BulkLMM

Real-time Linear Mixed Model Applications for Association Mapping on Large Numbers of Quantitative Traits

Speaker: Zifan (Fred) Yu

Graduate Student of Data Science and Engineering

The Bredesen Center, UT Knoxville

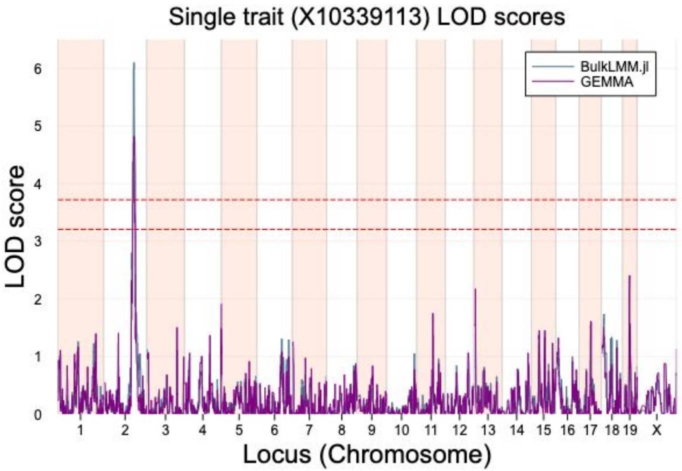
The Department of Preventive Medicine, UT Health Science Center

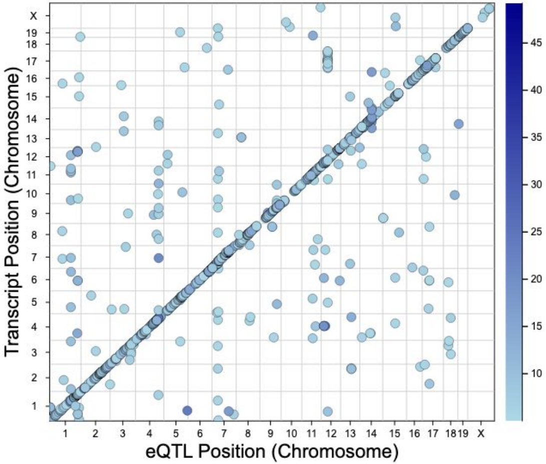
We will discuss...

- Our Design Goals of BulkLMM
- Overview of Methods
- Performance
- Discussion

What is BulkLMM?

BulkLMM.jl is a *Julia* package to perform **fast** genome scans of **over large numbers of quantitative traits** using linear mixed models. It is available on GitHub at https://github.com/senresearch/BulkLMM.jl

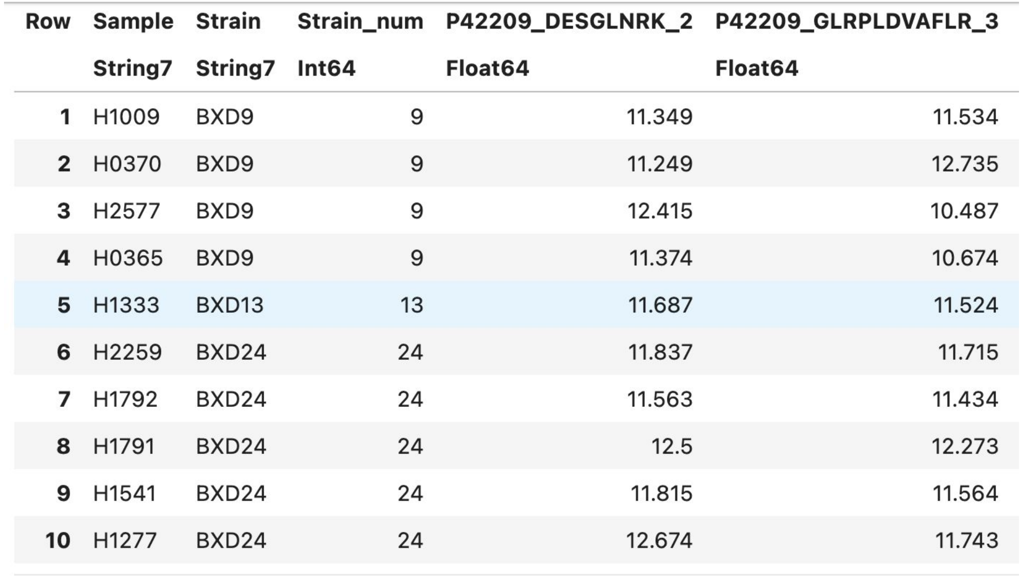

Single trait (X10339113) LOD scores



Motivating data

BXD Longevity Study

of Individual Liver Proteome



Row	Sample	Strain	Strain_num	P42209_DESGLNRK_2	P42209_GLRPLDVAFLR_3
	String7	String7	Int64	Float64	Float64
1	H1009	BXD9	9	11.349	11.534
2	H0370	BXD9	9	11.249	12.735
3	H2577	BXD9	9	12.415	10.487
4	H0365	BXD9	9	11.374	10.674
5	H1333	BXD13	13	11.687	11.524
6	H2259	BXD24	24	11.837	11.715
7	H1792	BXD24	24	11.563	11.434
8	H1791	BXD24	24	12.5	12.273
9	H1541	BXD24	24	11.815	11.564
10	H1277	BXD24	24	12.674	11.743

Data information:

- 248 samples, 50 BxD strains
- 7321 measured genetic markers
- **32445** liver proteome

Overview of our methods

Statistical Framework

Standard Linear Mixed Model (LMM) - notation from Henderson (1984)

$$y = X_0\beta_0 + X_g\beta_g + Zu + \epsilon$$

$$\text{assume } u \sim N_{q\times 1}(0, \sigma_g^2 K_g), \;\; \epsilon \sim N_{n\times 1}(0, \sigma_e^2 I)$$

Notations:

$y_{n\times 1}$ - a vector of a quantitative gene expression trait

β_g, β_0 - fixed marker (β_g) and non-marker effects (β_0)

$u_{q\times 1}$ - a vector of random polygenic effects with genetic variance σ_g^2

