Scalable algorithms for GWAS, genotype imputation, and ancestry inference

Benjamin Chu

Graduate program in Biomathematics
Department of Computational Medicine
University of California, Los Angeles

University of Michigan (online), December 14, 2020

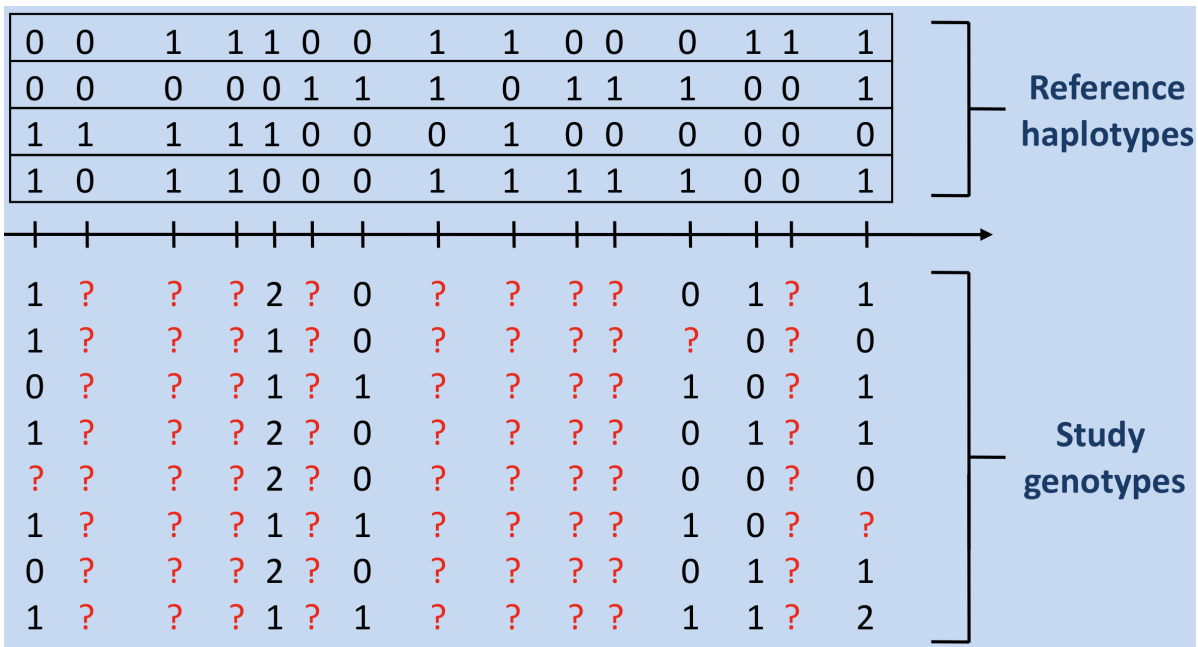
Outline

- Phasing, Imputation, and admixture estimation
  - Introduction
  - What we did
  - Potential future projects
- IHT for association studies
  - Introduction
  - What we did
  - Potential future projects

Part 1: Imputation, phasing, and admixture inference

What is genotype imputation?

- Michigan Imputation Server imputes > 10 million genomes annually
- Purpose: increase number of markers; meta analysis
- Inputs
  1. GWAS data with $\sim 10^6$ unphased genotypes (entries $0, 1, 2$)
  2. Reference panel with $10^7 \sim 10^8$ phased genotypes (entries $0, 1$)
- Output: Phased genotypes at all markers



Source: `http://mathgen.stats.ox.ac.uk/impute/impute_v2.html`

Background

- Reference panels are getting **big**!
  - 2012: 1k samples at 28M SNPs (1000 genomes phase 1)
  - 2016: 32k samples at 39M SNPs (Haplotype reference consortium)
  - 2019: 97k samples at 308M sites from sequences (TOPMed)
- Genotype imputation usually employ hidden Markov models (HMM)
  - e.g. Minimac 4, Beagle 5.1, Impute 5
  - All based on Li and Stephens, *Genetics, 165(4):2213–2233, 2003*
  - Very similar accuracy
  - Requires prephasing data = chaining softwares
- HMM methods are slow!
  - They are $> 10,000$ times faster since version 1 of the initial software, but some problems are still (increasingly) unmanageable.

Case study: HMM methods are slow

Using Minimac 4 (used on imputation server),

- Imputing 1000 samples[1], with HRC panel (2016) requires $12,892$ seconds $\approx$ 3.5h on chromosome 10
- With 23 pairs chromosome and 500k samples in UK Biobank, this imputation will take $3.5 \times 23 \times 500$ hours $\approx$ **4.7** years.

