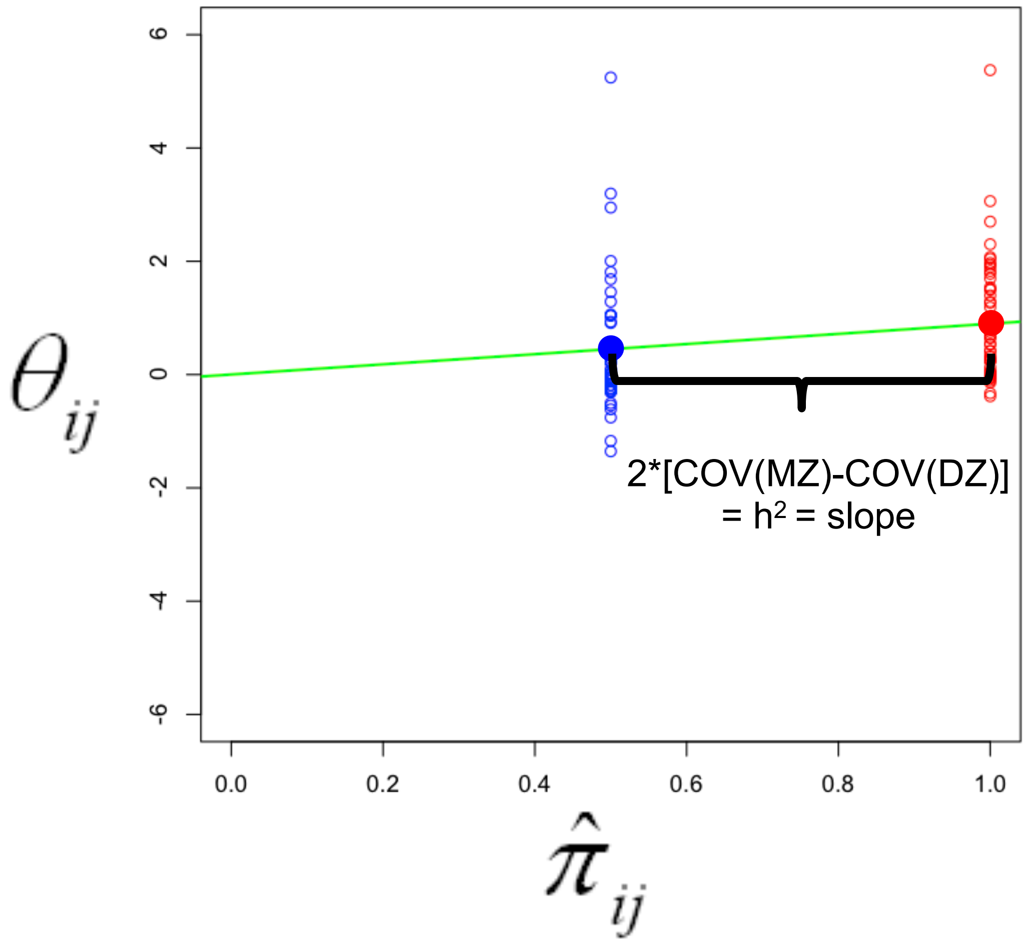
$$\theta_{ij} = Z_i Z_j$$

$$E[\theta_{ij}] = COV(Z_i, Z_j)$$

$$E[\theta_{ij} \mid \hat{\pi}_{ij}] = \hat{\beta}_0 + \hat{\beta}_1 \hat{\pi}_{ij}$$

$$\hat{\beta}_1 = \hat{h}^2$$

(the slope of the regression is an estimate of h^2)