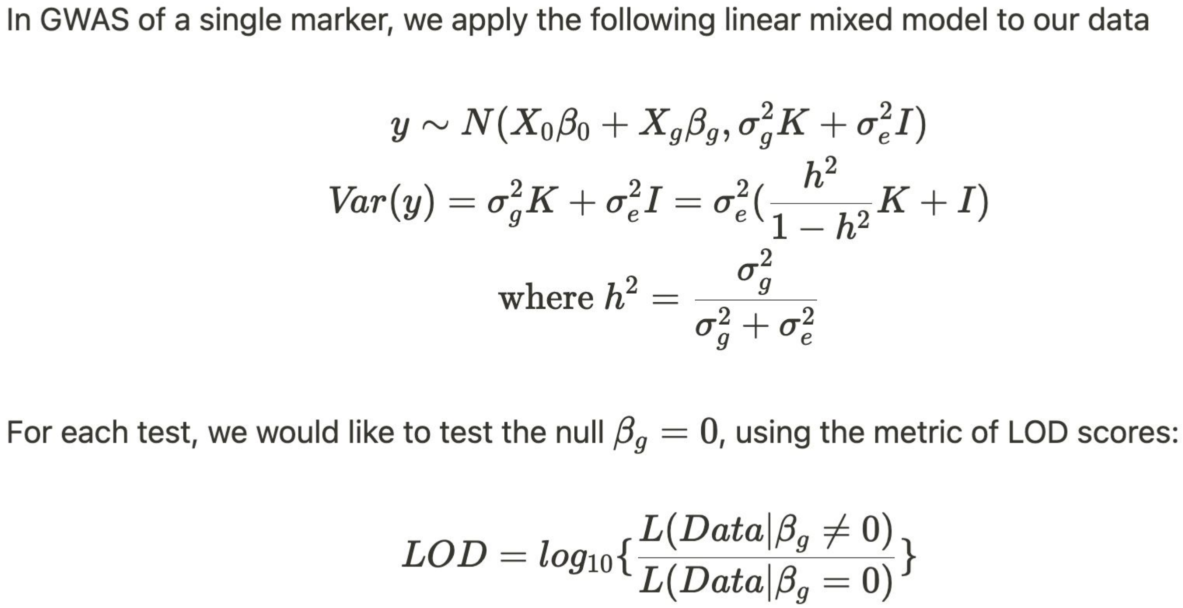
$\epsilon_{n\times 1}$ - a vector of residual errors with unexplained variance σ_e^2

X_0, X_g, Z are the design matrices for effects β_0, β_g, u

K_g is the kinship matrix with element $k_{i,j}$ representing pairwise genetic relatedness

Statistical Framework

Linear Mixed Model (LMM):



In GWAS of a single marker, we apply the following linear mixed model to our data

$$y \sim N(X_0\beta_0 + X_g\beta_g, \sigma_g^2 K + \sigma_e^2 I)$$

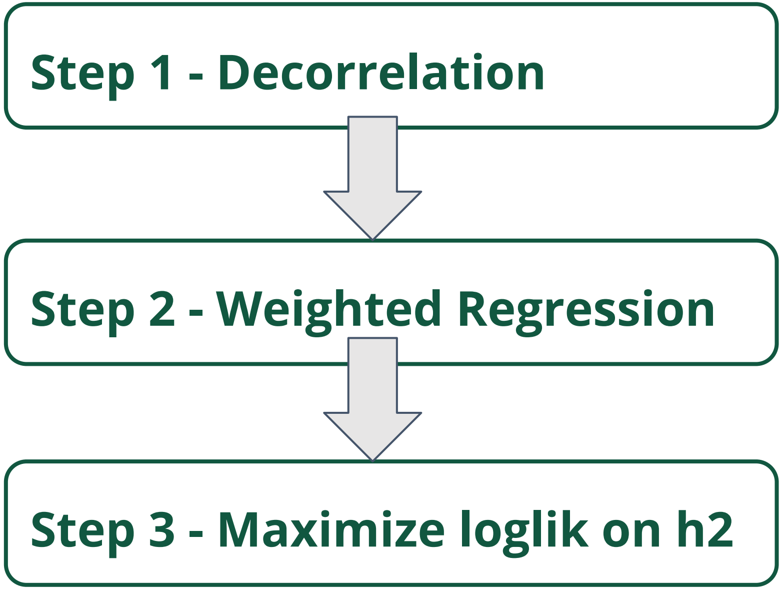
$$Var(y) = \sigma_g^2 K + \sigma_e^2 I = \sigma_e^2 (\frac{h^2}{1-h^2} K + I)$$

$$\text{where } h^2 = \frac{\sigma_g^2}{\sigma_g^2 + \sigma_e^2}$$

For each test, we would like to test the null $\beta_g = 0$, using the metric of LOD scores:

$$LOD = log_{10}\{\frac{L(Data|\beta_g \neq 0)}{L(Data|\beta_g = 0)}\}$$

Evaluating the LMM



Step 1 - Decorrelation

- Decompose K as $K = UDU^T$
- Apply the transformation:

  $y^* = U^T y,\ X^* = U^T X$
- $y^* \sim N(\ X^*\beta, \sigma_e^2(\delta D + I)\ ), \delta = \frac{h^2}{1-h^2}$

Step 2 - Weighted Regression

- For a given h^2, we construct $W = [(\delta\lambda_i + 1)^{-1}]_{i=1}^{n}$
- Apply the transformation:

  $y^\dagger = Wy^*,\ X^\dagger = WX^*$
- $y^\dagger \sim N(\ X^\dagger\beta,\ \sigma_e^2 I\ )$

Step 3 - Maximize loglik on h2

- After getting the OLS solutions $\hat{\beta}(h^2),\ \hat{\sigma}_e^2(h^2)$, plug them back in the log-likelihood
- Perform any numerical method to optimize $l(y^\dagger|h^2)$ on h^2

Computational speed-up methods

Fast calculation of LOD scores

For simple linear regression...

$$
\begin{aligned}
LOD_{ij} &= -\frac{n}{2}log_{10}(\frac{RSS_{1ij}}{RSS_{0i}}) \\
&= \frac{n}{2}log_{10}(\frac{RSS_{0j}}{RSS_{1ij}}) \\
&= -\frac{n}{2}log_{10}(1-r_{ij}^2)
\end{aligned}
$$

Q: Can we apply this convenient fact to LMMs?

As we could calculate R as...

