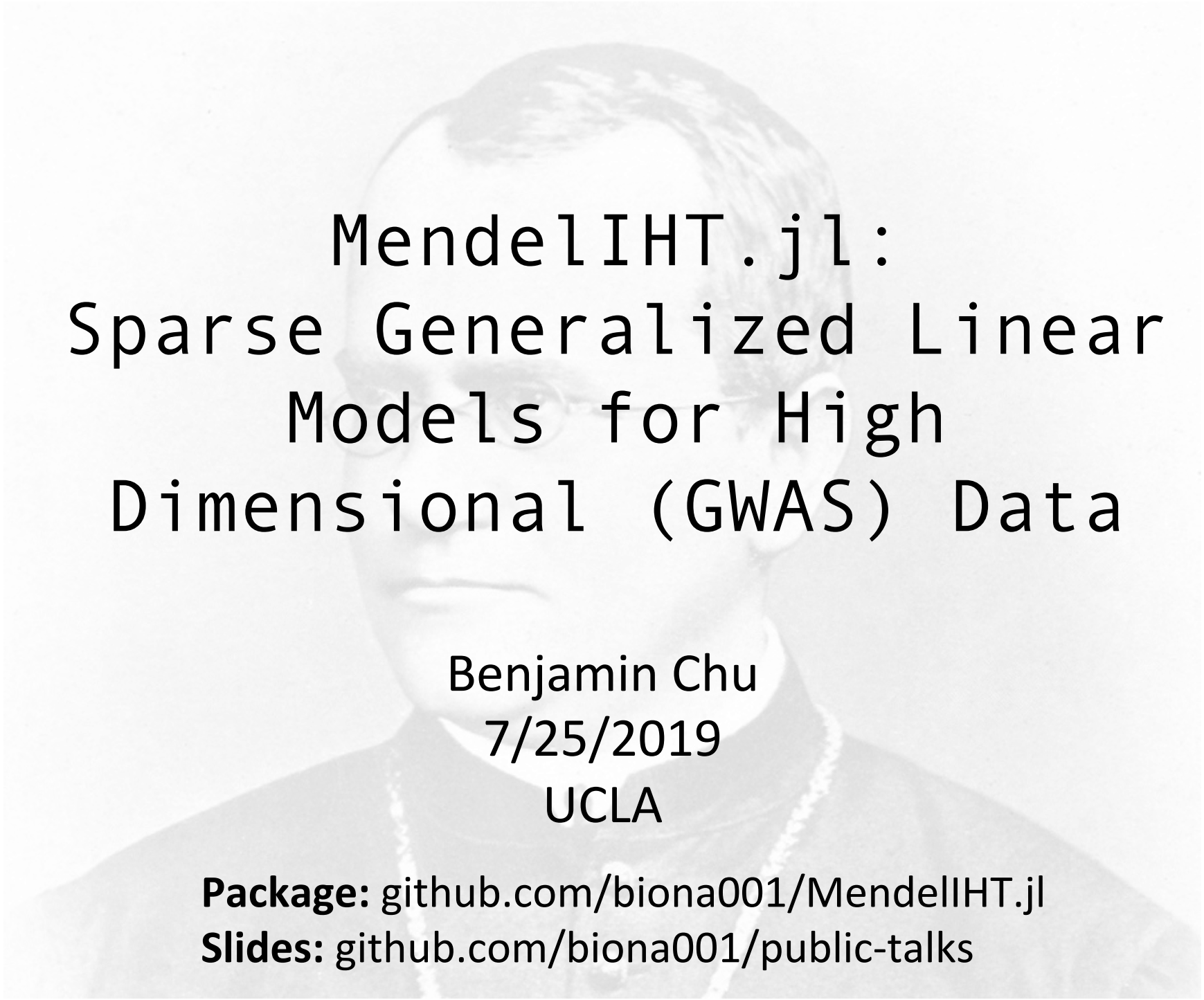

MendelIHT.jl: Sparse Generalized Linear Models for High Dimensional (GWAS) Data

Benjamin Chu

7/25/2019

UCLA

Package: github.com/biona001/MendelIHT.jl

Slides: github.com/biona001/public-talks

OpenMendel: Statistical Genetics Research

Visit us: github.com/OpenMendel

OpenMendel Option	Description
`MendelAimSelection.jl`	Selects the most informative SNPs for predicting ancestry
`MendelEstimateFrequencies.jl`	Estimates allele frequencies from pedigree data
`MendelGameteCompetition.jl`	Tests for association under the gamete competition model
`MendelGeneticCounseling.jl`	Computes risks in genetic counseling problems
`MendelGWAS.jl`	Tests for association in genome-wide data
`MendelIHT.jl`	GWAS using Iterative Hard Thresholding (forthcoming)
`MendelImpute.jl`	Genotype imputation (forthcoming)
`MendelKinship.jl`	Computes kinship and other identity coefficients
`MendelLocationScores.jl`	Maps a trait via the method of location scores
`OrdinalGWAS.jl`	Implements GWAS for ordinal categorial phenotypes
`MendelTwoPointLinkage.jl`	Implements two-point linkage analysis
`MendelBase.jl`	Base functions for OpenMendel
`MendelGeneDropping.jl`	Simulates genotypes based on pedigrees
`MendelSearch.jl`	Optimization routines
`MendelTraitSimulate.jl`	Trait simulation using GLM and GLMM (forthcoming)
`SnpArrays.jl`	Utilities for handling compressed storage of biallelic SNP data
`VCFTools.jl`	Utilities for handling compressed storage of sequence data
`VarianceComponentModels.jl`	Utilities for fitting and testing variance components models

Human Genetics (2019): https://link.springer.com/article/10.1007/s00439-019-02001-z

Generalized Linear Models Regression

- Given *dense* $\mathbf{y}, \mathbf{X}$ find *sparse* β such that:

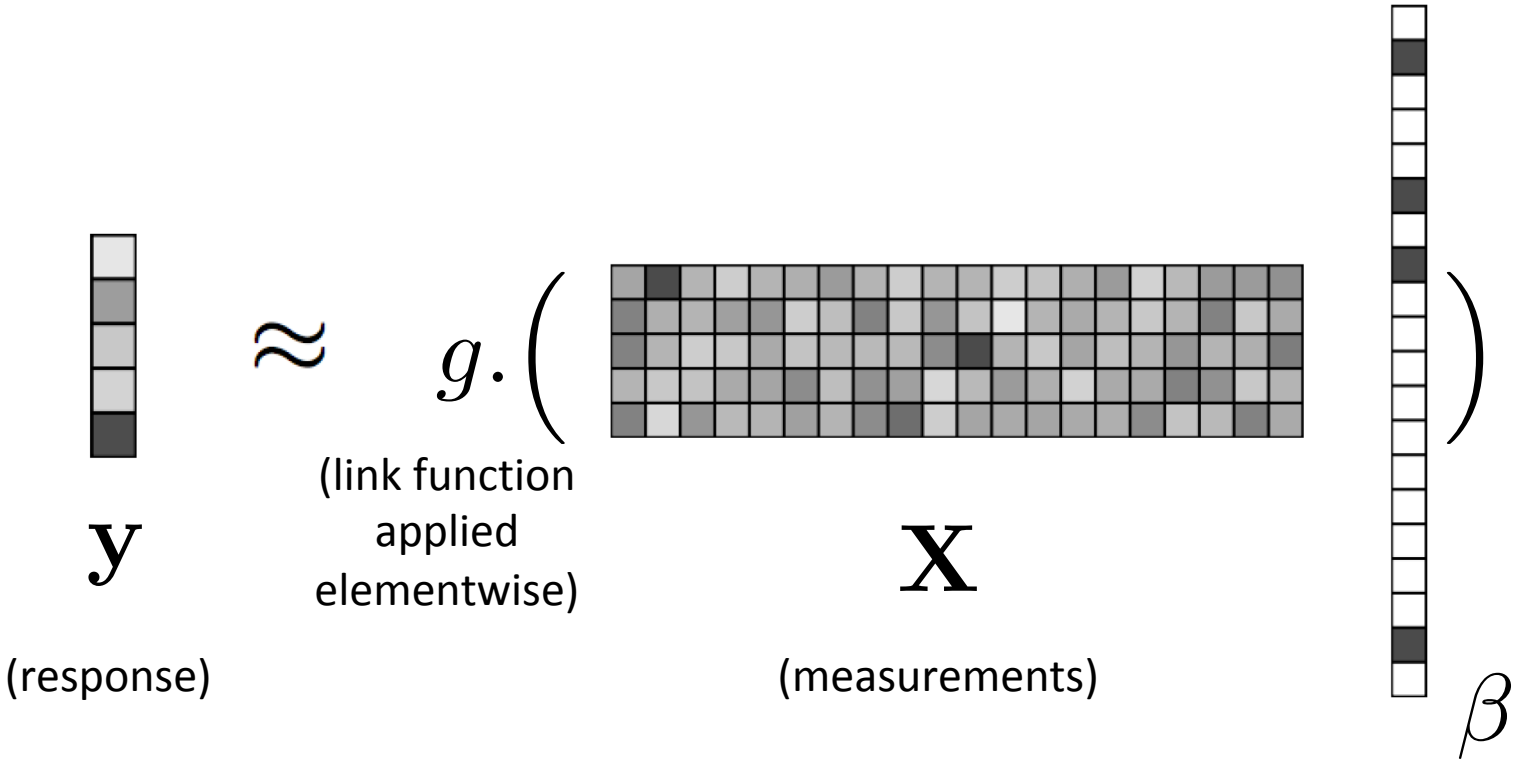

Figure: https://www.cs.cmu.edu/~ggordon/10725-F12/scribes/10725_Lecture2.pdf

Generality & Existing Methods

- **Examples**: Normal, Poisson, multinomial, gamma, negative binomial, inverse Gaussian, Cox…
- **Solutions**: *iterative hard-thresholding* (MendelIHT.jl), lasso, marginal regression, elastic net, MCP, stepwise regression…

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

TP = true positives (higher means good)

FP = false positives (lower means good)

See our paper: https://www.biorxiv.org/content/10.1101/697755v1

Example: Logistic regression

Consider $\mathbf{y} \in \{0,1\}^{1000\times 1}, \mathbf{X} \in \mathbb{R}^{1000\times 10000}$ where

$$
\begin{aligned}
X_{ij} &\sim N(0,1) \\
\beta_i &\sim N(0,1) \\
y_i &\sim \text{Bernoulli}(p_i) \\
p_i &= \text{logit}^{-1}(\mathbf{x}_i^t \beta) \\
Z &= \mathbf{1} \text{ (intercept)} \\
k &= 10 \text{ (how many non-zero entries)}
\end{aligned}
$$

Solve using MendelIHT:

g()↓

```
julia> result = L0_reg(X, Z, y, 1, k, Bernoulli(), LogitLink())
```

Example: Logistic regression (cont.)

```
Compute time (sec):      0.18402791023254395
Final loglikelihood:     -338.65497510724015
Iterations:              36
Max number of groups:    1
Max predictors/group:    10
```

Row	true_β	estimated_β
	Float64	Float64
1	-0.20377	0.0
2	0.887386	0.987261
3	0.0499892	0.0
4	-0.0954796	0.0
5	0.270736	0.400961
6	0.197487	0.0
7	-2.23291	-2.27334
8	-0.92738	-0.829479
9	-0.848911	-0.78015
10	-1.35671	-1.31519

Unbiased parameter estimates

Recall: MendelIHT is designed for GWAS

- Memory management is key in bioinformatics.