2*[COV(MZ)-COV(DZ)] = h^2 = slope

Regression estimates of h^2

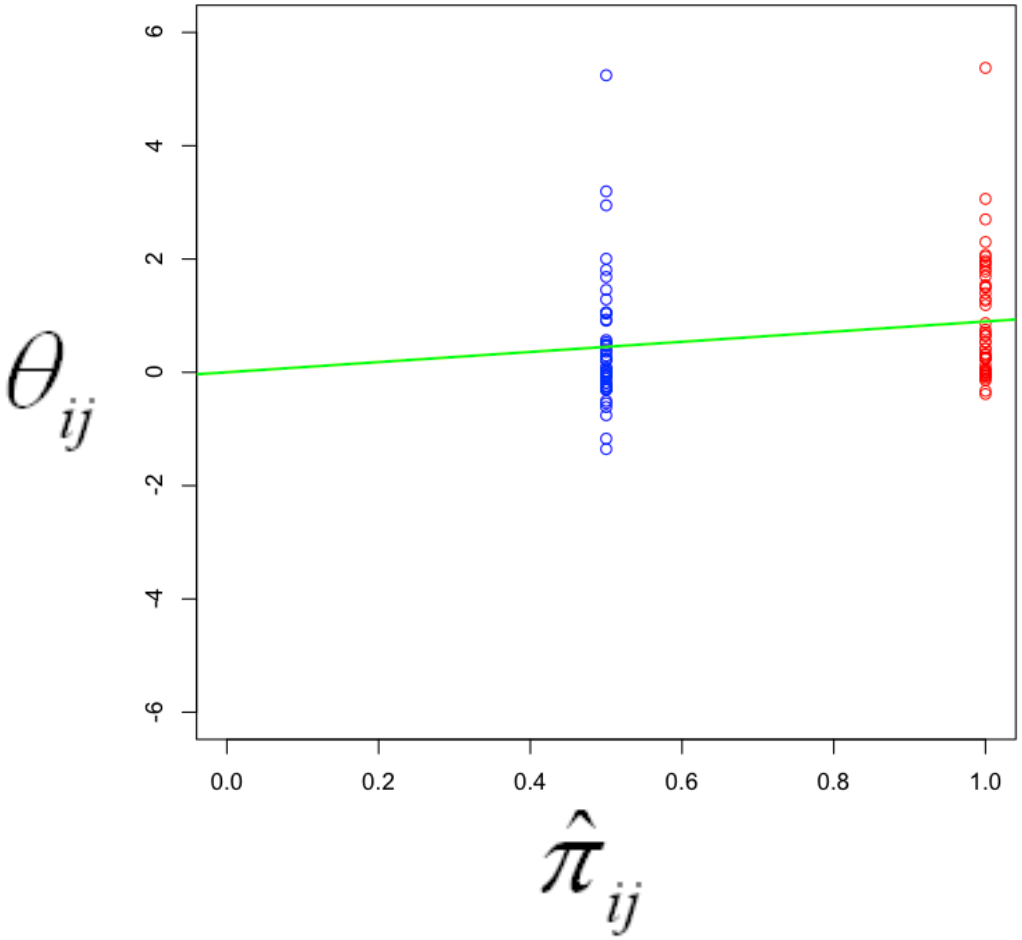
$$\theta_{ij} = Z_i Z_j$$

$$E[\theta_{ij}] = COV(Z_i, Z_j)$$

$$E[\theta_{ij} \mid \hat{\pi}_{ij}] = \hat{\beta}_0 + \hat{\beta}_1 \hat{\pi}_{ij}$$

$$\hat{\beta}_1 = \hat{h}^2$$

(the slope of the regression is an estimate of h^2)



Regression estimates of h^2

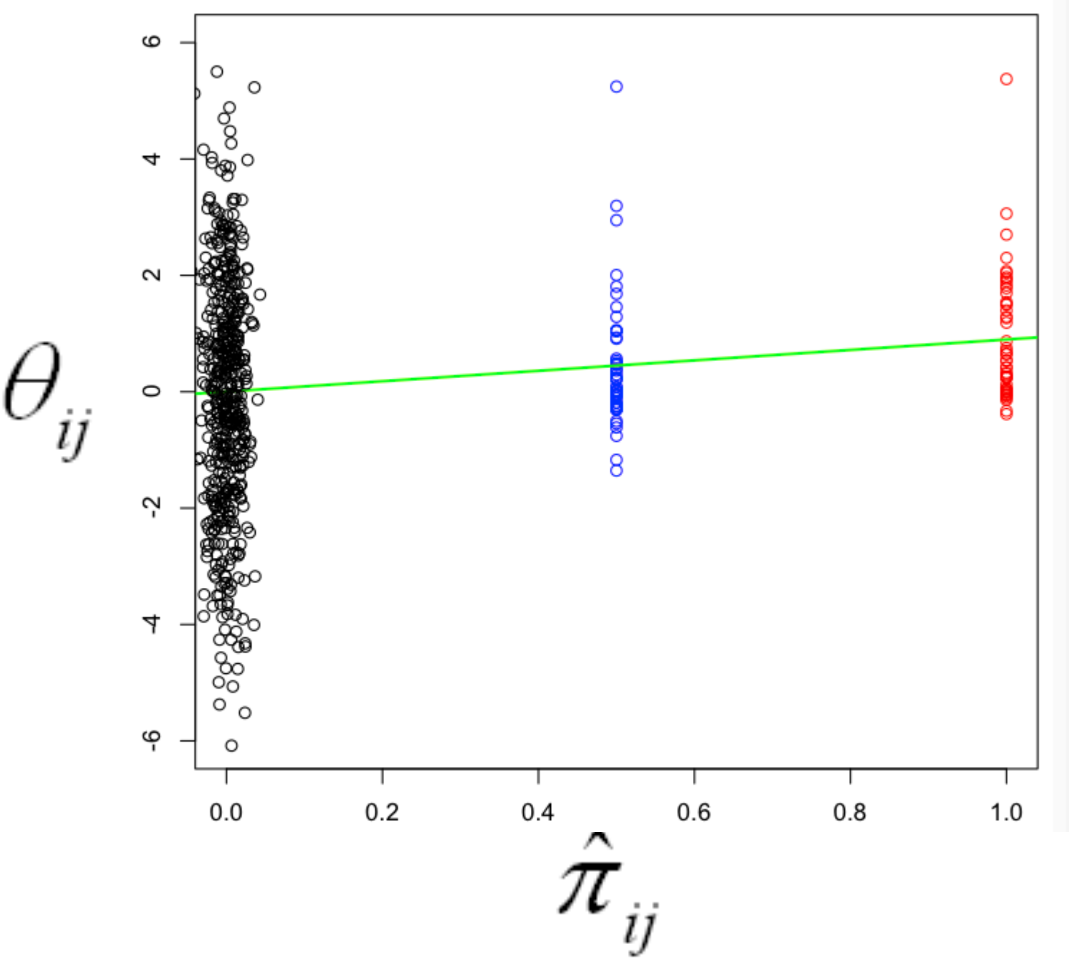
$$\theta_{ij} = Z_i Z_j$$

$$E[\theta_{ij}] = COV(Z_i, Z_j)$$

$$E[\theta_{ij} \mid \hat{\pi}_{ij}] = \hat{\beta}_0 + \hat{\beta}_1 \hat{\pi}_{ij}$$

$$\hat{\beta}_1 = \hat{h}^2$$

(the slope of the regression is an estimate of h^2)



