Finemap Mixer: A variational Bayesian Approach for Genetic Finemapping

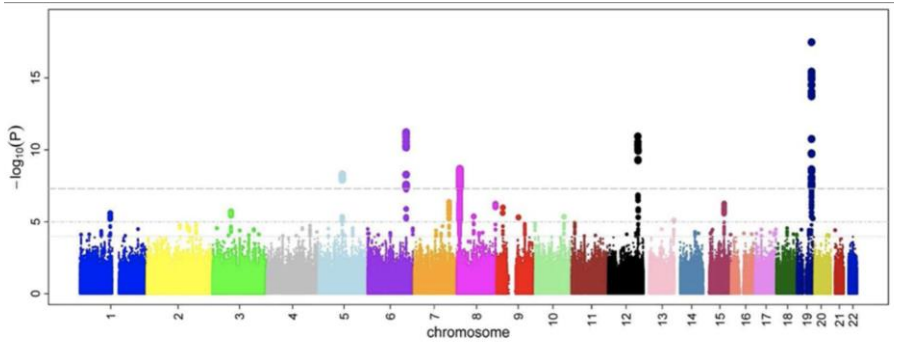
Bayram Cevdet Akdeniz

UNIVERSITAS OSLOENSIS MDCCCXI

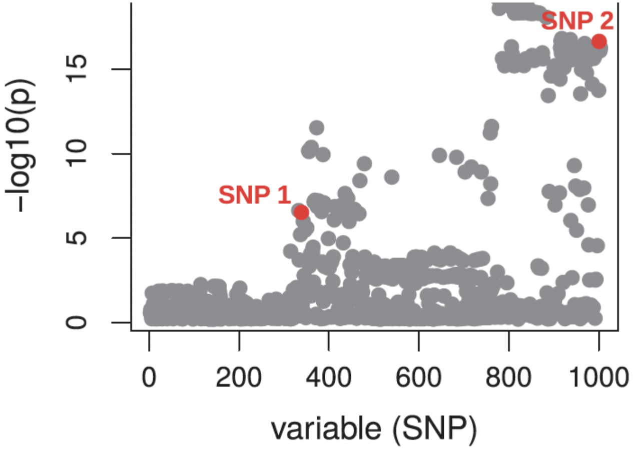
Finemapping

Trait+ genome -> GWAS (plink etc.)

-Manhattan plot -> gives -10 log(pval)
-to see associated SNPs/loci



- ➢ SNPs with lowest p-value are really the causal ?
- ➢ Not actually since:
- it has not been measured/imputed on a given study
- a causal SNP may not have the lowest p-value due to insufficient power
- Linkage Disequilibrium can lead to highly correlated association results and many false significant SNPs at a locus



- ➢ Fine-mapping methods aims selecting and prioritizing variants most likely responsible for traits

Literature

- There are many existing methods in the literature: heuristic, penalized regression, Bayesian methods[1]

- Bayesian methods are more successfull-> More flexible and suitable for finemapping

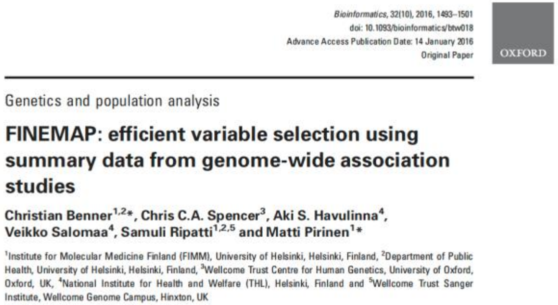
- BIMBAM (Servin 2007), CAVIAR (Hormozdiari 2014), CAVIARBF (Chen 2015), and PAINTOR (Kichaev 2014) -> rely on exhaustive search of the possible causal configurations on given model

- Not computational friendly -> $\binom{M}{k}$ possible configurations

[1] Schaid, Daniel J., Wenan Chen, and Nicholas B. Larson. "From genome-wide associations to candidate causal variants by statistical fine-mapping." *Nature Reviews Genetics* 19.8 (2018): 491-504.

Literature

- ➢ FINEMAP (C. S. Benner 2016) focuses on computation problem by using Cholesky Decomposition, and then search possible configurations via the Shotgun Stochastic Search



Bioinformatics, 32(10), 2016, 1493–1501
doi: 10.1093/bioinformatics/btw018
Advance Access Publication Date: 14 January 2016
Original Paper

OXFORD

Genetics and population analysis

FINEMAP: efficient variable selection using summary data from genome-wide association studies

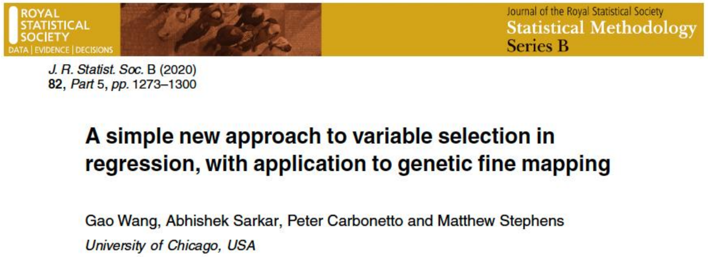
Christian Benner[1,2]*, Chris C.A. Spencer[3], Aki S. Havulinna[4], Veikko Salomaa[4], Samuli Ripatti[1,2,5] and Matti Pirinen[1]*

[1]Institute for Molecular Medicine Finland (FIMM), University of Helsinki, Helsinki, Finland, [2]Department of Public Health, University of Helsinki, Helsinki, Finland, [3]Wellcome Trust Centre for Human Genetics, University of Oxford, Oxford, UK, [4]National Institute for Health and Welfare (THL), Helsinki, Finland and [5]Wellcome Trust Sanger Institute, Wellcome Genome Campus, Hinxton, UK

-Fairly faster than the previous methods

-Provides similar accuracy

- ➢ SuSiE (Wang 2020) modelling locus with the Sum of Single effects model and optimizing by eliminating the effect of each causal SNP iteratively using Iterative Bayesian stepwise selection (IBSS).