Our software `MendelImpute.jl`

- takes $\sim$ 20 days
- requires no prephasing and much less computer memory
- generates ancestry data automatically
- might lead to better data compression

[1] on local computer: 64 GB RAM; Intel i9 9920X CPU with 12 cores

Imputation within a small genomic region (window)

Consider genotype vector $\boldsymbol{x}$ and (unique) reference haplotypes $\boldsymbol{h}_1, ..., \boldsymbol{h}_d$ in the small genomic window. Imputation is done by minimizing

$$||\boldsymbol{x} - \boldsymbol{h}_i - \boldsymbol{h}_j||_2^2 = ||\boldsymbol{x}||_2^2 + ||\boldsymbol{h}_i||_2^2 + ||\boldsymbol{h}_j||_2^2 + 2\boldsymbol{h}_i^t\boldsymbol{h}_j - 2\boldsymbol{x}^t\boldsymbol{h}_i - 2\boldsymbol{x}^t\boldsymbol{h}_j$$

over all $\boldsymbol{h}_i, \boldsymbol{h}_j$ pairs. Solution:

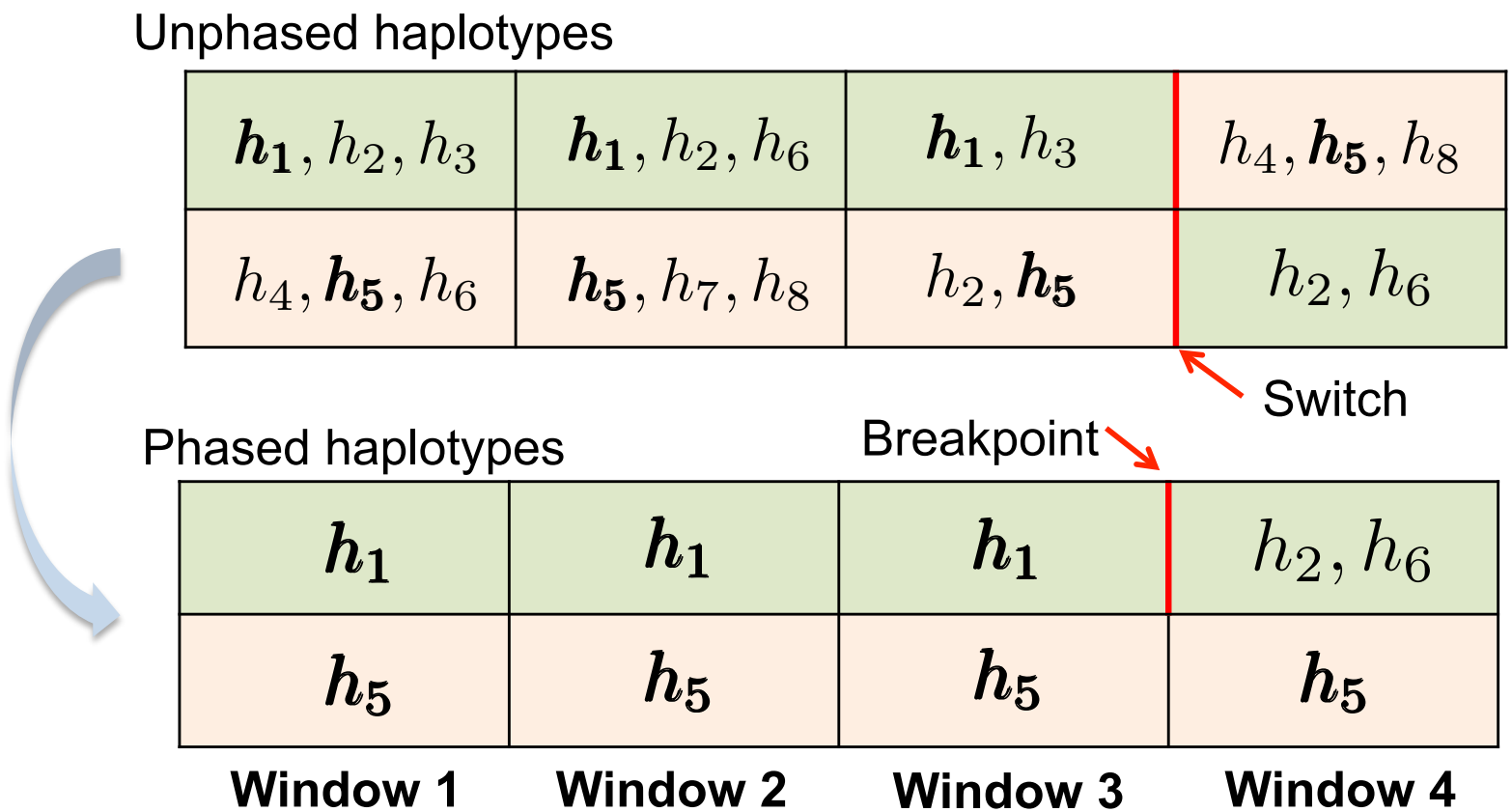
1. Collect $||\boldsymbol{h}_i||_2^2 + ||\boldsymbol{h}_j||_2^2 + 2\boldsymbol{h}_i^t\boldsymbol{h}_j$ terms into matrix $\boldsymbol{M}$
2. Collect $-2\boldsymbol{x}^t\boldsymbol{h}_i - 2\boldsymbol{x}^t\boldsymbol{h}_j$ terms into matrix $\boldsymbol{N}$
3. Search for the minimum in the upper triangular matrix $\boldsymbol{M} + \boldsymbol{N}$