Regression estimates of h^2_{snp}

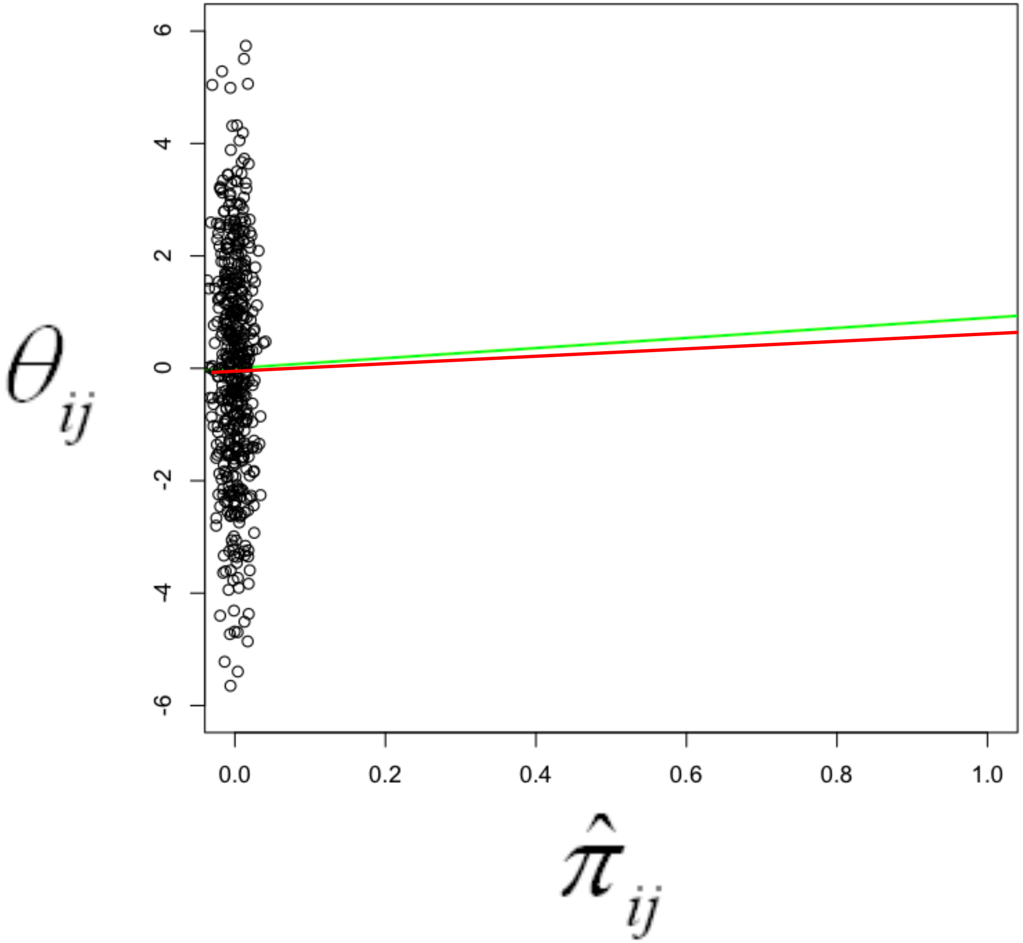
$$\theta_{ij} = Z_i Z_j$$

$$E[\theta_{ij}] = COV(Z_i, Z_j)$$

$$E[\theta_{ij} \mid \hat{\pi}_{ij}] = \hat{\beta}_0 + \hat{\beta}_1 \hat{\pi}_{ij}$$

$$\hat{\beta}_1 = \hat{h}^2$$

(the slope of the regression is an estimate of h^2)



Interpreting h^2 estimated from SNPs (h^2_{snp})

- If close relatives included (e.g., sibs), $h^2_{snp} \cong h^2$ estimated from a family-based method, because great influence of extreme pihats. Interpret h^2_{snp} as from these designs.
- If use ‘unrelateds’ (e.g., pihat < .05):
  - h^2_{snp} = proportion of V_P due to V_A captured by SNPs. Upper bound % V_P GWAS can detect
  - Gives idea of the aggregate importance of CVs tagged by SNPs
  - By not using relatives who also share environmental effects: (a) V_A estimate 'uncontaminated' by V_C & $V_{NA;}$ (b) does not rely on family study assumptions (e.g., r(MZ) > r(DZ) for only genetic reasons)

Comparison of approaches for estimating h^2_{snp}

APPROACH (METHOD)	**ADVANTAGES**	**DISADVANTAGES**
HE-regression	Fast. Point estimates usually unbiased	Large SEs (~30% larger than REML). SE estimates biased. Limited model building.

Comparison of approaches for estimating h^2_{snp}

APPROACH (METHOD)	**ADVANTAGES**	**DISADVANTAGES**
HE-regression	Fast. Point estimates usually unbiased	Large SEs (~30% larger than REML). SE estimates biased. Limited model building.
LD-score regression	Requires only summary statistics; mostly robust to stratification/relatedness	Does not give good estimates of variance due to rare CVs

Comparison of approaches for estimating h^2_{snp}

APPROACH (METHOD)	**ADVANTAGES**	**DISADVANTAGES**
HE-regression	Fast. Point estimates usually unbiased	Large SEs (~30% larger than REML). SE estimates biased. Limited model building.
LD-score regression	Requires only summary statistics; mostly robust to stratification/relatedness	Does not give good estimates of variance due to rare CVs
GREML (e.g., GCTA	Point estimates & SEs usually unbiased. Well maintained & easy to use	Limited model-building (e.g., no nonlinear constraints).