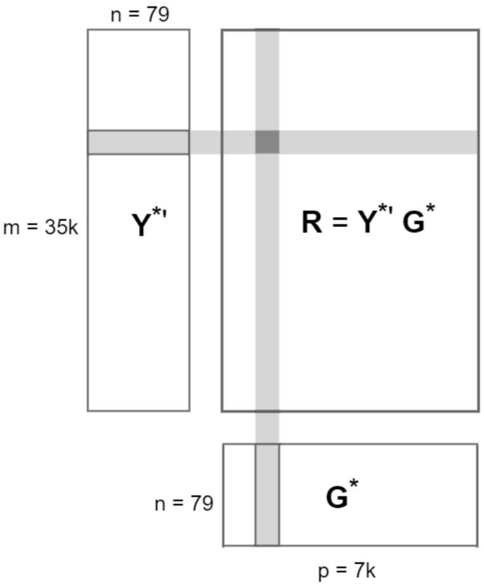
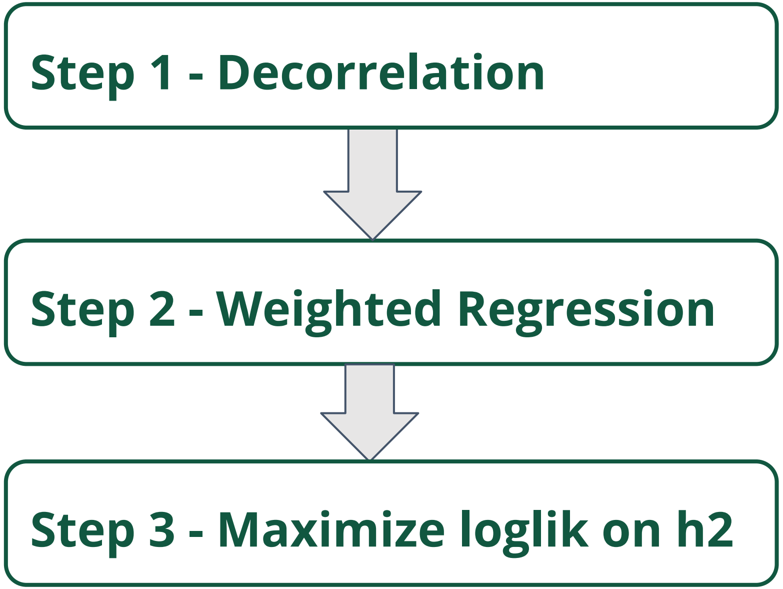


Figure borrowed from *Trotter et al.(2021)* https://doi.org/10.1093/g3journal/jkab254

Recall: Evaluating the LMM



Step 1 - Decorrelation

- Decompose K as $K = UDU^T$
- Apply the transformation:

  $y^* = U^T y,\ X^* = U^T X$
- $y^* \sim N(\ X^*\beta, \sigma_e^2(\delta D + I)\ ), \delta = \frac{h^2}{1-h^2}$

Step 2 - Weighted Regression

- For a given h^2, we construct $W = [(\delta\lambda_i + 1)^{-1/2}]_{i=1}^n$
- Apply the transformation:

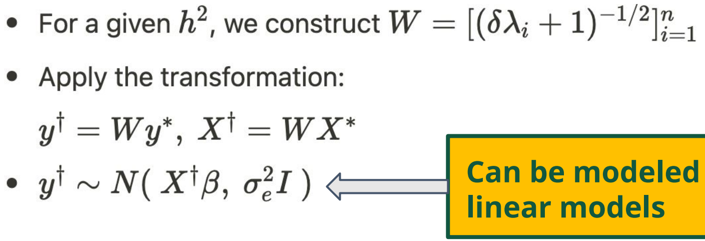
  $y^\dagger = Wy^*,\ X^\dagger = WX^*$
- $y^\dagger \sim N(\ X^\dagger\beta,\ \sigma_e^2 I\ )$

Step 3 - Maximize loglik on h2

- After getting the OLS solutions $\hat{\beta}(h^2),\ \hat{\sigma}_e^2(h^2)$, plug them back in the log-likelihood
- Perform any numerical method to optimize $l(y^\dagger|h^2)$ on h^2

Applying the trick to LMM

Step 2 - Weighted Regression



- For a given h^2, we construct $W = [(\delta\lambda_i + 1)^{-1/2}]_{i=1}^{n}$
- Apply the transformation:

  $y^\dagger = Wy^*,\ X^\dagger = WX^*$
- $y^\dagger \sim N(\ X^\dagger\beta,\ \sigma_e^2 I\ )$

Can be modeled as linear models

In order to get to the point of evaluating on the transformed y “dagger”, **the key is to get the heritability estimate.**

Some important observations:

1. If we don’t assume heritability differ by marker (“LMM-exact”), but **can estimate h2 once from the null model, then we can apply the same W to test all markers** (“LMM-null”)

2. Moreover, suppose more than one traits **have the same h2 estimated from null, we can group them as columns in a matrix, and use a common W** to compute the LOD scores together…

Bulkscan-Null-Grid

Extended from the "LMM-null" simplification, we may further take the shortcut, by estimating the h2 under the null **using a grid-search approach.**

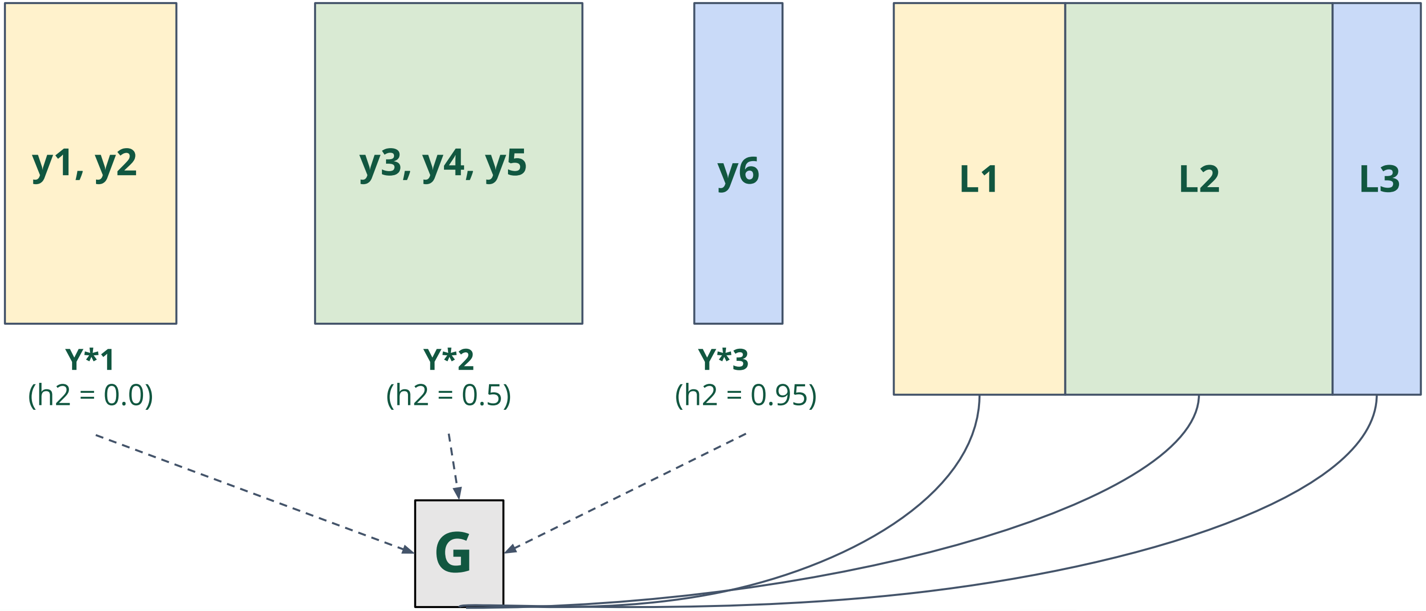
This has two benefits:

- We omitted the numerical optimization which may take longer to converge.
- More importantly, with a finite number of candidate values for the h2's for a large number of traits, **it is more likely that more than one traits will share the same heritability**

We can then group traits with the same h2 to calculate the LOD scores in one matrix multiplication!