ROYAL STATISTICAL SOCIETY
DATA | EVIDENCE | DECISIONS

Journal of the Royal Statistical Society
Statistical Methodology
Series B

J. R. Statist. Soc. B (2020)
82, *Part* 5, *pp.* 1273–1300

A simple new approach to variable selection in regression, with application to genetic fine mapping

Gao Wang, Abhishek Sarkar, Peter Carbonetto and Matthew Stephens
University of Chicago, USA

- Uses variational Bayesian approach

-gives the list of credible sets of causal SNPs

- claims better accuracy than other methods.

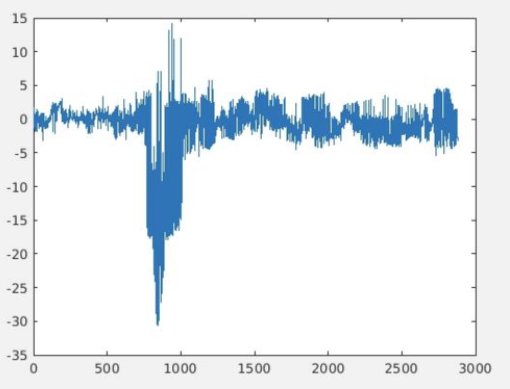
Our method

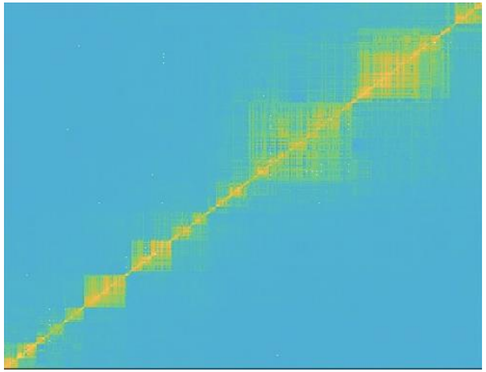
- We have proposed a finemap method to detect causal SNPs using Mixer Model and Variational Bayesian approach

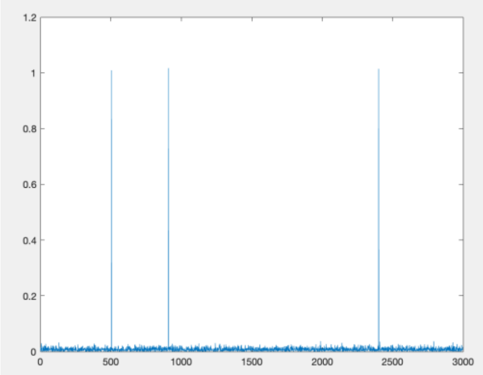
- It will be based on optimization of the likelihood obtained from Mixer model

- hence no need for exhaustive search among locus

- Z- scores + LD Matrix -> Causal SNPs







Model

$$y_k = \sum_{i=1}^{M} g_{ki}\beta_i + e \quad \leftrightarrow \quad \mathbf{y} = G\beta + e \qquad\qquad z_j = \sum_{i=1}^{M} a_{ij}\beta_i + \epsilon \quad \leftrightarrow \quad \mathbf{z} = A\beta + \epsilon \qquad\qquad p(\epsilon_j) = N(\epsilon_j|0,\sigma_0^2)$$

$$p(\beta_i) = (1-\pi_1)N(\beta_i|0,0) + \pi_1 N(\beta_i|0,\sigma_\beta^2),$$

$$p(z_j|\beta_1,\ldots,\beta_M,\theta) = N\left(z_j\middle|\sum_{i=1}^{M} a_{ij}\beta_i, \sigma_0^2\right), \qquad\qquad \theta = (\pi_1, \sigma_\beta^2, \sigma_0^2)$$

Define a latent variable u_i (equals to 1 if causal and 0 o.w.)

$$p(\beta_i|u_i = 0,\theta) = N(\beta_i|0,0), \qquad p(\beta_i|u_i = 1,\theta) = N(\beta_i|0,\sigma_\beta^2),$$
$$p(u_i|\theta) = Bern(u_i|\pi_1).$$

Full probabilistic model: $p(z,\beta,u|\theta) = p(z|\beta,\theta)\cdot p(\beta|u,\theta)\cdot p(u|\theta))$

Aim: To optimize likelohood: $p(\vec{z}|\theta) = \prod_j \int_u \int_\beta p(z_j,\vec{\beta},\vec{u},\theta)\,du\,d\beta$

Not easy to solve, z may be dependent on multiple betas

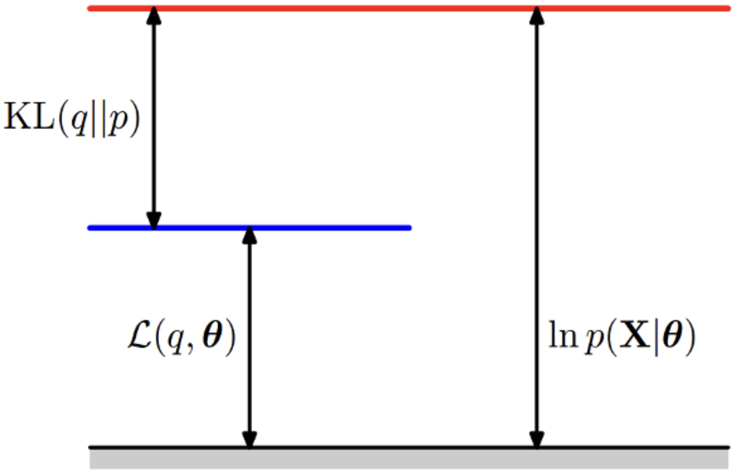
To optimize:

$$p(\vec{z}|\theta) = \prod_j \int_u \int_\beta p(z_j, \vec{\beta}, \vec{u}, \theta) du d\beta$$

Not easy to solve, z may be dependent on multiple betas

Solution: optimize evidence lower bound (ELBO) of this function

$$\begin{aligned}\log p(z|\theta) &= E_{q(\beta,u)}[\log p(z,\beta,u|\theta) - \log q(\beta,u)] + KL(q(\beta,u)||p(\beta,u|z,\theta)) \geq \\ &\geq E_{q(\beta,u)}[\log p(z,\beta,u|\theta) - \log q(\beta,u)] = \mathcal{L}(q,\theta) \to \max_{q,\theta}\end{aligned}$$