II. Genomic Relatedness Matrices

Genomic Relatedness Matrices

Consider $\mathbf{S}$, an N×m matrix of genotypes expressed as reference-allele counts, where N is the number of participants and m is the number of markers (SNPs, say):

$$\mathbf{S}=\begin{bmatrix}0 & 2 & 0 & 1 & 1 & \cdots \\ 0 & 1 & 1 & 1 & 2 & \cdots \\ 0 & 1 & 1 & 0 & 0 & \cdots \\ 0 & 2 & 1 & 0 & 2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots\end{bmatrix}$$

Genomic Relatedness Matrices

Let $\mathbf{W}$ denote $\mathbf{S}$, after its columns have been standardized to have zero mean and unit variance. That is, the *ij*th element of $\mathbf{W}$ is

$$w_{ij} = \frac{s_{ij} - 2\mathscr{p}_j}{\sqrt{2\mathscr{p}_j(1 - \mathscr{p}_j)}}$$

where $\mathscr{p}_j$ is the reference-allele frequency of marker *j*.

Genomic Relatedness Matrices

The GRM is then

$$\mathbf{G} = \frac{1}{m}\mathbf{W}\mathbf{W}^T$$

and thus is an *N*×*N* matrix of genomic-relatedness coefficients. These coefficients are “allele-frequency-weighted” IBS coefficients.

In a random sample from a homogenous, randomly-mating population:

- The diagonal elements are expected to equal 1.
- The off-diagonals are expected to equal zero.
- However, there will be variance around these expectations. We will use this variance to get leverage on estimating $V_{A,SNP}$.

Genomic Relatedness Matrices

- OpenMx does not compute GRMs from raw genotype data—use GCTA, plink, etc.
- Going from genotypes to GRM can be more complicated—correction for possible uneven LD around trait-relevant loci[1].
- Possible to use >1 GRM in analysis—bin markers by, e.g.
  - Chromosome.
  - Allele frequency.
  - Biological pathway.

[1] Speed, D., et al. (2013). *AJHG*, *91*, 1011-1021. doi: 10.1016/j.ajhg.2012.10.010.

III. GREML

GREML Model

(here, *n*=3, *q*=2 fixed effects, *m*=3 SNPs)

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{W}\mathbf{u} + \mathbf{e}$$

$$\begin{bmatrix} 3 \\ -5 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 & -1.2 \\ 1 & 0.8 \\ 1 & 0.4 \end{bmatrix} * \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{bmatrix} +$$

observed
y

design matrix
of fixed effects
(intercept & 1
covariate)

fixed
effects

GREML Model

(here, *n*=3, *q*=2 fixed effects, *m*=3 SNPs)

$$\mathbf{y} = \mathbf{X\beta} + \mathbf{Wu} + \mathbf{e}$$

$$\begin{bmatrix} 3 \\ -5 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 & -1.2 \\ 1 & 0.8 \\ 1 & 0.4 \end{bmatrix} * \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{bmatrix} + \overset{\text{n} \times \text{m}}{\begin{bmatrix} 1.15 & 2.10 & -.68 \\ -.58 & -.23 & .03 \\ 1.15 & -.23 & .03 \end{bmatrix}} * \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \hat{u}_3 \end{bmatrix} +$$

observed y

design matrix of fixed effects (intercept & 1 covariate)