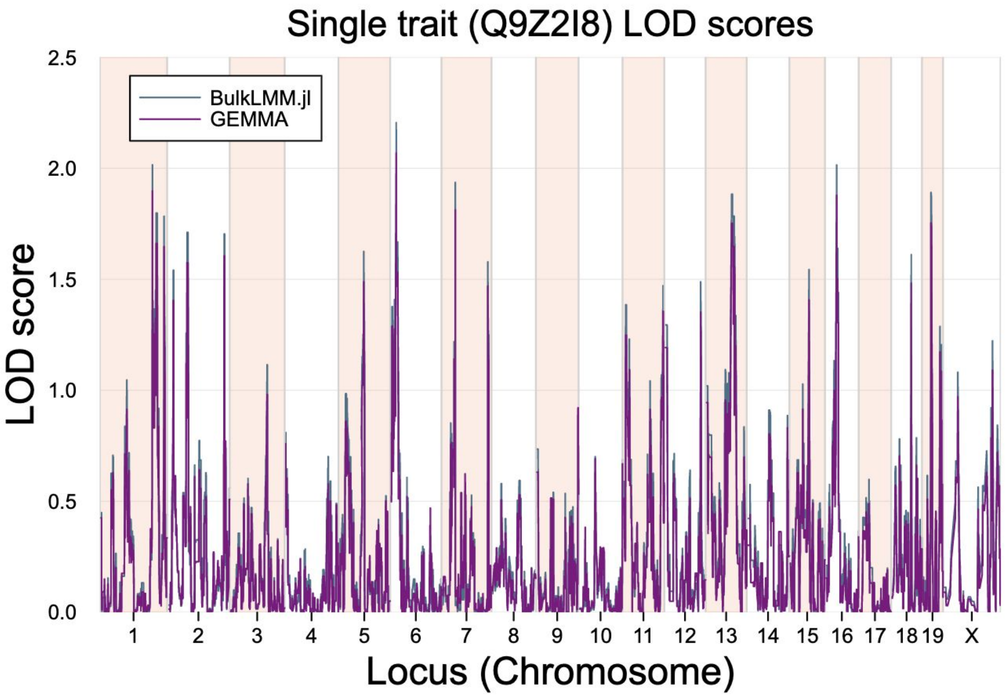
Bulkscan-Null-Grid

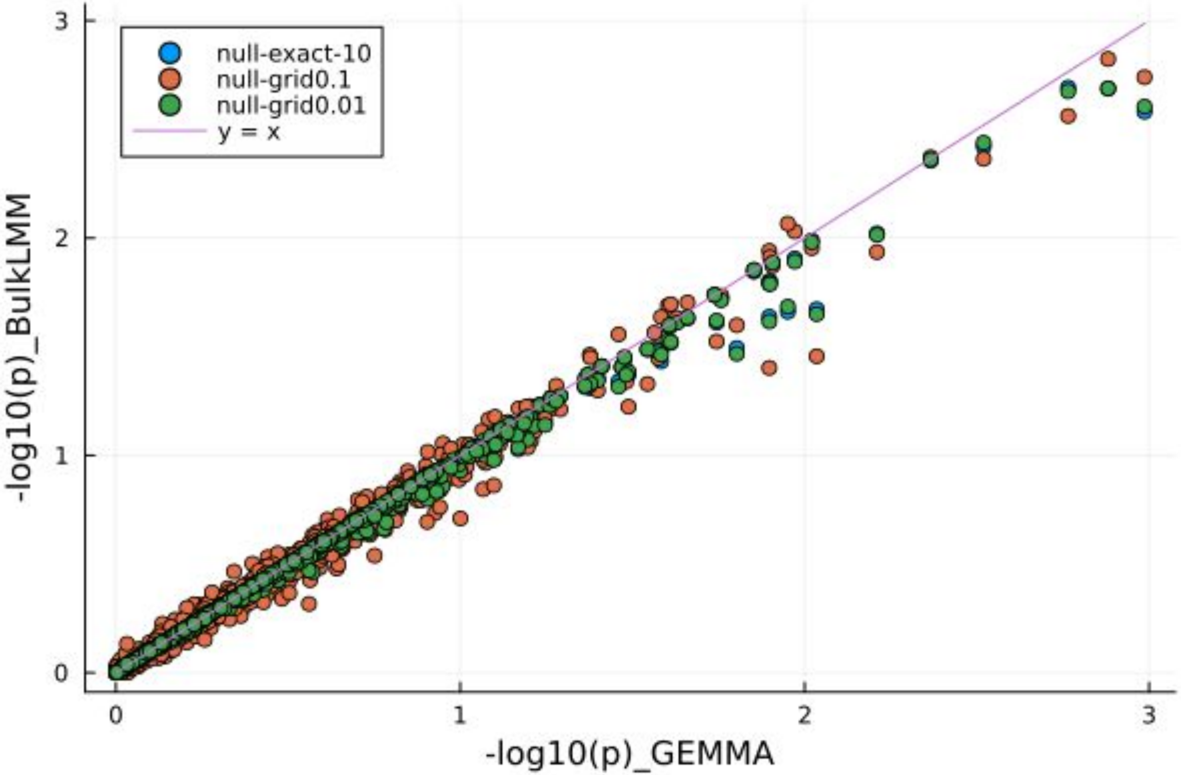
Y = (y1, y2, y3, y4, y5, y6), and we used 3 candidate h2's on the grid **[0.0, 0.5, 0.95]...**