$\mathrm{KL}(q||p)$
$\mathcal{L}(q,\boldsymbol{\theta})$
$\ln p(\mathbf{X}|\boldsymbol{\theta})$

If parametric family(q) is chosen conveniently and optimized, then the parameters of the parametric family (u_i and q_i) can be used to finemap the given locus

KL(q||p) : is a measure that of how distribution q differs from p

Chosen parametric family:

$$q(\beta, u) = \prod_{i=1}^{M} Bern(u_i|q_i)N(\beta_i|\mu_i, \sigma_i^2);$$

μ_i: mean value of the estimation of β_i

σ_i: std of β_i

q_i: posterior prob. of being causal

Will use this parametric family to obtain ELBO and optimize ELBO to determine μ_i q_i for finemapping

We need to get the first derivatives for optimization

$$\mathcal{L}(q, \theta) = \underbrace{E_{q(\beta)} \sum_{j=1}^{M} \log p(z_j|\beta, \theta)}_{T_1} - \underbrace{E_{q(u)} \sum_{i=1}^{M} KL(q(\beta_i)||p(\beta_i|u_i, \theta))}_{T_2} - \underbrace{\sum_{i=1}^{M} KL(q(u_i)||p(u_i|\theta))}_{T_3}$$

$$\mathcal{L}(q,\theta) = \underbrace{E_{q(\beta)}\sum_{j=1}^{M}\log p(z_j|\beta,\theta)}_{T_1} - \underbrace{E_{q(u)}\sum_{i=1}^{M} KL(q(\beta_i)||p(\beta_i|u_i,\theta))}_{T_2} - \underbrace{\sum_{i=1}^{M} KL(q(u_i)||p(u_i|\theta))}_{T_3}$$

- Requires the calculation of first derivatives wrt decision variables
- Needs some special treatments (For details: https://www.biorxiv.org/content/biorxiv/early/2022/12/02/2022.11.30.518509/DC1/embed/media-1.pdf?download=true)
- For instance, it is not easy to calculate derivatives with the form of $\dfrac{\partial E_{q(a)} f(b)}{\partial a}$
- We have such cases in T_1
- Reparametrization Trick* is used to solve such kind of issue using parametric

$\boldsymbol{\epsilon} \in [\epsilon_1 \epsilon_2 \ldots \epsilon_M] \sim \boldsymbol{N(0, I)}$ such that $\beta = \mu + \sigma \boldsymbol{\epsilon}$

- Enables to rewrite the problematic expression as follows

$$E_{q(\beta_i|\mu_i,\sigma_i^2)} \log p(z|\beta_i,\theta) = E_\epsilon \log p(z|\beta_i(\epsilon,\mu_i,\sigma_i^2),\theta)$$

*M. Titsias and M. LÅLazaro-Gredilla, *"Doubly stochastic variational bayes for nonconjugate inference," in International conference on machine learning. PMLR, 2014, pp. 1971–1979.*

$$\mathcal{L}(q,\theta) = \underbrace{E_{q(\beta)}\sum_{j=1}^{M}\log p(z_j|\beta,\theta)}_{T_1} - \underbrace{E_{q(u)}\sum_{i=1}^{M}KL(q(\beta_i)||p(\beta_i|u_i,\theta))}_{T_2} - \underbrace{\sum_{i=1}^{M}KL(q(u_i)||p(u_i|\theta))}_{T_3}$$

$$T_1 = -M\log\left[\sqrt{2\pi\sigma_0^2}\right] - \frac{1}{2\sigma_0^2}\sum_{j=1}^{M}\sum_{i=1}^{M}(a_{ij}^2\sigma_i^2) - \frac{1}{2\sigma_0^2}\sum_{j=1}^{M}(z_j - \sum_{i=1}^{M}a_{ij}\mu_i)^2$$

$$T_2 = \sum_{i=1}^{M}E_{q(u)}KL(q(\beta_i)||p(\beta_i|u_i,\theta)) = \sum_{i=1}^{M}(1-q_i)\left(\log(\frac{\delta}{\sigma_i}) + \frac{\sigma_i^2+\mu_i^2}{2\delta^2} - \frac{1}{2}\right) +$$
$$\sum_{i=1}^{M}q_i\left(\log(\frac{\sigma_\beta}{\sigma_i}) + \frac{\sigma_i^2+\mu_i^2}{2\sigma_\beta^2} - \frac{1}{2}\right).$$