fixed effects

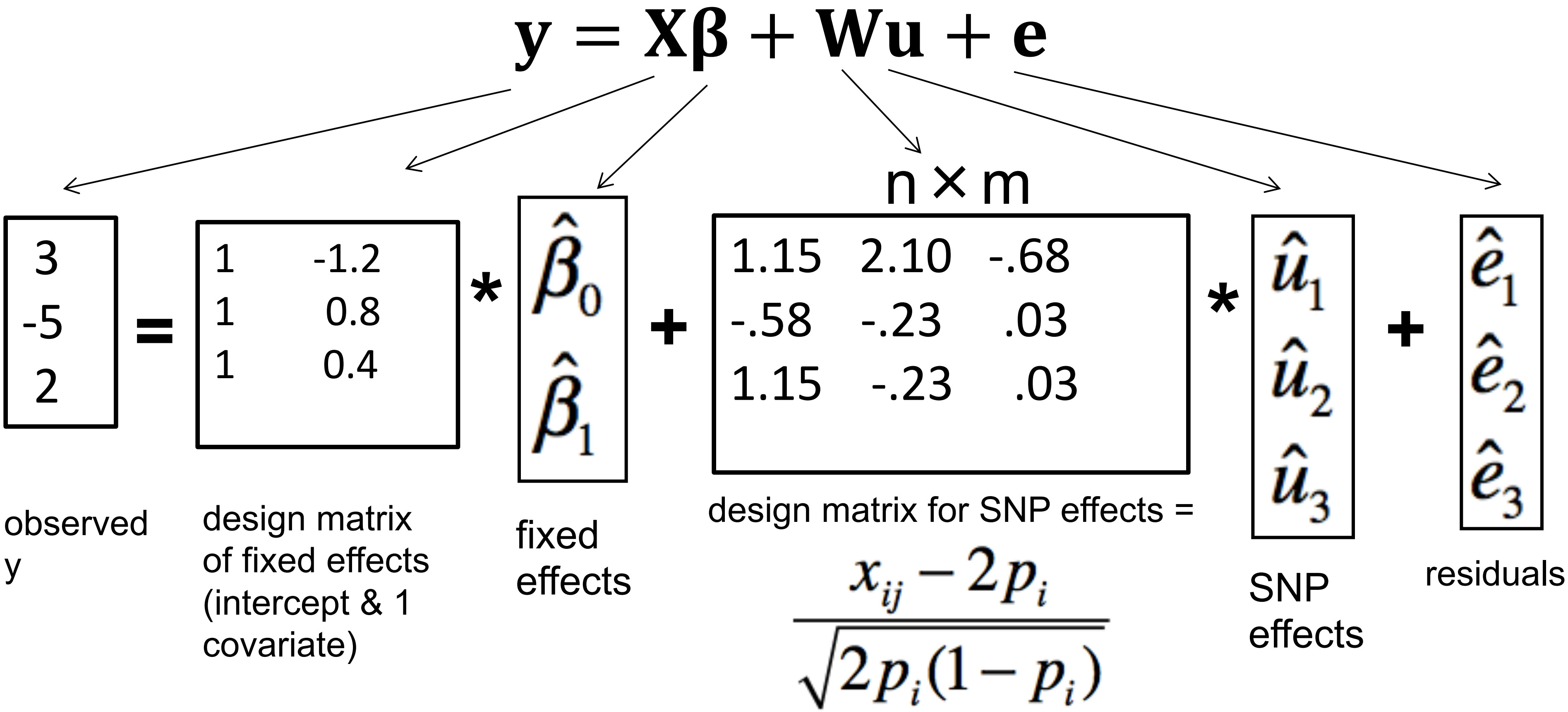
design matrix for SNP effects =

$$\frac{x_{ij} - 2p_i}{\sqrt{2p_i(1-p_i)}}$$

SNP effects

GREML Model

(here, *n*=3, *q*=2 fixed effects, *m*=3 SNPs)



$$\mathbf{y} = \mathbf{X\beta} + \mathbf{Wu} + \mathbf{e}$$

n × m

$$\begin{bmatrix} 3 \\ -5 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 & -1.2 \\ 1 & 0.8 \\ 1 & 0.4 \end{bmatrix} * \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{bmatrix} + \begin{bmatrix} 1.15 & 2.10 & -.68 \\ -.58 & -.23 & .03 \\ 1.15 & -.23 & .03 \end{bmatrix} * \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \hat{u}_3 \end{bmatrix} + \begin{bmatrix} \hat{e}_1 \\ \hat{e}_2 \\ \hat{e}_3 \end{bmatrix}$$

observed y

design matrix of fixed effects (intercept & 1 covariate)

fixed effects

design matrix for SNP effects =

$$\frac{x_{ij} - 2p_i}{\sqrt{2p_i(1-p_i)}}$$

SNP effects

residuals

GREML Model

(after removing fixed effects on y)

$$\mathbf{y} - \mathbf{X\beta} = \mathbf{Wu} + \mathbf{e}$$

$$\begin{bmatrix} -.64 \\ -2.58 \\ 3.21 \end{bmatrix} = \begin{bmatrix} 1.15 & 2.10 & -.68 \\ -.58 & -.23 & .03 \\ 1.15 & -.23 & .03 \end{bmatrix} * \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \hat{u}_3 \end{bmatrix} + \begin{bmatrix} \hat{e}_1 \\ \hat{e}_2 \\ \hat{e}_3 \end{bmatrix}$$