where

- $\boldsymbol{M}, \boldsymbol{N}$ are efficiently assembled from $\boldsymbol{H}^t\boldsymbol{H}$ and $\boldsymbol{X}^t\boldsymbol{H}$
- Parallelization is achieved by treating different windows independently

Phasing: extend haplotypes across windows

- Unique haplotypes $\boldsymbol{h}_i, \boldsymbol{h}_j$ expands to sets of equivalent haplotypes.
- These sets are intersected across windows.
- Eventually intersection becomes empty, representing ancient or contemporary recombination events
- We parallelize phasing over samples, since samples are independent.



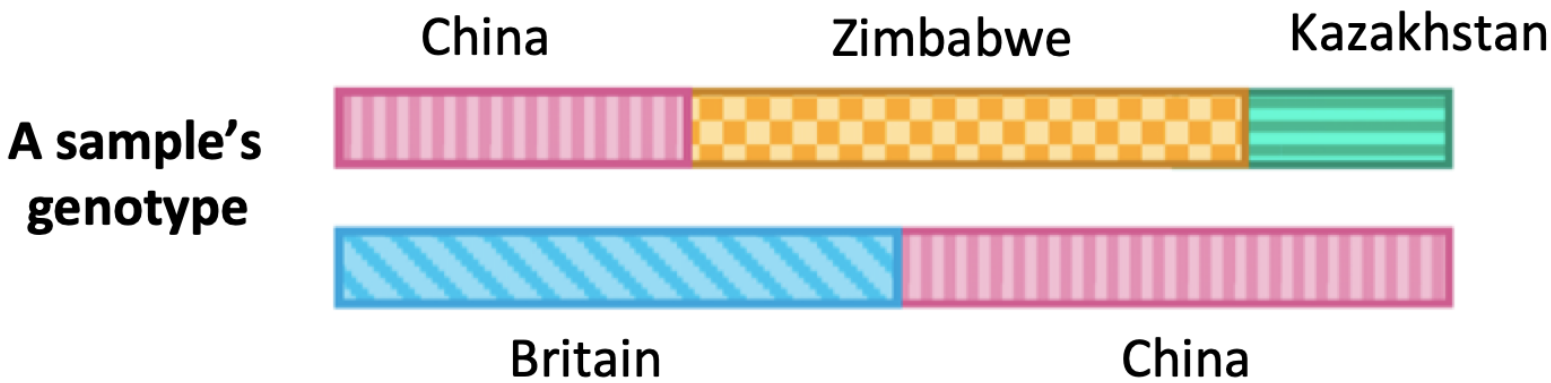
Comparison with HMM methods

1000G chr10	Error Rate	Time (sec)	Memory (GB)
`MendelImpute`	1.09E-02	39	3.7
`Beagle 5.1`	5.51E-03	196	9.6
`Minimac 4`	5.24E-03	728	10.2
1000G chr20	Error Rate	Time (sec)	Memory (GB)
`MendelImpute`	3.28E-02	13	2.6
`Beagle 5.1`	1.68E-02	33	4.9
`Minimac 4`	1.65E-02	159	5.0
HRC chr10	Error Rate	Time (sec)	Memory (GB)
`MendelImpute`	6.87E-03	154	7.3
`Beagle 5.1`	1.90E-03	1961	32.4
`Minimac 4`	1.71E-03	14604	22.5
HRC chr20	Error Rate	Time (sec)	Memory (GB)
`MendelImpute`	1.36E-03	133	6.2
`Beagle 5.1`	5.28E-04	2457	27.4
`Minimac 4`	6.34E-04	17507	33.2

Conclusion: 10-100x faster, 3-5x more memory efficient, 2-4x less accurate.