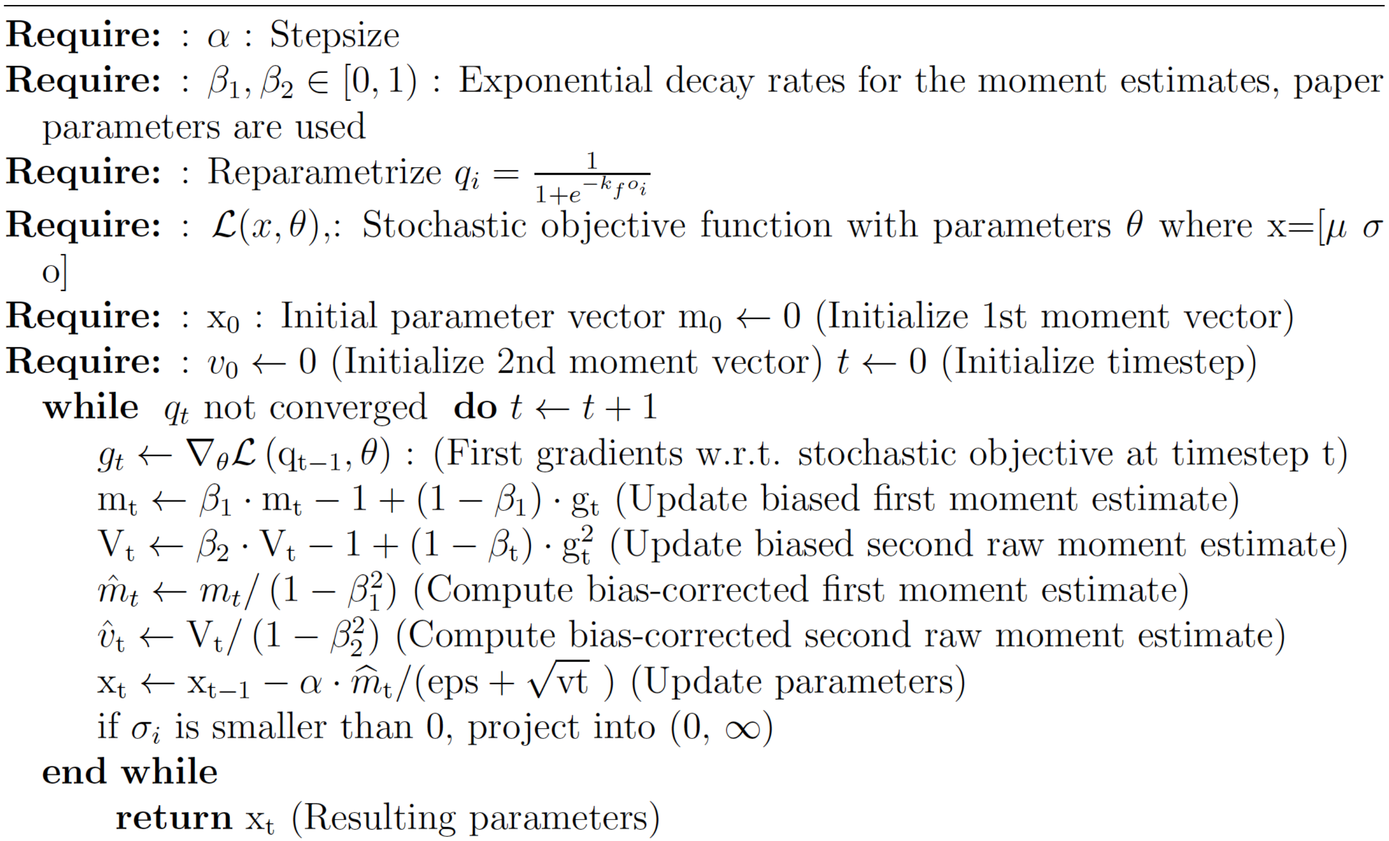
$$T_3 = \sum_{i=1}^{M}q_i\log\frac{q_i}{\pi_1} + (1-q_i)\log(\frac{1-q_i}{1-\pi_1})$$

- Some hyperparameters need to be either determined or pre-optimized such as δ, σ_0^2, π_1 and σ_β^2

	T$_1$	T$_2$	T$_3$	L$_{q,\theta}$
$\partial\mu_i$	$\frac{1}{\sigma_0^2}\sum_{j=1}^{M}$ $a_{ij}(z_j - \sum_{k=1}^{M}a_{kj}\mu_k)$	$\frac{(1-q_i)}{\delta^2} + \frac{(q_i)}{\sigma_\beta^2}$	0	$\frac{1}{\sigma_0^2}\sum_{j=1}^{M}a_{ij}(z_j - \sum_{k=1}^{M}a_{kj}\mu_k)$ $+ \frac{(1-q_i)\mu_i}{\delta^2} + \frac{(q_i)\mu_i}{\sigma_\beta^2}$
$\partial\sigma_i^2$	$\frac{-1}{4\sigma_0^2}\sum_{j=1}^{M}2a_{ij}^2$	$\frac{1}{2}\left(\frac{(1-q_i)\mu_i}{\delta^2} + \frac{(q_i)\mu_i}{\sigma_\beta^2} - \frac{1}{\sigma_i^2}\right)$	0	$\frac{-1}{4\sigma_0^2}\sum_{j=1}^{M}2a_{ij}^2 + \frac{1}{2}\left(\frac{(1-q_i)\mu_i}{\delta^2} + \frac{(q_i)\mu_i}{\sigma_\beta^2} - \frac{1}{\sigma_i^2}\right)$
∂q_i	0	$\log(\frac{\sigma_\beta}{\delta}) - \frac{\sigma_i^2+\mu_i^2}{2\delta^2} + \frac{\sigma_i^2+\mu_i^2}{2\sigma_\beta^2}$	$\sum_{i=1}^{M}\frac{\pi-q_i}{\pi-\pi^2}$	$\log(\frac{\sigma_\beta}{\delta}) - \frac{\sigma_i^2+\mu_i^2}{2\delta^2} + \frac{\sigma_i^2+\mu_i^2}{2\sigma_\beta^2} + \log\frac{q_i}{\pi} - \log\frac{1-q_i}{1-\pi}$
∇_θ	$\begin{bmatrix} 0 \\ 0 \\ \frac{T_A - M\sigma_0^2}{2\sigma_0^4} \end{bmatrix}$	$\begin{bmatrix} 0 \\ \sum_{i=1}^{M}\frac{-q_i}{2\sigma_\beta^4}\left(\sigma_i^2+\mu_i^2-\sigma_\beta^2\right) \\ 0 \end{bmatrix}$	$\begin{bmatrix} \sum_{i=1}^{M}\frac{\pi-q_i}{\pi-\pi^2} \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \sum_{i=1}^{M}\frac{\pi-q_i}{\pi-\pi^2} \\ \sum_{i=1}^{M}\frac{-q_i}{2\sigma_\beta^4}\left(\sigma_i^2+\mu_i^2-\sigma_\beta^2\right) \\ \frac{T_A - M\sigma_0^2}{2\sigma_0^4} \end{bmatrix}$