residuals
y

$$\frac{x_{ij} - 2p_i}{\sqrt{2p_i(1-p_i)}}$$

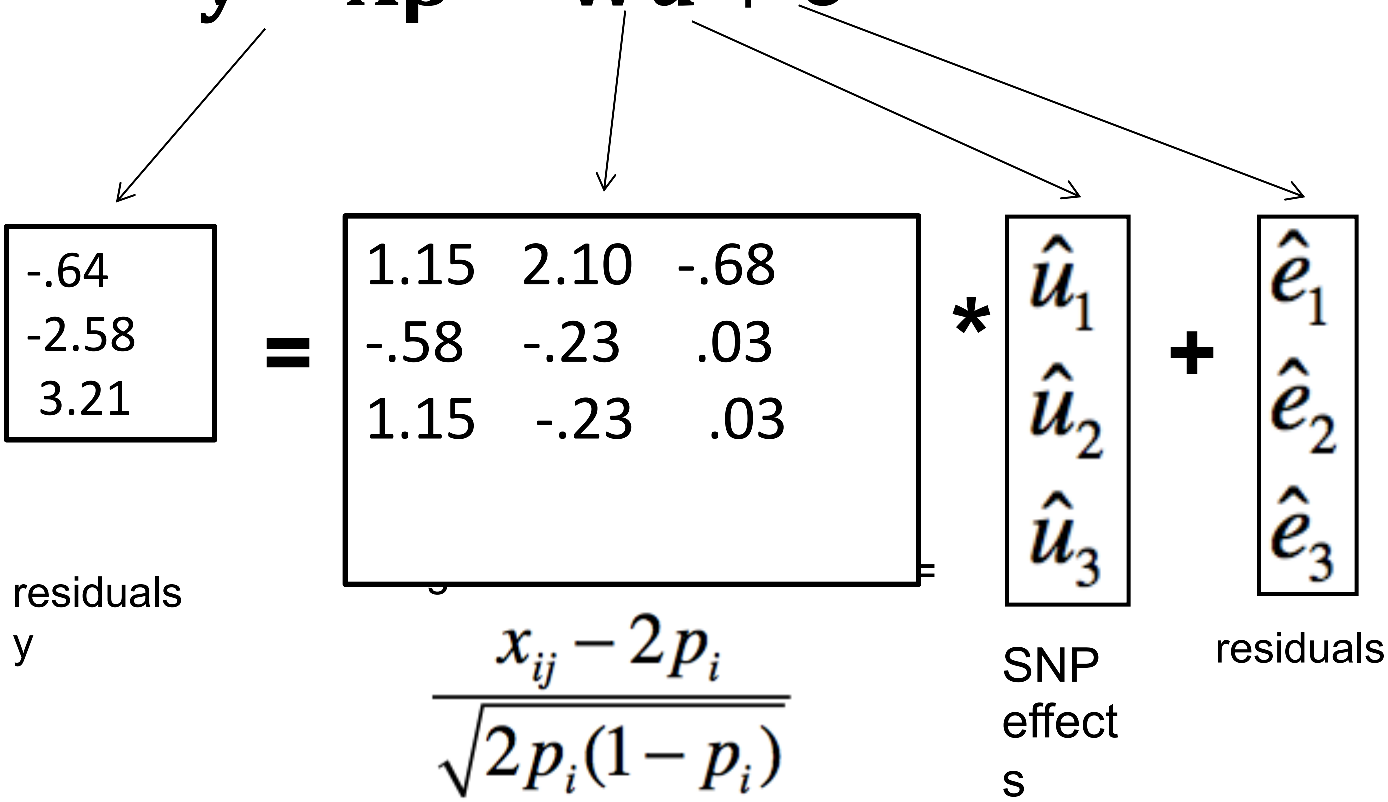
SNP
effects

residuals

GREML Model

(after removing fixed effects on y)

$$\mathbf{y} - \mathbf{X}\boldsymbol{\beta} = \mathbf{W}\mathbf{u} + \mathbf{e}$$



$$\begin{bmatrix} -.64 \\ -2.58 \\ 3.21 \end{bmatrix} = \begin{bmatrix} 1.15 & 2.10 & -.68 \\ -.58 & -.23 & .03 \\ 1.15 & -.23 & .03 \end{bmatrix} * \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \hat{u}_3 \end{bmatrix} + \begin{bmatrix} \hat{e}_1 \\ \hat{e}_2 \\ \hat{e}_3 \end{bmatrix}$$

residuals
y

$$\frac{x_{ij} - 2p_i}{\sqrt{2p_i(1-p_i)}}$$

SNP
effect
s

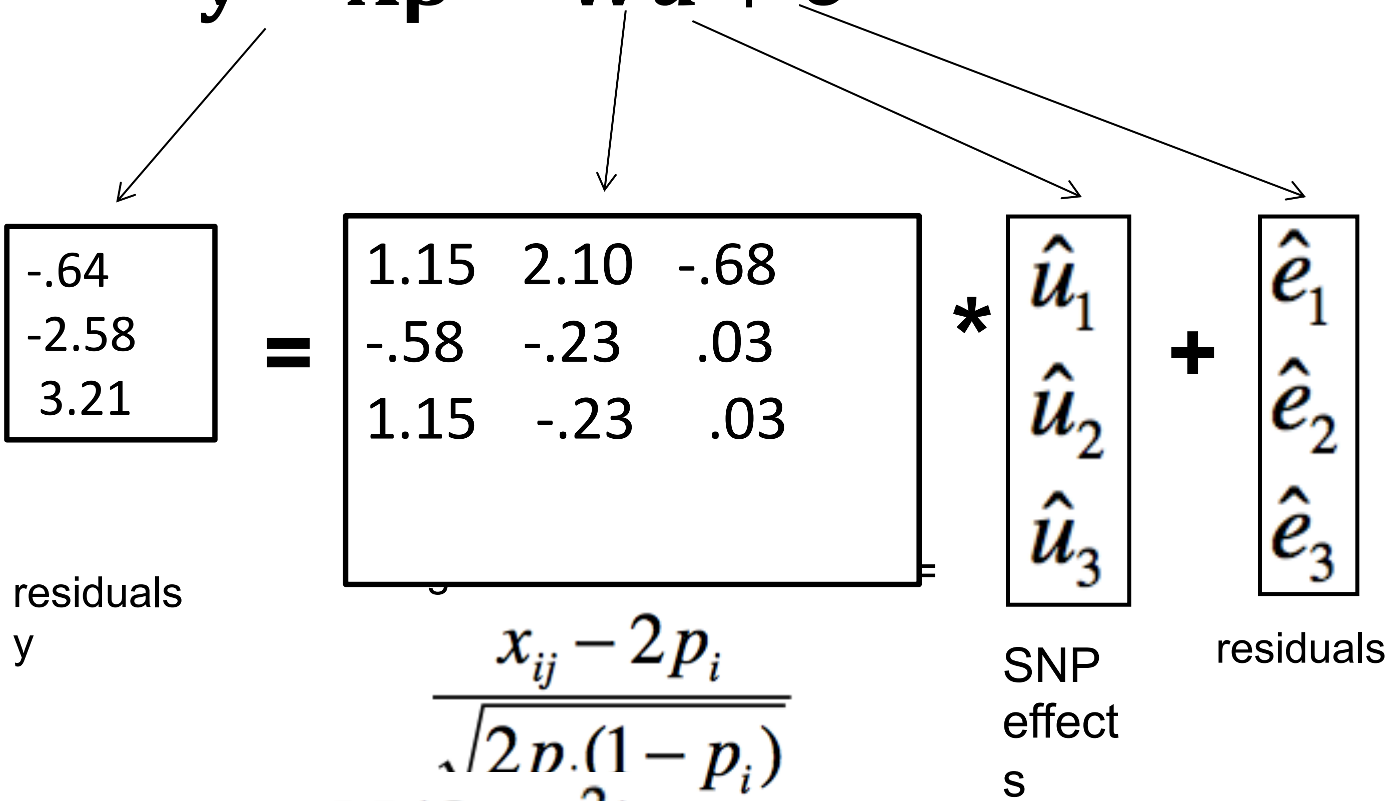
residuals

We aren’t interested in estimating each *u*$_i$ because *m* >> *n* usually, and because such individual estimates would be unreliable. Instead, estimate the variance of *u*$_i$.