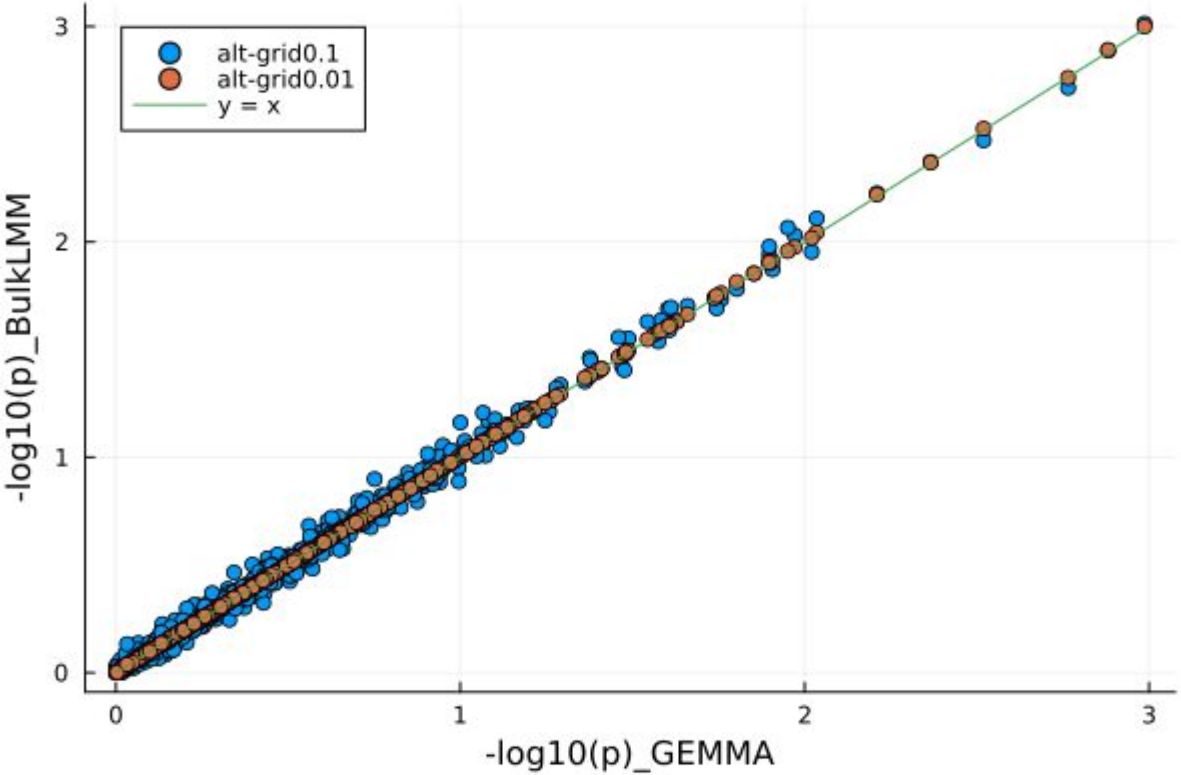


Results & Performance

QTL plot of trait Q972I8




3
2
1
0
-log10(p)_BulkLMM
0
1
2
3
-log10(p)_GEMMA
● null-exact-10
● null-grid0.1
● null-grid0.01
y = x



alt-grid0.1
alt-grid0.01
y = x
-log10(p)_BulkLMM
-log10(p)_GEMMA
0
1
2
3
0
1
2
3

Performance (compare with GEMMA)

Method	Runtime (s)	Error (from GEMMA)
Null-Exact	~ 110	0.0094
Null-Grid (h2-grid: 0.1 / 0.01)	**~ 3.6 / 18**	0.018 / 0.0096
Alt-Grid (h2-grid: 0.1 / 0.01)	~ 50 / 460	**0.011 / 0.00097**
GEMMA (Alt-Exact)	~ 70k	–/–

Details of the experiments:

- BXD Data: n = 248, p = 7321, **m = 32554**
- Environment: Intel(R) Xeon(R) Silver 4214 CPU @ 2.20GHz (24 cores); Julia 1.9.2 with 24 threads
- To compare with GEMMA, we run GEMMA iteratively on 1000 randomly selected traits and scale by m/1000
- Errors are based on mean absolute difference over the 7321 LOD scores for the 1000 selected traits

Performance (compare with GEMMA)

Method	Runtime (s)	Error (from GEMMA)
Null-Exact	Slow when n, p are large	Accurate when n is large
Null-Grid (h2-grid: 0.1 / 0.01)	Fastest	Accurate as n is large
Alt-Grid (h2-grid: 0.1 / 0.01)	Slow when p or h2-grid is large	Most accurate
GEMMA (Alt-Exact)	~ 70k	–/–

Details of the experiments:

- BXD Data: n = 248, p = 7321, **m = 32554**
- Environment: Intel(R) Xeon(R) Silver 4214 CPU @ 2.20GHz (24 cores); Julia 1.9.2 with 24 threads
- To compare with GEMMA, we run GEMMA iteratively on 1000 randomly selected traits and scale by m/1000
- Errors are based on mean absolute difference over the 7321 LOD scores for the 1000 selected traits

Discussion

Discussions

Strengths:

- BulkLMM is **fast** for scanning **a large number of traits** without losing too much accuracy
- Integration in Julia:
  - Easy to code, intuitive syntax
  - Flexible (CPU configuration, multi-dispatch)
- Great for downstream manipulation and analysis

Other Features:

- Efficient permutation testing framework
- MAP optimization of h2 when including conjugate prior on residual variances
- Weighted residual variances

Discussions

Limitations:

- The key improvement in runtime relies on doing **1-df tests**
- Nature of **large m**, **low sample size n** (hard to measure many traits on a lot of individuals)
- Accurate methods may require large memory when data size is large
- Can not deal with more than two variance components (more than one sources of the random effects)

Future steps:

- Command line version, integration to other languages
- Applications for studying strain means v.s. individual measurements
- Publishing the paper

Acknowledgements

Senresearch group:

Śaunak Sen - Principal Investigator of the project, ground work, primary guidance

Gregory Farage - Main co-author, code review and refinement, package development, front-end integration

Harper Kolehmainen - Intern of summer 2023, front-end integration and downstream analysis

Special Thanks to:

GN group: for providing and assisting with accessibility to data, use of GEMMA

Robert W. Williams (UTHSC), Karl Broman (UW-Madison)

Thank you for listening

References:

1. Chelsea Trotter, Hyeonju Kim, Gregory Farage, Pjotr Prins, Robert W Williams, Karl W Broman, Śaunak Sen, Speeding up eQTL scans in the BXD population using GPUs, *G3 Genes|Genomes|Genetics*, Volume 11, Issue 12, December 2021, jkab254, https://doi.org/10.1093/g3journal/jkab254
2. Lippert, Christoph, et al. “FaST Linear Mixed Models for Genome-Wide Association Studies.” *Nature Methods*, vol. 8, no. 10, 4 Sept. 2011, pp. 833–835, https://doi.org/10.1038/nmeth.1681. Accessed 3 June 2021.
3. Runcie, Daniel E., and Lorin Crawford. “Fast and Flexible Linear Mixed Models for Genome-Wide Genetics.” *PLOS Genetics*, vol. 15, no. 2, 8 Feb. 2019, p. e1007978, https://doi.org/10.1371/journal.pgen.1007978. Accessed 12 Nov. 2019.
4. Xiang Zhou and Matthew Stephens (2012). Genome-wide efficient mixed-model analysis for association studies. *Nature Genetics* 44, 821–824.
5. Zhang, Z., Ersoz, E., Lai, CQ. *et al.* Mixed linear model approach adapted for genome-wide association studies. *Nat Genet* 42, 355–360 (2010). https://doi.org/10.1038/ng.546

Interested in exploring more?