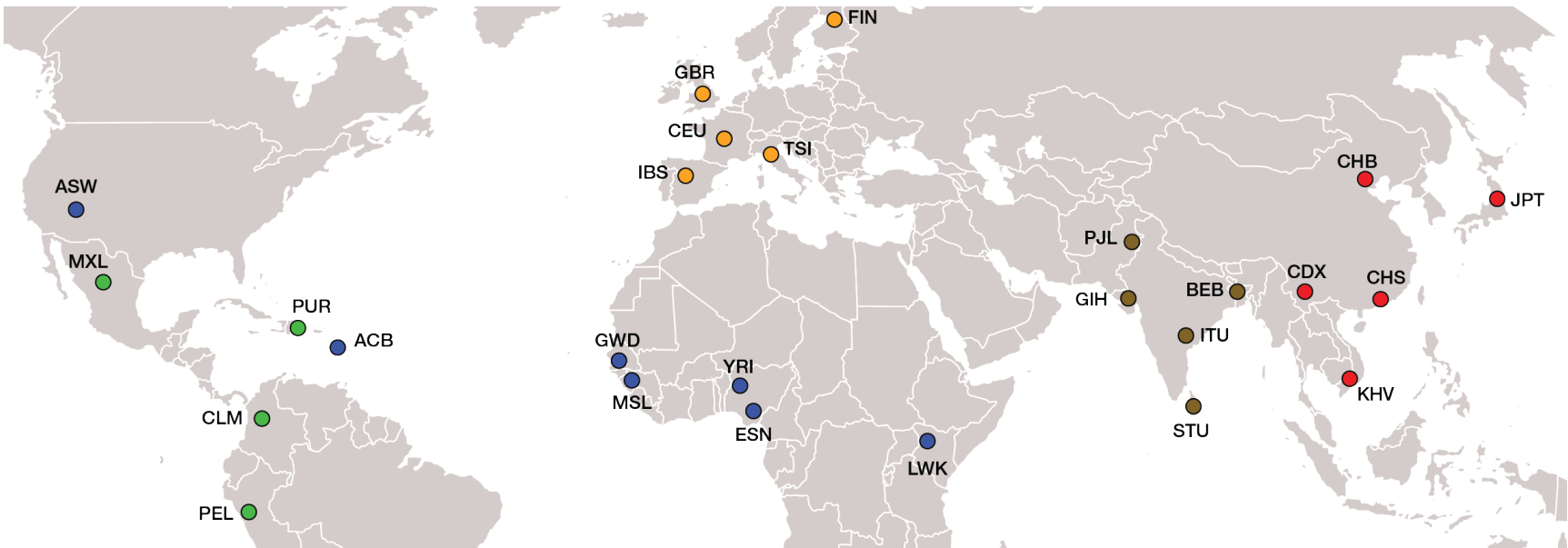
Extension to ancestry inference

- After imputation, each (unphased) genotype is decomposed into a mosaic of haplotype segments from the reference panel.
- If reference samples are labeled with ethnic or country origin, we can visualize local ancestry pattern. We call this chromosome painting.