Table 1: All partial derivatives of $\mathcal{L}_{q,\theta}$

- **Optimization:** Adaptive Moment Estimation (Adam) algorithm to optimize ELBO
- ADAM: computing adaptive learning rate for each parameter using the exponantially decaying average of the first and second moments of first derivarives
- Efficient computation & easy implementation



```
Require: : $\alpha$ : Stepsize
Require: : $\beta_1, \beta_2 \in [0, 1)$ : Exponential decay rates for the moment estimates, paper
  parameters are used
Require: : Reparametrize $q_i = \frac{1}{1+e^{-k_f o_i}}$
Require: : $\mathcal{L}(x, \theta)$,: Stochastic objective function with parameters $\theta$ where x=[$\mu$ $\sigma$
  o]
Require: : $\mathrm{x}_0$ : Initial parameter vector $\mathrm{m}_0 \leftarrow 0$ (Initialize 1st moment vector)
Require: : $v_0 \leftarrow 0$ (Initialize 2nd moment vector) $t \leftarrow 0$ (Initialize timestep)
  while $q_t$ not converged do $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_\theta \mathcal{L}\left(\mathrm{q}_{\mathrm{t}-1}, \theta\right)$ : (First gradients w.r.t. stochastic objective at timestep t)
    $\mathrm{m}_\mathrm{t} \leftarrow \beta_1 \cdot \mathrm{m}_\mathrm{t} - 1 + (1 - \beta_1) \cdot \mathrm{g}_\mathrm{t}$ (Update biased first moment estimate)
    $\mathrm{V}_\mathrm{t} \leftarrow \beta_2 \cdot \mathrm{V}_\mathrm{t} - 1 + (1 - \beta_\mathrm{t}) \cdot \mathrm{g}_\mathrm{t}^2$ (Update biased second raw moment estimate)
    $\hat{m}_t \leftarrow m_t / \left(1 - \beta_1^2\right)$ (Compute bias-corrected first moment estimate)
    $\hat{v}_\mathrm{t} \leftarrow \mathrm{V}_\mathrm{t} / \left(1 - \beta_2^2\right)$ (Compute bias-corrected second raw moment estimate)
    $\mathrm{x}_\mathrm{t} \leftarrow \mathrm{x}_{\mathrm{t}-1} - \alpha \cdot \widehat{m}_\mathrm{t} / (\mathrm{eps} + \sqrt{\mathrm{vt}}\ )$ (Update parameters)
    if $\sigma_i$ is smaller than 0, project into (0, $\infty$)
  end while
    return $\mathrm{x}_\mathrm{t}$ (Resulting parameters)
```

- We need to do some modifications on ADAM algorithm

- Our decision variables have some constraints (σ_i^2 and q_i)
- To satisfy these constraints, we may either employ Reparamtrization (REP) or Projected Gradient (PG) approaches.
- For optimization of q_i we are using REP by reparametrizing q_i with another variable o_i as

$$q_i = \frac{1}{1 + e^{-k_f o_i}}$$

- And corresponding derivative can be easily calculated as

$$\frac{\partial \mathcal{L}_{q,\theta}}{\partial o_i} = \frac{\partial \mathcal{L}_{q,\theta}}{\partial q_i} \frac{\partial q_i}{\partial o_i}.$$

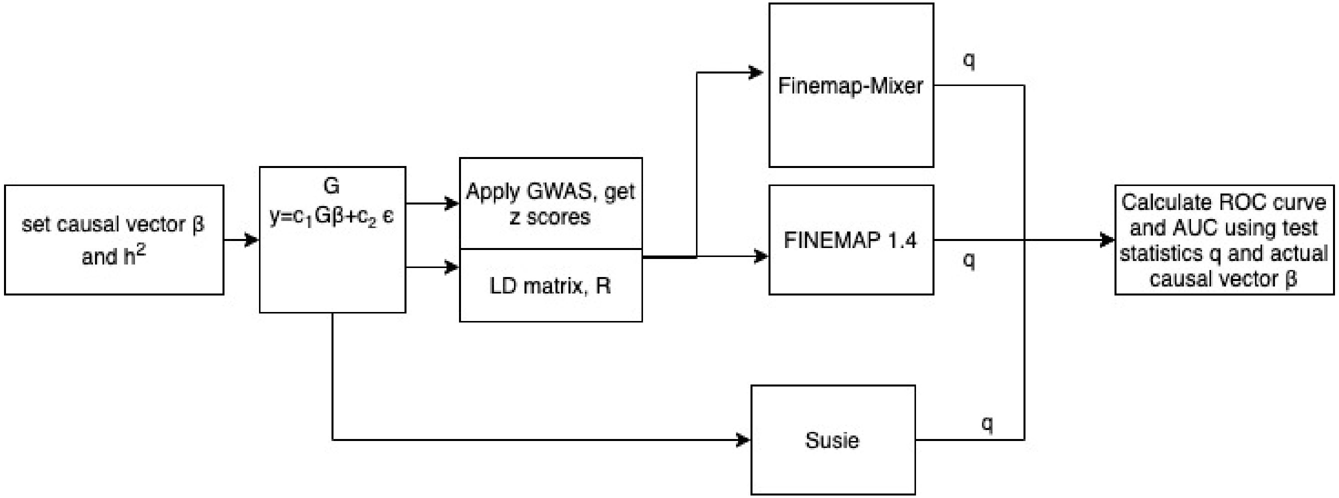
- For σ_i^2 we are using Projected Gradient which is basically projecting the calculated σ_i^2 to defined space which is (0, ∞) in our case (just rescale them if there is some negative)

Simulations

- Evaluated the performance on synthetic data and real data(UKB Height data)
- Compare the performance with most recent Finemap Methods (Finemap 1.4 and SuSiE)

Synthetic data simulations

- In Hapgen data, a randomly chosen locus with M SNPs is used(1-10 causal SNPs will be assumed)
- Examined scenarios for different heritabilities
- Simulated phenotype data (y) with desired heritability (h2), b= [b_1 , $b_{2\ ...}$ $b_{j.\ ...}$ b_M], b_j =1 if causal, 0 o.w.