Further questions or comments?

➢ Please report to ***Issues*** on the GitHub page

➢ Contact the author (me): zyu20@uthsc.edu or on GitHub (id: learningMalanya)

Appendix:

Backup slides start here...

- What is a kinship matrix?
- Permutation testing framework
- Details about Bulkscan methods and demonstrations
- Weighted residual variances structure
- Bayesian posterior mode estimation formula

Our design goals

- **Why Linear Mixed Models ?**
  - Interpretable modeling of family structure (kinship matrix)

- **Why julia ?**
  - Easy to code;
  - Runs fast;
  - Other features: multiple dispatch, multi-processing…

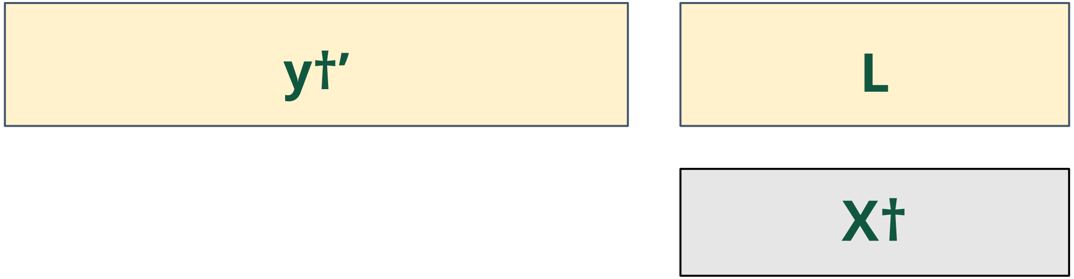
Compared to existing software (e.g., GEMMA, R/qtl), our program is designed to give the user a quick overview of the association tests of **many traits**.

Bulkscan-Null-Exact

Y = (y1, y2, y3, y4, y5, y6), for “Null-Exact” bulkscan method we process each trait independently, with each iteration doing:

Step 1: estimate h2 from null model, construct W and transform data to get **y†’, X†;**

Step 2: apply the matrix operation, taking the left matrix **as just one trait (vector)**



y†’

L

X†

To speed up Null-Exact, we parallelize the processes to have them run concurrently.

Bulkscan-Alt-Grid

But, can we apply some shortcuts in evaluating “LMM-Exact” - meaning that **to also estimate the heritability independently for each marker**?

Yes! Notice that:

- For a given h2, we can compute the “LOD scores” for multiple traits and markers using the matrix multiplication scheme:

  while they are **not technically the LOD scores under linear mixed models**, it still allows us to compute

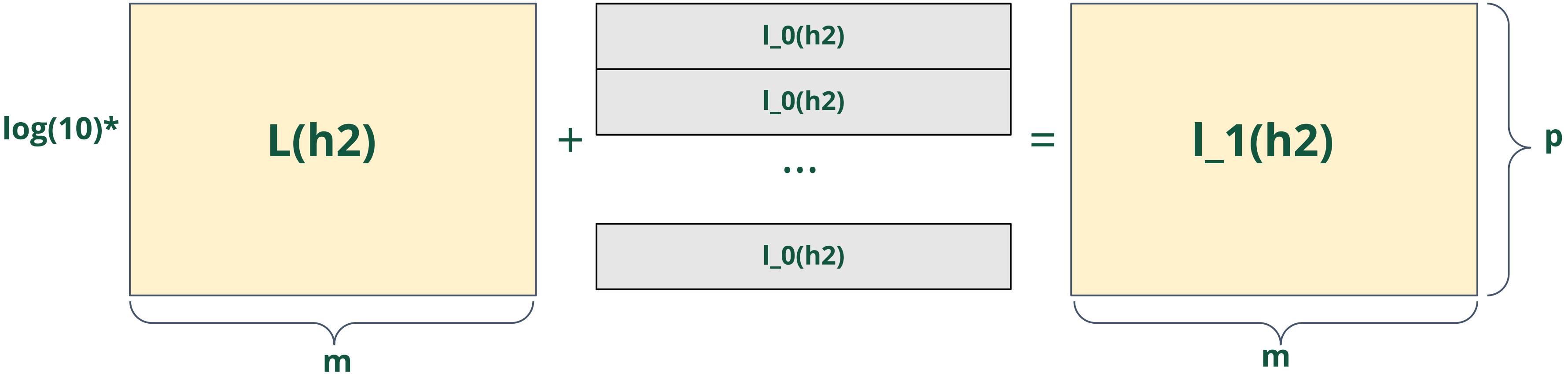
  $$L(h_k^2) = [l_1(h_k^2) - l_0(h_k^2)]/log(10)$$

  for every pair of traits and markers.

- We can then use $l_1(h_k^2) = log(10) * L(h_k^2) + l_0(h_k^2)$
  for optimization of loglikelihood of alternative model on h2

Bulkscan-Alt-Grid

For a given h2-grid, and for each value h2 in the h2-grid, we do:



log(10)*

L(h2)

m

+

l_0(h2)

l_0(h2)