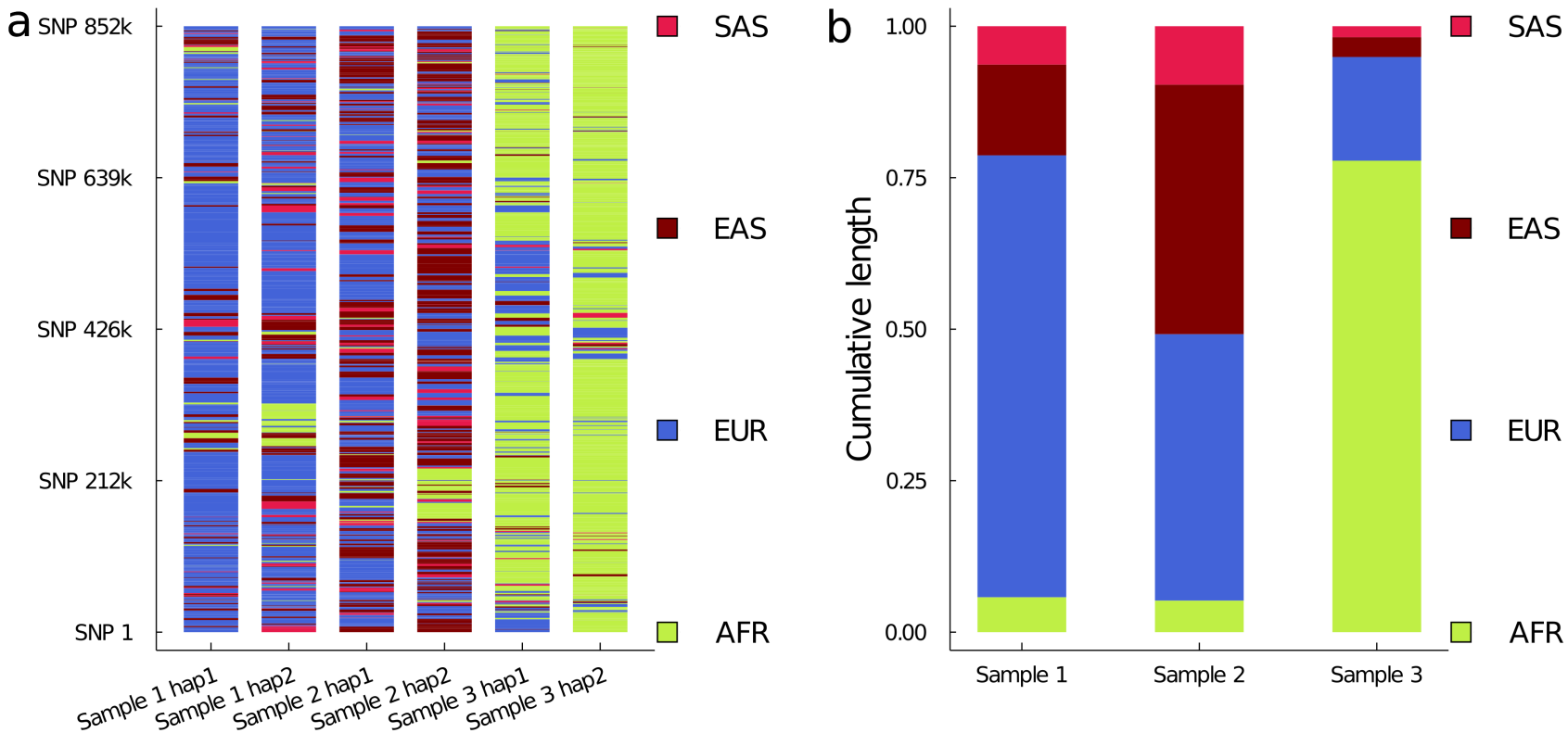
Test data: 1000 genomes project (v3)

- 2504 samples from 26 populations
- Many admixed, many not (4 grandparents all from local area)
- Does not include Native American populations



Source: https://gadget.biosci.gatech.edu/learn.html

Chromosome 18 painting on PUR, PEL, and ASW



For comparison, their ADMIXTURE estimates are

- PUR (Peurto Rican): 0.164589, 0.136001, 0.647827, 0.051582
- PEL (Peruvian): 0.803278, 0.175677, 1.0e-5, 0.021035
- ASW (African American): 1.0e-5, 0.140135, 1.0e-5, 0.859845

Data Compression

- Large genotype files are painful to transfer.
- Data compression can be achieved by phasing. Simply send:
  1. Start position for each haplotype segment
  2. Pointer to the relevant haplotype in the reference panel

Data Set	vcf.gz size (MB)	compressed size (MB)	compression ratio
Sim 10K	10.07	0.05	201
Sim 100K	10.71	0.04	267
Sim 1M	11.05	0.04	276
1000G Chr10	31.53	1.37	23
1000G Chr20	13.93	0.43	46
HRC Chr10	155.71	7.47	21
HRC Chr20	70.42	5.86	12

Table: Output file size comparison of VCF and our compression strategy.