- $\mathbf{X}$ can have $\sim 10^6$ samples and $\sim 10^7$ covariates.
  - Largest problem: **80 TB** (Float64)
  - Medium problem: **400 GB** (Float64)
  - Small problem: **12 GB** (Float64)
- Difficult even with cloud/cluster resources.
- **Solution**: invoke *SnpArrays.jl* for compressed linear algebra routines.

Fast Linear Algebra with Compressed Data

- Since $X_{ij} \in \{0, 1, 2\}$, we store $\mathbf{X}$ as 2 bitarrays.
  - Less vulnerable to data swapping with the hard disk.
  - Faster than BLAS **even without multi-threading**

Matrix Size	BLAS (multithreaded)	SnpArrays (single thread)	# of Times Faster
1.2 GB	0.080 s	0.663 s	0.12
12 GB	3.966 s	5.612 s	0.71
24 GB	57.073 s	11.227 s	5.08
48 GB	175.689 s	26.540 s	6.62

Matrix-vector multiplication benchmarked on 2019 macbook pro with 16GB of RAM, equipped with 8 cores, each are 9th generation i9, 2.3 GHz CPUs

Conclusions

- IHT is better than lasso and marginal tests
- MendelIHT works for generalized linear model regression problems
- MendelIHT is especially powerful for GWAS due to memory savings from compressed storage

Acknowledgements

- JuliaCon 2019 for Travel Assistence
- Google Summer of Code 2018
- NIH T32-HG002536 (GATP) training grant

References

- Package: github.com/biona001/MendelIHT.jl
- Chu et al. *Multivariate GWAS, Generalized Linear Models, Prior Weights, and Double Sparsity.* BioRxiv Preprint: 10.1101/697755
- Keys et al. *Iterative hard thresholding for model selection in genome-wide association studies.* Genetic Epidemiology 2017;41:756–768.
- Zhou et al. *OpenMendel: a cooperative programming project for statistical genetics.* Human Genetics 2019;p. 1–11.