…

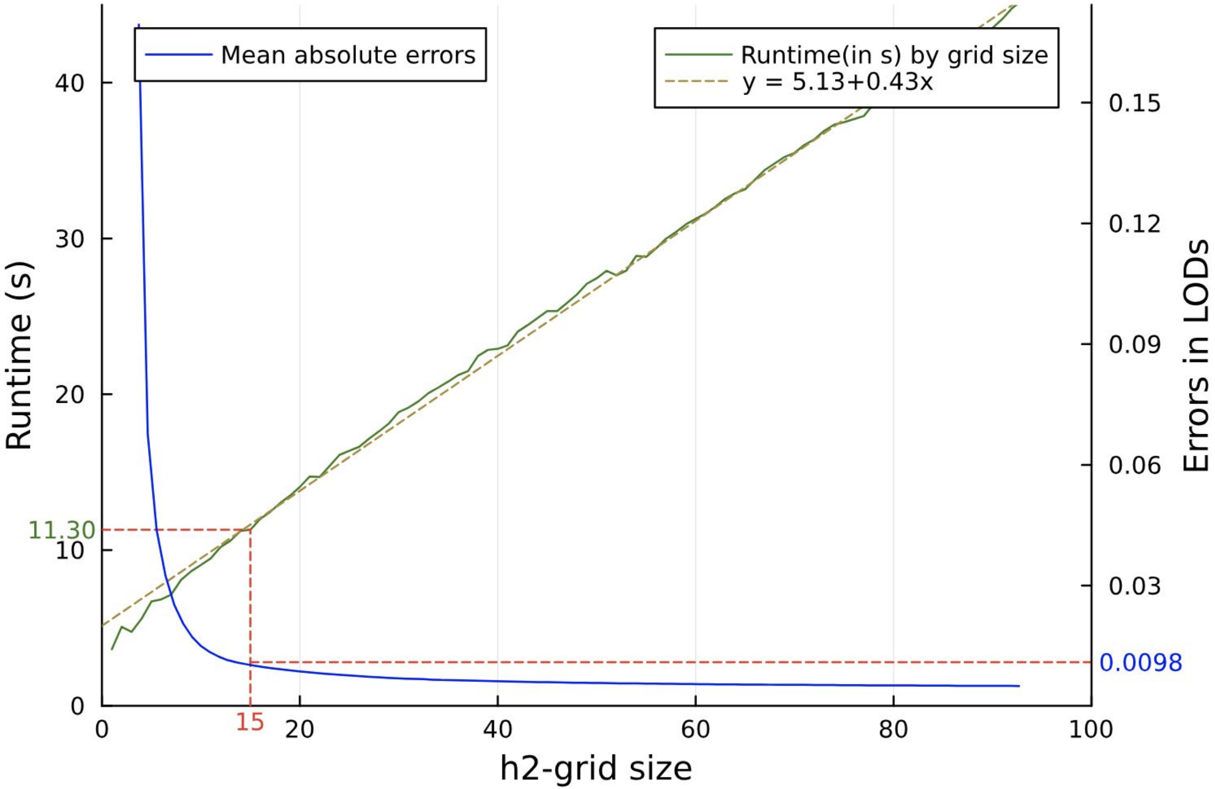
l_0(h2)

=

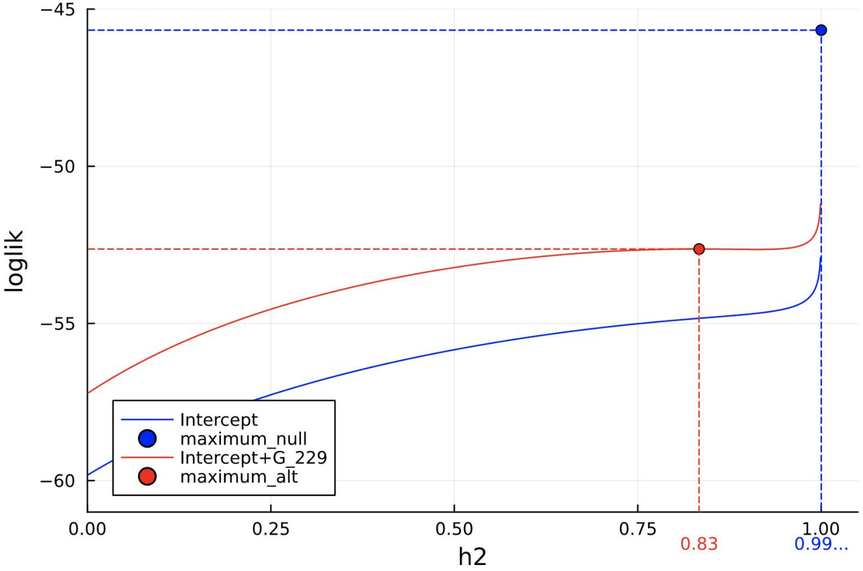
l_1(h2)

m

p


Mean absolute errors
Runtime(in s) by grid size
y = 5.13+0.43x
Runtime (s)
0
10
20
30
40
11.30
Errors in LODs
0.03
0.06
0.09
0.12
0.15
0.0098
h2-grid size
0
15
20
40
60
80
100

Bayesian Boundary Avoidance



What is wrong here?

- Estimated loglik of the **null model** is larger than that of the **alt. Model**

Why will this occur?

- **Heritability of 1** blows up the likelihood
- It suggests **environmental variance << genetic variance**

How can we deal with this issue?

- Imposing a prior belief that environmental variance **can't be too small**

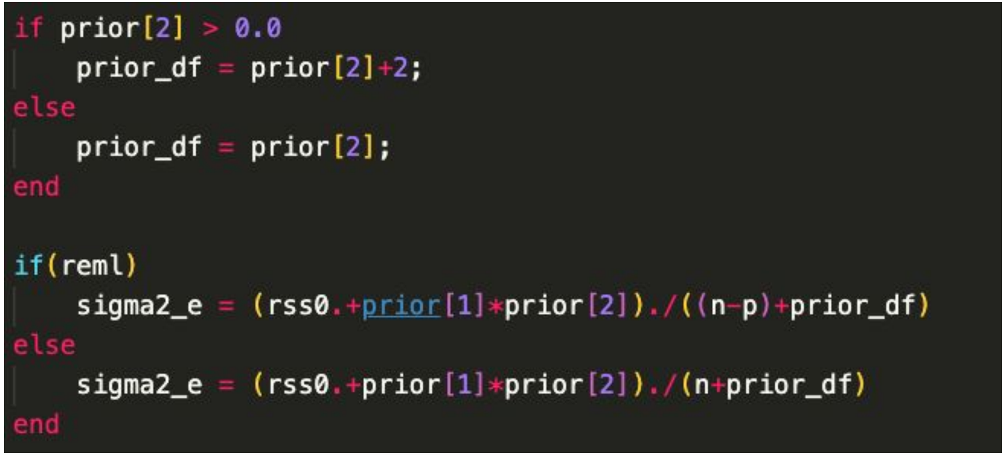
Bayesian Boundary Avoidance

MAP estimate of $p(\sigma_e^2|y^\dagger, v_0, \tau_0^2) = \text{Scaled-Inv-}\chi^2(v_n, \tau_n^2)$ **:**

$$v_n = n + v_0,\ \tau_n^2 = \frac{n}{n+v_0}s^2 + \frac{v_0}{n+v_0}\tau_0^2$$

$$s^2 = (y^\dagger - X^\dagger\beta)^T(y^\dagger - X^\dagger\beta)/n$$

$$\hat{\sigma}_e^2 = \frac{v_n\tau_n^2}{v_n+2} = \frac{ns^2+v_0\tau_0^2}{n_0+v_0+2}$$



```
if prior[2] > 0.0
    prior_df = prior[2]+2;
else
    prior_df = prior[2];
end

if(reml)
    sigma2_e = (rss0.+prior[1]*prior[2])./((n-p)+prior_df)
else
    sigma2_e = (rss0.+prior[1]*prior[2])./(n+prior_df)
end
```