set causal vector β and h^2

G

$y=c_1 G\beta + c_2\,\epsilon$

Apply GWAS, get z scores

LD matrix, R

Finemap-Mixer

FINEMAP 1.4

Susie

q

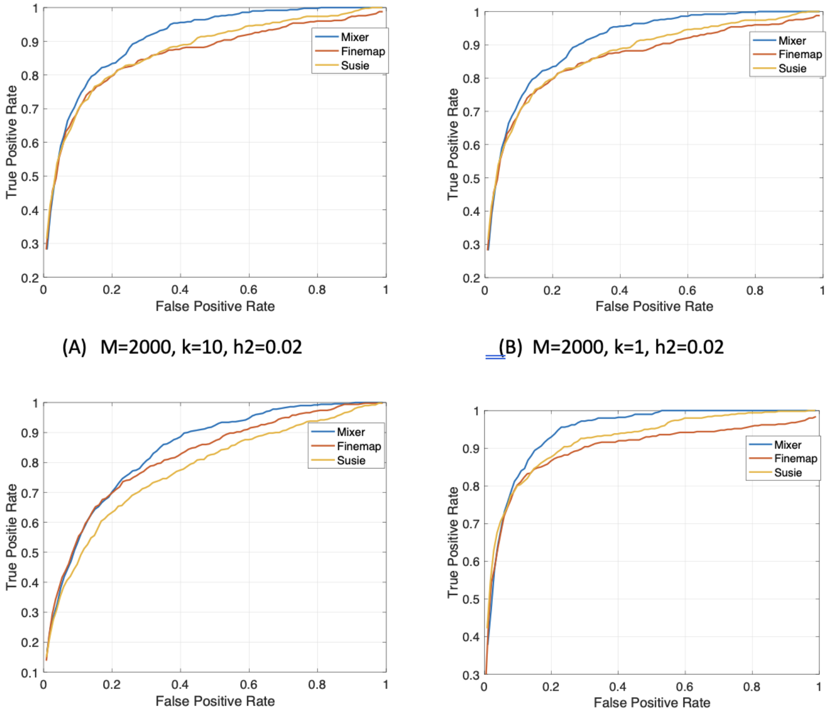
q

q

Calculate ROC curve and AUC using test statistics q and actual causal vector β

UiO : University of Oslo

- **Simulations with Synthetic Data**

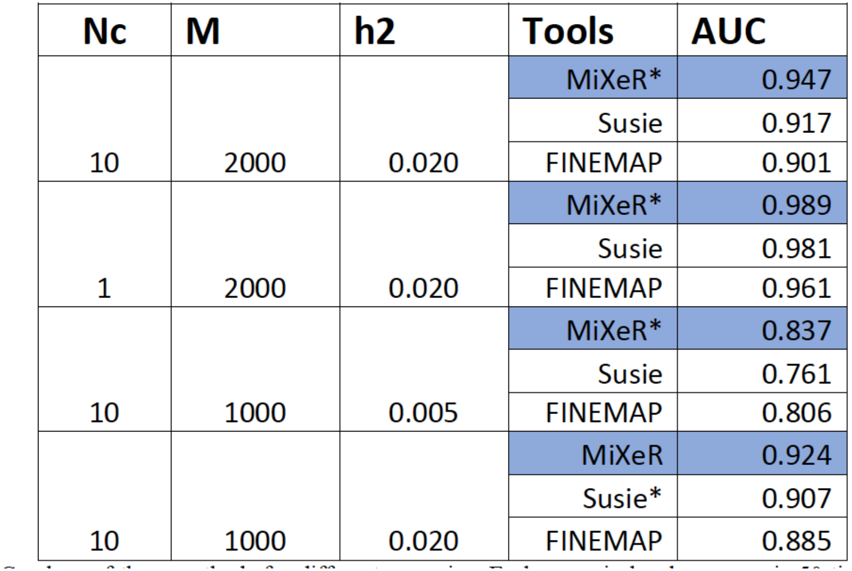


(A) M=2000, k=10, h2=0.02

(B) M=2000, k=1, h2=0.02

(C) M=1000, k=10, h2=0.005

(D) M=1000, k=10, h2=0.02



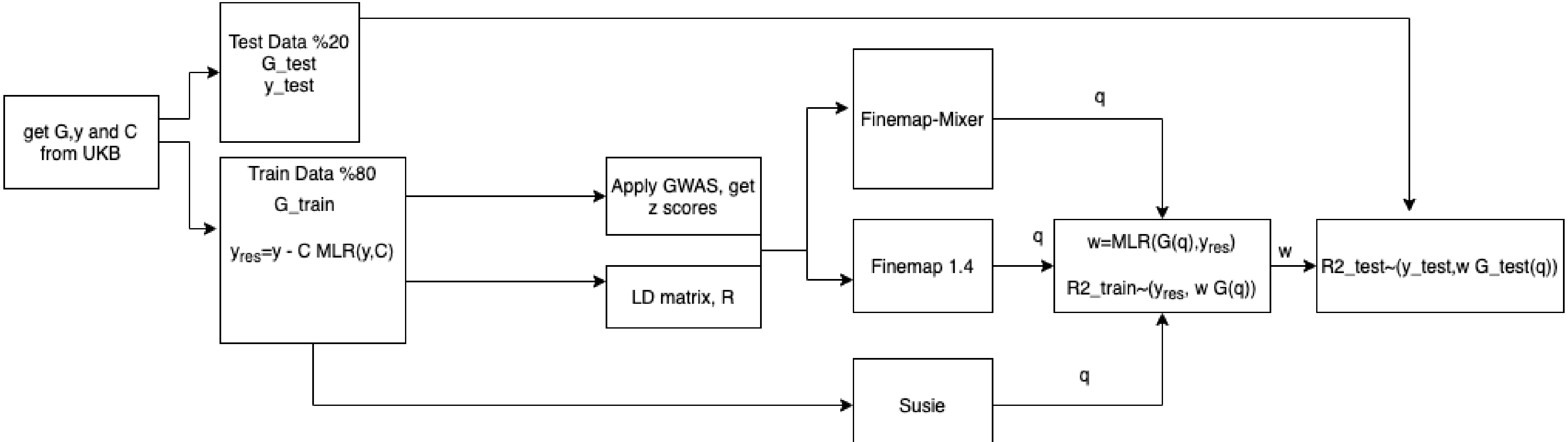
Nc	M	h2	Tools	AUC
10	2000	0.020	MiXeR*	0.947
			Susie	0.917
			FINEMAP	0.901
1	2000	0.020	MiXeR*	0.989
			Susie	0.981
			FINEMAP	0.961
10	1000	0.005	MiXeR*	0.837
			Susie	0.761
			FINEMAP	0.806
10	1000	0.020	MiXeR	0.924
			Susie*	0.907
			FINEMAP	0.885

we have compared the performance of three methods using Area Under Curve (AUC) of ROC curves for 31 different scenarios. For most scenarios (21/31), Finemap-MiXeR outperforms the other methods in AUC. Although there are some cases where other methods are better (FINEMAP; 6/31, SuSiE 1/31),