Potential future work

1. Ancestry inference
    - Quantify linkage patterns
    - Promote new use for reference panels: deriving ancestry information
    - Construct different confidence level for different haplotype segments
2. `MendelImpute2`: Improve error rate
    - How to account for within-window breakpoints?
    - Implement overlapping windows strategy
    - Phasing step should account for ancestry origin
    - Parallel data import
3. Compare output haplotypes across programs

Part 2: Iterative hard thresholding in Genome wide association studies

Genome wide association studies

- We want to study genetic basis of a (usually continuous) phenotype
  - Examples: Height, developed cancer, number of seeds per plant
- Recruit n samples, genotype each sample at p SNPs, producing

$$\boldsymbol{X}^{n\times p} = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix}, \quad x_{ij} \in \{0, 1, 2\}$$

- Also measure each individual's phenotype

$$\boldsymbol{y}^{n\times 1} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

- Goal: which SNPs are associated with the phenotype?

Methods for Associations Testing

1. Naive model
   - $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim N(0, \sigma_e^2 \boldsymbol{I})$
   - Method: solve $\underset{\boldsymbol{\beta}}{\operatorname{argmin}} ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2$ with solution $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}\boldsymbol{y}$
   - Comments: There are ∞ solutions since $p > n$
2. Marginal association model
   - $\boldsymbol{y} = \boldsymbol{x}_j\beta_j + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim N(0, \sigma_e^2 \boldsymbol{I})$
   - Method: hypothesis testing for each SNP: $H_0 : \beta_j = 0, H_a : \beta_j \neq 0$
     - Score test, LRT, Wald test ...etc
     - Reject H_0 if $p < 5 \times 10^{-8}$
   - Comment: Assumes every SNP is independent of each other
3. Linear mixed model
   - $\boldsymbol{y} = \boldsymbol{x}_j\beta_j + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim N(0, \frac{\sigma_g^2}{p}\boldsymbol{X}\boldsymbol{X}^t + \sigma_e^2 I)$
   - Method: hypothesis testing for each SNP: $H_0 : \beta_j = 0, H_a : \beta_j \neq 0$
   - Comment: Hard to estimate σ_g, σ_e. Expensive to form $\boldsymbol{X}\boldsymbol{X}^t$. Have to calculate determinants and solve huge linear equations. Current analysis methods scale poorly for non-Gaussian phenotypes.
4. Penalized regression

Option 4: Penalized regression methods

Minimize the original model $\ell(\boldsymbol{\beta}) = ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2$ plus constraints $p(\boldsymbol{\beta}, \boldsymbol{\lambda})$.

- **lasso**: minimize $f(\boldsymbol{\beta}) = \ell(\boldsymbol{\beta}) + \lambda||\boldsymbol{\beta}||_1$
- **MCP**: minimize $f(\boldsymbol{\beta}) = \ell(\boldsymbol{\beta}) + \sum_{i=2}^{p} q(|\beta_j|)$ where

$$q(\beta_i) = \begin{cases} \lambda\beta_i - \beta_i^2/(2\lambda) & 0 \leq \beta_i \leq \gamma\lambda \\ \gamma\lambda^2/2 & \beta_i > \gamma\lambda \end{cases}$$

- **Iterative hard thresholding (IHT):**

$$\text{minimize} \quad \ell(\boldsymbol{\beta}) \quad s.t. \quad ||\boldsymbol{\beta}||_0 \leq k.$$

Shrinkage of lasso tend to encourage too many false positives. The ℓ_0 norm of IHT enforces sparsity without shrinkage.

IHT background

Problem: minimizes $f(\boldsymbol{\beta}) = ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 \quad s.t. \quad ||\boldsymbol{\beta}||_0 \leq k \in \mathbb{Z}.$

Solution: IHT iterates via

$$\boldsymbol{\beta}_{n+1} = \overbrace{P_{S_k}}^{(3)}\big(\boldsymbol{\beta}_n - \underbrace{s_n}_{(2)}\overbrace{\nabla f(\boldsymbol{\beta}_n)}^{(1)}\big).$$

1. Compute gradient $\nabla f(\boldsymbol{\beta}) = -\boldsymbol{X}^t(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})$
2. Compute step size $s = \frac{||\nabla f(\boldsymbol{\beta})||^2}{\nabla f(\boldsymbol{\beta})^t J(\boldsymbol{\beta}) \nabla f(\boldsymbol{\beta})} > 0$
3. Project to sparsity by setting all but k largest entries to 0.