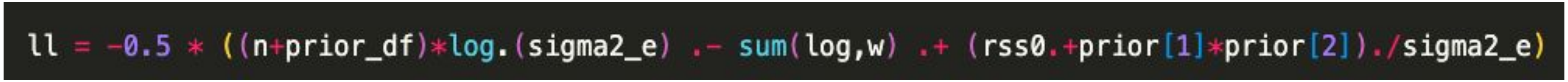
Bayesian Boundary Avoidance

Objective function (posteriori) under MAP estimates:

$$p(\sigma_e^2|y^\dagger, v_0, \tau_0^2) = \text{Scaled-Inv-}\chi^2(v_n, \tau_n^2) \tag{1}$$

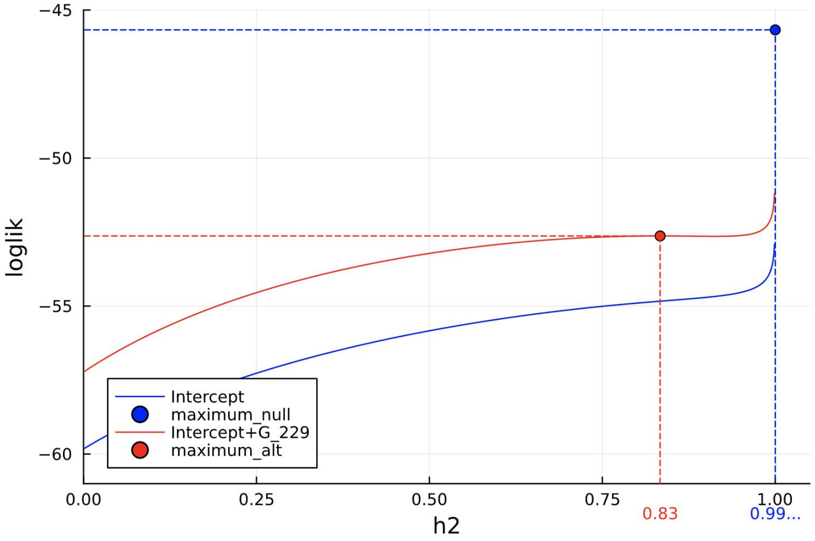
$$\propto (\sigma_e^2)^{-(\frac{v_n}{2}+1)} exp\{-\frac{v_n\tau_n^2}{2\sigma_e^2}\} \tag{2}$$

$$= exp\{-\frac{n+v+2}{2} log(\sigma_e^2) - \frac{ns^2 + v_0\tau_0^2}{2\sigma_e^2}\} \tag{3}$$

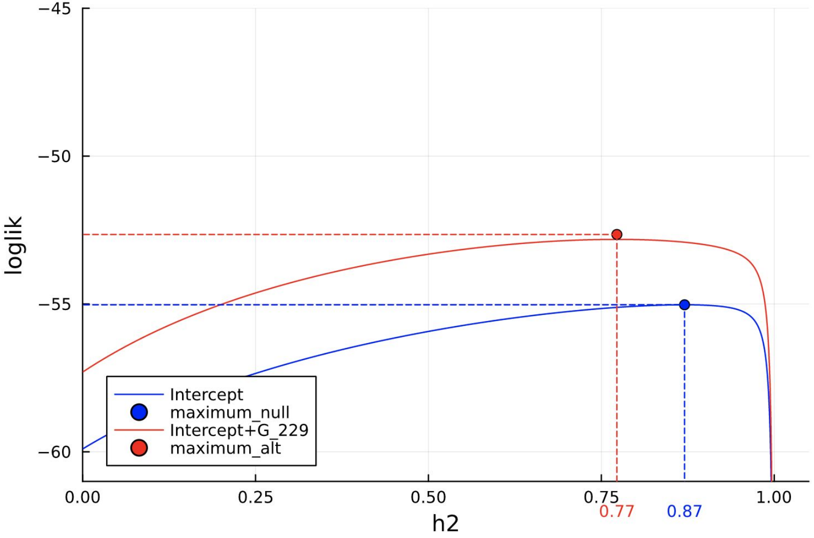


```
ll = -0.5 * ((n+prior_df)*log.(sigma2_e) .- sum(log,w) .+ (rss0.+prior[1]*prior[2])./sigma2_e)
```

Bayesian Boundary Avoidance

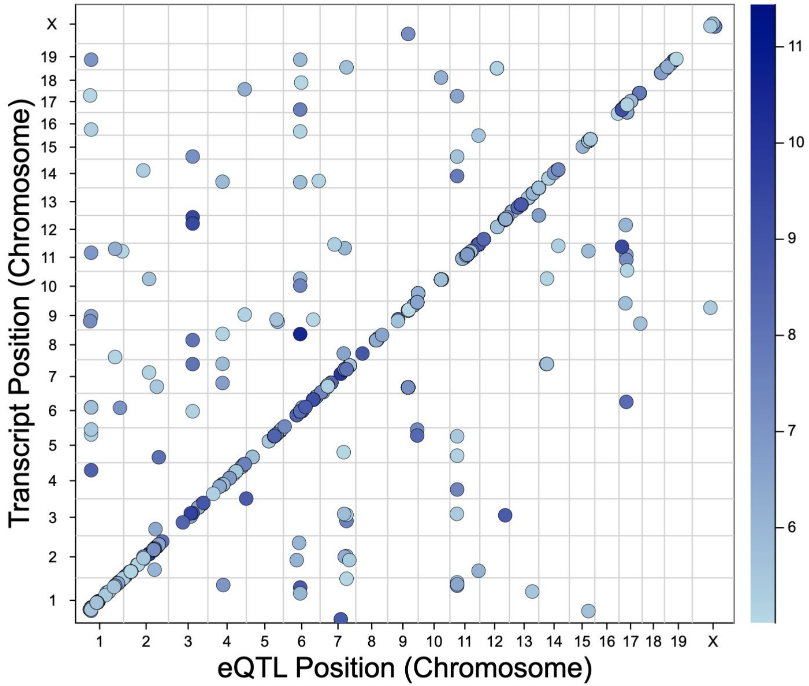


Normal likelihood



Posterior: with prior Scaled-Inv-Chisq(0.1, 1.0)

Expression QTL (eQTL) Plot



(Threshold = 5.0)