Simulation with real data

- UK Biobank genome data is used
- Height is chosen as phenomic data
- Effects of covariates (age,sex, etc.) are eliminated
- Focused on most heritable loci

- We do not have ground truth causals as we have in synthetic data
- Instead once we get posteriors we choose causal SNPs (G_{snp}), and estimate phenotype using Multiple Linear Regression (MLR)



get G,y and C from UKB

Test Data %20
G_test
y_test

Train Data %80
G_train
y_{res}=y - C MLR(y,C)

Apply GWAS, get z scores

LD matrix, R

Finemap-Mixer

Finemap 1.4

Susie

q

q

q

w=MLR(G(q),y_{res})
R2_train~(y_{res}, w G(q))

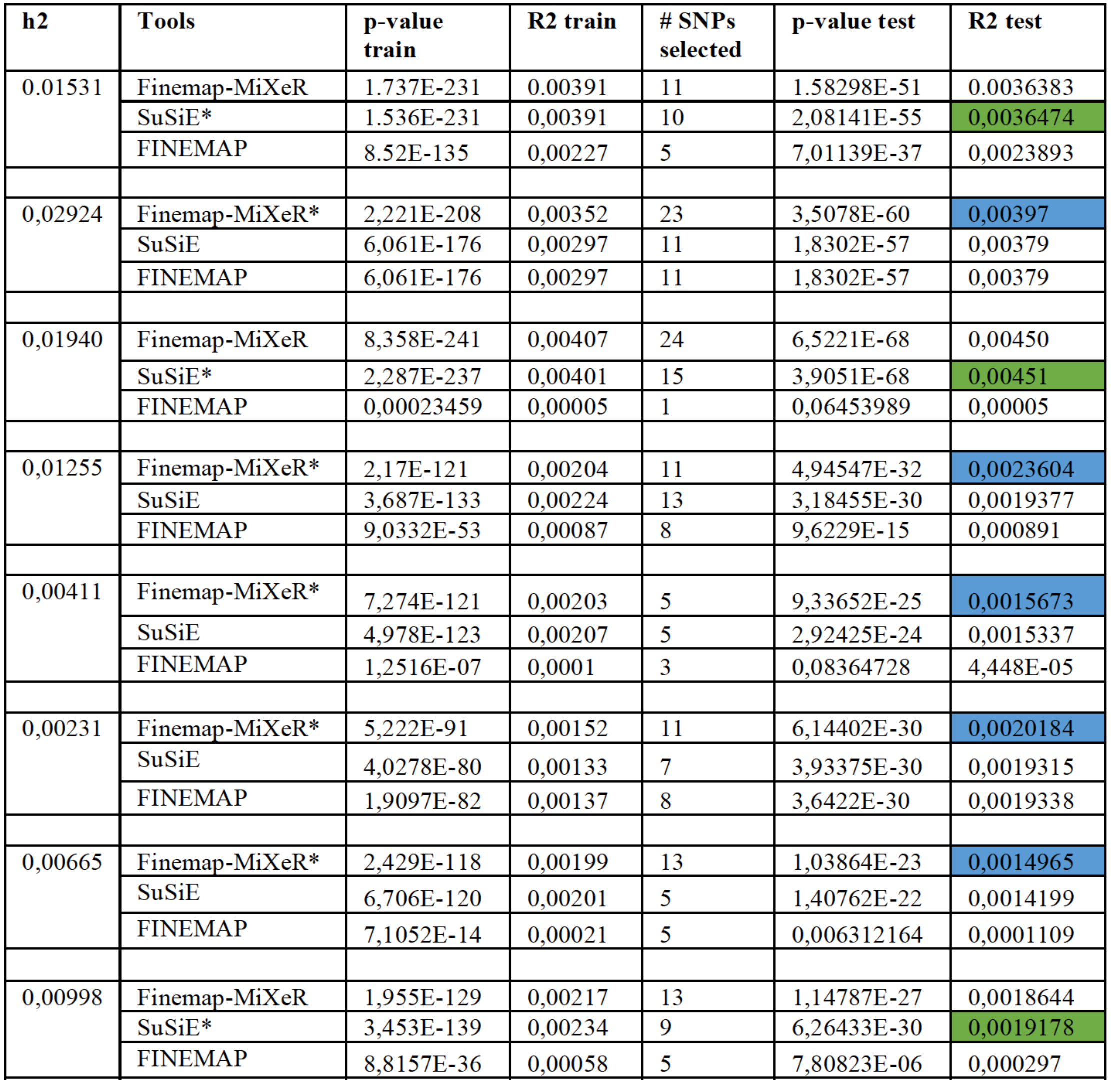
w

R2_test~(y_test,w G_test(q))

- **Simulation with real data**

- $\hat{y} \sim MLR(G_{snp}, y)$, $R2 \sim (y, \hat{y})$
- data is splitted into %80 for train and rest for test