GREML Model

(after removing fixed effects on y)

$$\mathbf{y} - \mathbf{X\beta} = \mathbf{Wu} + \mathbf{e}$$



$$\begin{bmatrix} -.64 \\ -2.58 \\ 3.21 \end{bmatrix} = \begin{bmatrix} 1.15 & 2.10 & -.68 \\ -.58 & -.23 & .03 \\ 1.15 & -.23 & .03 \end{bmatrix} * \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \hat{u}_3 \end{bmatrix} + \begin{bmatrix} \hat{e}_1 \\ \hat{e}_2 \\ \hat{e}_3 \end{bmatrix}$$

residuals y

$$\frac{x_{ij} - 2p_i}{\sqrt{2p_i(1-p_i)}}$$

SNP effects

residuals

We assume $u \sim N(0,\sigma_u^2)$

and therefore $\sigma_A^2 = \sum_{i=1}^{m} \sigma_u^2 = m\sigma_u^2$

GREML Model

(we treat *u* as random and estimate σ_u^2 and thus σ_A^2)

$$\begin{aligned}\mathrm{var}(\mathbf{y} \mid \mathbf{X}) &= \mathbf{W}\mathbf{W}^T\sigma_u^2 + \mathbf{I}\sigma_e^2 \\ &= \mathbf{W}\mathbf{W}^T(\sigma_A^2/m) + \mathbf{I}\sigma_e^2 \\ &= \mathbf{G}\sigma_A^2 + \mathbf{I}\sigma_e^2\end{aligned}$$

$$\boxed{\begin{matrix} .41 & 1.65 & -2.05 \\ 1.65 & 6.66 & -8.28 \\ -2.05 & -8.28 & 10.3 \end{matrix}} = \boxed{\begin{matrix} 1.02 & -.01 & -.02 \\ -.01 & 1.00 & .02 \\ -.02 & .02 & .98 \end{matrix}}\ \sigma_A^2 + \boxed{\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}}\ \sigma_e^2$$

observed n-by-n var/covar matrix of residuals y

Genomic Relationship Matrix (GRM) at measured SNPs. Each element =

$$\hat{\pi}_{jk} = \frac{1}{m}\sum_i \frac{(x_{ij} - 2p_i)(x_{ik} - 2p_i)}{2p_i(1-p_i)}$$

Identity matrix

GREML Model

(we treat *u* as random and estimate σ_u^2 and thus σ_A^2)

$$
\begin{aligned}
\mathrm{var}(\mathbf{y} \mid \mathbf{X}) &= \mathbf{W}\mathbf{W}^T\sigma_u^2 + \mathbf{I}\sigma_e^2 \\
&= \mathbf{W}\mathbf{W}^T(\sigma_A^2/m) + \mathbf{I}\sigma_e^2 \\
&= \mathbf{G}\sigma_A^2 + \mathbf{I}\sigma_e^2
\end{aligned}
$$

$$
\underbrace{\boxed{\begin{matrix} .41 & 1.65 & -2.05 \\ 1.65 & 6.66 & -8.28 \\ -2.05 & -8.28 & 10.3 \end{matrix}}}_{\text{observed var/covar}}
\;\mathbf{=}\;
\underbrace{\boxed{\begin{matrix} 1.02 & -.01 & -.02 \\ -.01 & 1.00 & .02 \\ -.02 & .02 & .98 \end{matrix}}\,\sigma_A^2
\;\mathbf{+}\;
\boxed{\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}}\,\sigma_e^2}_{\text{implied var/covar}}
$$

REML find values of σ_A^2 & σ_e^2 that maximizes the likelihood of the observed data. Intuitively, this makes the observed and implied var-covar matrices be as similar as possible.

Individual QC

- Remove individuals missing > ~.02
- Remove close relatives (e.g., --grm-cutoff 0.05)
  - Correlation between pi-hats and shared environment can inflate h^2_{snp} estimates
- Control for stratification (usually 5 or 10 PCs)
  - Different prevalence rates (or ascertainments) between populations can show up as h^2_{snp}
- Control for plates and other technical artifacts
  - Be careful if cases & controls are not randomly placed on plates (can create upward bias in h^2_{snp})

Big picture: Using SNPs to estimate h^2

- Independent approach to estimating h^2
  - Different assumptions than family models. Increasingly tortuous reasoning to suggest traits aren't heritable because methodological flaws
- When using SNPs with same allele frequency distribution as CVs, provides unbiased estimate of h^2
- When using common (array) SNPs to estimated relatedness, generally provides downwardly biased estimate of h^2
  - "Still missing" h^2 ($h^2_{family} – h^2_{snp}$) provides insight into the importance of rare variants, non-additive, or biased h^2_{family}.
- But not a panacea. Biases still exist. Issues need to be worked out (e.g., assortative mating, etc.).