Generalized linear models for non-Gaussian phenotypes

For non-Gaussian phenotypes (e.g. $y_i \in \{0, 1\}$) we can model y_i as generalized linear models (GLM)

$$\mu_i = \mathrm{E}(y_i) = g(\boldsymbol{x}_i^t \boldsymbol{\beta})$$

where g is a nonlinear inverse link function. The loglikelihood, score (gradient), and expected information (expected negative Hessian) is

$$L(\boldsymbol{\beta}) = \sum_{i=1}^n \left[\frac{y_i \theta_i - b(\theta_i)}{a(\phi_i)} + c(y_i, \phi_i) \right] \approx \sum_{i=1}^n \left[\frac{y_i g(\boldsymbol{x}_i^t \boldsymbol{\beta}) - b[g(\boldsymbol{x}_i^t \boldsymbol{\beta})]}{a(\phi_i)} \right]$$

$$\nabla L(\boldsymbol{\beta}) = \sum_{i=1}^n (y_i - \mu_i) \frac{b''[g(\boldsymbol{x}_i^t \boldsymbol{\beta})] g'(\boldsymbol{x}_i^t \boldsymbol{\beta})}{\sigma_i^2} \boldsymbol{x}_i^t = \boldsymbol{X}^t \boldsymbol{W}_1 (\boldsymbol{y} - \boldsymbol{\mu})$$

$$J(\boldsymbol{\beta}) = \sum_{i=1}^n \frac{b''[g(\boldsymbol{x}_i^t \boldsymbol{\beta})]^2}{\sigma_i^2} g'(\boldsymbol{x}_i^t \boldsymbol{\beta}) \boldsymbol{x}_i \boldsymbol{x}_i^t = \boldsymbol{X}^t \boldsymbol{W}_2 \boldsymbol{X}$$

where y_i has mean $\mu_i = b'[g(\boldsymbol{x}_i^t \boldsymbol{\beta})]$ and variance $\sigma_i^2 = b''[g(\boldsymbol{x}_i^t \boldsymbol{\beta})] a(\phi_i)$. These provide all the necessary ingredients for running IHT on GLMs.

We also did a few more things...

- Derivation for optimal step size s_n for each IHT iteration
- Doubly sparse projection for sparsity within and between groups.
- Incorporation of prior weights.
- Block estimation for nuisance parameters

Simulation study: IHT vs lasso vs marginal testing

True Positives: higher is better. **False Positives**: lower is better

	Normal	Logistic	Poisson	Neg Bin
IHT True Positives	8.84	6.28	7.2	9.0
IHT False Positives	0.02	0.1	1.28	0.98
Lasso True Positives	9.52	8.16	9.28	NA
Lasso False Positives	31.26	45.76	102.24	NA
Marginal True Positives	7.18	5.76	9.04 (5.94)	5.98
Marginal False Positives	0.06	0.02	1527.9 (0.0)	0.0

Table: IHT achieves the best balance of false positives and true positives compared to lasso and marginal (single-snp) regression. TP = true positives, FP = false positives. There are $k = 10$ causal SNPs. Best model size for IHT and lasso were chosen by cross validation. () = zero-inflated Poisson regression.