III. Combining GREML & SEM.

GSEM[1]

- R package by Beate St Pourcain (https://gitlab.gwdg.de/beate.stpourcain/gsem).
- 1 dedicated function each for fitting CommPthwy, IndePthwy, & "Cholesky".
- Specialized—fast & lean.
- Uses fast BLAS (e.g., ATLAS) for good performance.
- ML fit.
- Path-coefficient parameterization.

[1] St Pourcain et.al. (2018). *Biological Psychiatry 83*: 598-606

mxGREML

- OpenMx feature.

- Available in *OpenMx* since v2.2 (June 2015).

- Still being developed.

IV. mxGREML Design

Overview of GREML in *OpenMx*

- All participants’ scores on all phenotypes get “stacked” into a single vector, **y**.
- Input dataset is in “vanilla” wide format--has 1 row per individual:

```
           y     x
[1,]  7.3119 -0.33
[2,]  0.5069 -0.64
[3,] -1.8111 -0.78
[4,] -8.7180 -0.12
[5,]  6.5651 -0.81
[6,] -2.2380 -0.14
```

Overview of GREML in *OpenMx*

- All participants' scores on all phenotypes get "stacked" into a single vector, **y**.
- "Definition variables" not allowed/needed.
  - User specifies onto which covariates each phenotype is to be regressed.

Overview of GREML in *OpenMx*

- All participants' scores on all phenotypes get "stacked" into a single vector, **y**.
- "Definition variables" not allowed/needed.
- Ordinal phenotypes (incuding binary) must be treated as though continuous.
  - (You correct the h^2 estimate for this fact later.)

Overview of GREML in *OpenMx*

- All participants' scores on all phenotypes get "stacked" into a single vector, **y**.
- "Definition variables" not allowed/needed.
- Ordinal phenotypes (incuding binary) must be treated as though continuous.
- User must specify model for **y**.
  - Mean of **y** conditioned on covariates, which are columns of matrix **X**.
  - var(**y** | **X**) is covariance matrix, **V**, which user must define.

GREML: New, Big Idea

- In previous analyses we've done so far in OpenMx, the unit of analysis was the family (e.g., twin pair).
- But if we can use DNA to determine the weak genetic resemblance among classically unrelated individuals, we can treat the entire sample as one large, extended "family".
- Thus, in GREML, the whole sample is one case, and the sole unit of analysis.

GREML in *OpenMx*: assumptions

1. Conditional on covariates **X**, phenotype vector **y** is a single draw from a multivariate-normal distribution having (in general) dense covariance matrix, **V**.
2. Random effects are normally distributed.
3. GLS regression (using $\mathbf{V}^{-1}$) is adequate model for phenotypic mean.

V. mxGREML Implementation

Overview of mxGREML Feature

0. Condensed matrix slots.

1. GREML expectation & (incl. automated data-structuring).

2. GREML fitfunction.

Large Matrices and Memory Efficiency

- Demo script…
- Main idea—when your OpenMx script involves large matrices that contain no free parameters:
  1. Place `options(mxCondenseMatrixSlots=TRUE)` near beginning of script.
  2. Always access slots of MxMatrix objects with `$`, and never with `@`.

GREML Expectation

- Compatible with GREML fitfunction and ML fitfunction (but…).
- In OpenMx terms, requires raw continuous data...
- But, strictly speaking, does not require raw genotypic or phenotypic data--at minimum, you need:
  - 1 or more GRMs.
  - Phenotype scores with covariates partialled out.

GREML Expectation

- Compatible with GREML fitfunction and ML fitfunction (but…).
- In OpenMx terms, requires raw continuous data.
- User tells it:
  - Which algebra/matrix is **V**.
  - Arguments for data-structuring.
  - Whether & how to resize **V** at runtime due to missing data.

Imagine we have 3 participants and 3 phenotypes, and we're using the same covariate, *x*, for all 3 phenotypes…

`blockByPheno=TRUE, staggerZeroes=TRUE`

$$\mathbf{y} = \begin{bmatrix} ALC_1 \\ ALC_2 \\ ALC_3 \\ CAN_1 \\ CAN_2 \\ CAN_3 \\ NIC_1 \\ NIC_2 \\ NIC_3 \end{bmatrix} \qquad \mathbf{X} = \begin{bmatrix} 1 & x_1 & 0 & 0 & 0 & 0 \\ 1 & x_2 & 0 & 0 & 0 & 0 \\ 1 & x_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & x_1 & 0 & 0 \\ 0 & 0 & 1 & x_2 & 0 & 0 \\ 0 & 0 & 1 & x_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & x_1 \\ 0 & 0 & 0 & 0 & 1 & x_2 \\ 0 & 0 & 0 & 0 & 1 & x_3 \end{bmatrix}$$

GREML fitfunction

- Support for analytic derivatives (which we will not do).
- Otherwise, use SLSQP, which can calculate numeric fitfunction derivatives in parallel.

mxGREML Practical

- In the interest of time, we will fit a very simple monophenotype AE model…
- See also: https://github.com/RMKirkpatrick/mxGREMLdemos .

Miscellaneous—stuff we didn’t cover

- **Be careful using GREML with any kind of ascertained sample.**
- Use of >1 GRM (or other such “relatedness matrix”).
- Computational shortcuts available for simple models (e.g., diagonalization).
- Technical aspects of computing GRMs.

Acknowledgements

- NIH grant DA026119
- Mike Neale (PI)
- Lindon Eaves
- Mike Hunter & Joshua Pritikin
- The rest of the OpenMx Development Team