Logistic GWAS on UK Biobank hypertension phenotypes

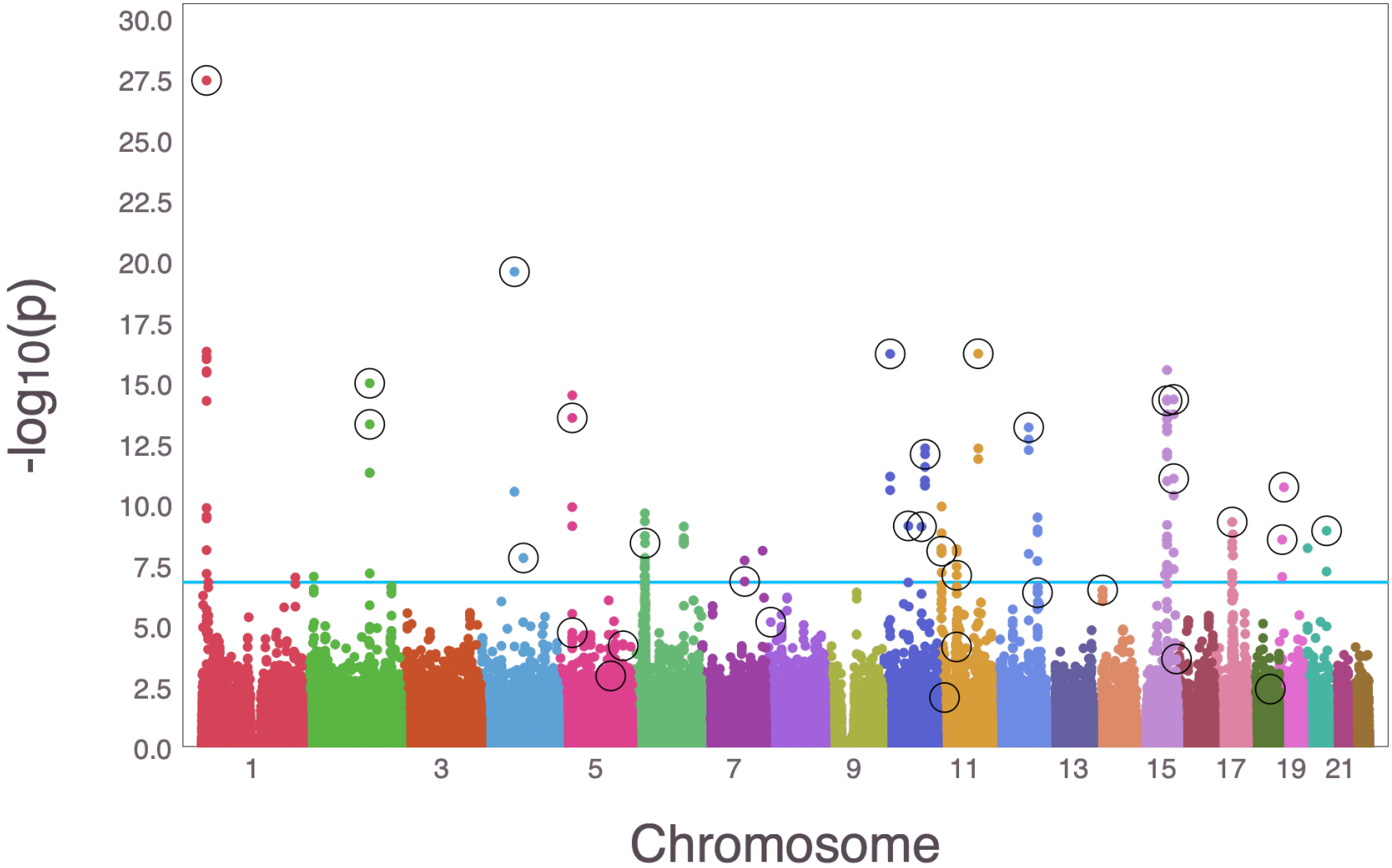


Figure: Manhattan plot comparing a logistic (univariate) GWAS vs logistic IHT on UK Biobank data ($n \approx 200,000$ and $p \approx 500,000$). Colored dots are $\log_{10}$ p-values from a logistic GWAS, and the circled dots are SNPs recovered by IHT.

IHT related ongoing and potential future work

1. Multivariate Gaussian IHT - each sample measures m (potentially correlated) phenotypes $\boldsymbol{y}_i^t = [y_{i1}, ..., y_{im}]$ **(ongoing)**
2. Heritability estimates using IHT: $var(\hat{\boldsymbol{y}})/var(\boldsymbol{y})$. Since SNPs are estimated as mean effects, IHT will likely outperform mixed-model estimates whose SNPs are in the variance **(potential project)**
3. Survival IHT - each sample i has time-to-event measurement y_i and indicator I_i for event **(potential project)**
4. Structured IHT - Each sample's phenotype y_i take ordered, discrete values. **(potential project)**
5. IHT vs transformed linear mixed model - Compare current IHT to mixed model approach where phenotypes are transformed to normal **(potential project)**

Thank you!!

Visit us at: `https://github.com/OpenMendel/`

Figure: Weekly meetings from the OpenMendel group

References

1. Chu, B. B., Keys, K. L., German, C. A., Zhou, H., Zhou, J. J., Sobel, E. M., ... & Lange, K. (2020). Iterative hard thresholding in genome-wide association studies: Generalized linear models, prior weights, and double sparsity. *GigaScience*, 9(6), giaa044.
2. Chu, B. B., Sobel, E., Wasiolek, R., Sinsheimer, J. S., Zhou, H., & Lange, K. (2020). A Fast Data-Driven Method for Genotyp Imputation, Phasing, and Local Ancestry Inference: MendelImpute. jl. *bioRxiv*.
3. Noah Zaitlen (2018) A Short Tutorial on Linear Mixed Model Association Testing in Genetics.
`https://www.youtube.com/watch?v=pTAXVTA0YQQ`