- in most cases (11/15), our method has the highest R2



h2	Tools	p-value train	R2 train	# SNPs selected	p-value test	R2 test
0.01531	Finemap-MiXeR	1.737E-231	0.00391	11	1.58298E-51	0.0036383
	SuSiE*	1.536E-231	0,00391	10	2,08141E-55	0,0036474
	FINEMAP	8.52E-135	0,00227	5	7,01139E-37	0,0023893
0,02924	Finemap-MiXeR*	2,221E-208	0,00352	23	3,5078E-60	0,00397
	SuSiE	6,061E-176	0,00297	11	1,8302E-57	0,00379
	FINEMAP	6,061E-176	0,00297	11	1,8302E-57	0,00379
0,01940	Finemap-MiXeR	8,358E-241	0,00407	24	6,5221E-68	0,00450
	SuSiE*	2,287E-237	0,00401	15	3,9051E-68	0,00451
	FINEMAP	0,00023459	0,00005	1	0,06453989	0,00005
0,01255	Finemap-MiXeR*	2,17E-121	0,00204	11	4,94547E-32	0,0023604
	SuSiE	3,687E-133	0,00224	13	3,18455E-30	0,0019377
	FINEMAP	9,0332E-53	0,00087	8	9,6229E-15	0,000891
0,00411	Finemap-MiXeR*	7,274E-121	0,00203	5	9,33652E-25	0,0015673
	SuSiE	4,978E-123	0,00207	5	2,92425E-24	0,0015337
	FINEMAP	1,2516E-07	0,0001	3	0,08364728	4,448E-05
0,00231	Finemap-MiXeR*	5,222E-91	0,00152	11	6,14402E-30	0,0020184
	SuSiE	4,0278E-80	0,00133	7	3,93375E-30	0,0019315
	FINEMAP	1,9097E-82	0,00137	8	3,6422E-30	0,0019338
0,00665	Finemap-MiXeR*	2,429E-118	0,00199	13	1,03864E-23	0,0014965
	SuSiE	6,706E-120	0,00201	5	1,40762E-22	0,0014199
	FINEMAP	7,1052E-14	0,00021	5	0,006312164	0,0001109
0,00998	Finemap-MiXeR	1,955E-129	0,00217	13	1,14787E-27	0,0018644
	SuSiE*	3,453E-139	0,00234	9	6,26433E-30	0,0019178
	FINEMAP	8,8157E-36	0,00058	5	7,80823E-06	0,000297

Computational Complexity

- The computational complexity of Finemap-MiXeR relies on the online calculation of the first derivatives of $\mathcal{L}(q,\theta)$.

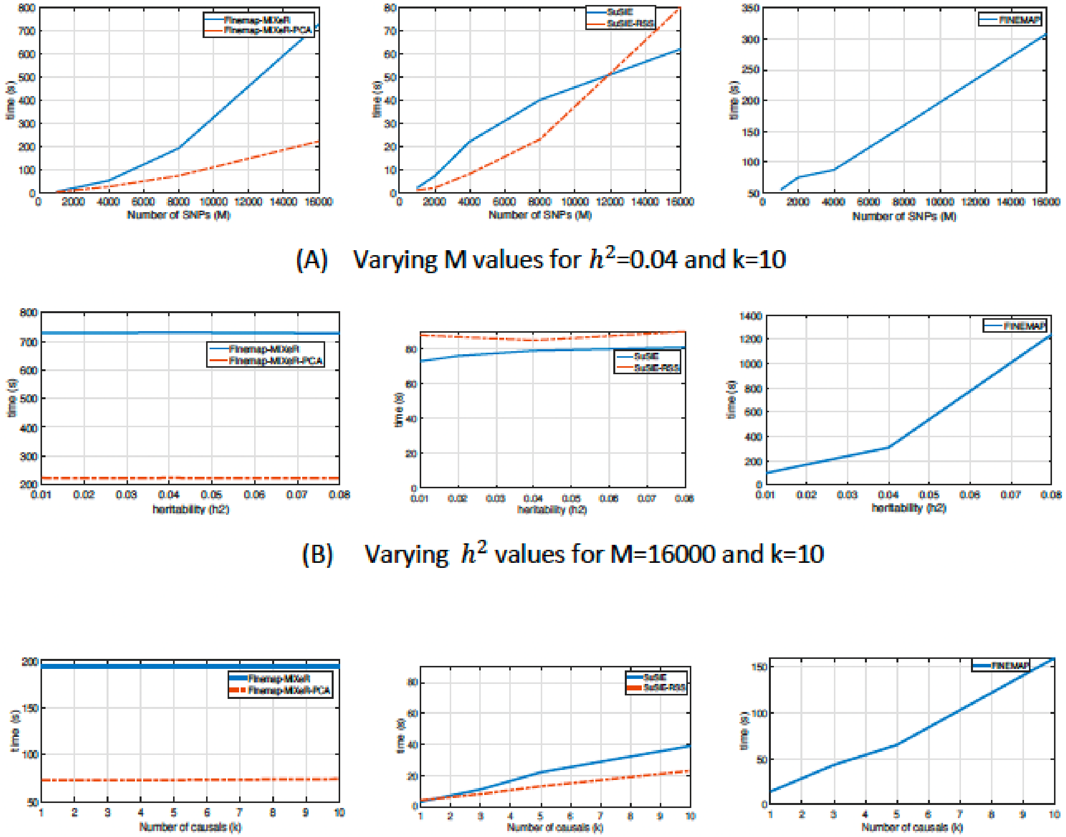
$$\frac{\partial \mathcal{L}_{q,\theta}}{\partial \boldsymbol{\mu}} = \frac{1}{\sigma_0^2}(A_1 + A_2\boldsymbol{\mu}))^T - \frac{(1-\boldsymbol{q}) \odot \boldsymbol{\mu}}{\delta^2} - \frac{\boldsymbol{q} \odot \boldsymbol{\mu}}{\sigma_\beta^2}$$

$$\frac{\partial \mathcal{L}_{q,\theta}}{\partial q_i} = -\left(\log(\frac{\sigma_\beta}{\delta}) - \frac{\sigma_i^2 + \mu_i^2}{2\delta^2} + \frac{\sigma_i^2 + \mu_i^2}{2\sigma_\beta^2} + \log\frac{q_i}{\pi} - \log\frac{1-q_i}{1-\pi}\right)$$

$$\frac{\partial \mathcal{L}_{q,\theta}}{\partial \sigma_i^2} = \left(\frac{-1}{4\sigma_0^2}\sum_{j=1}^{M} 2a_{ij}^2\right) - \frac{(1-q_i)}{\delta^2} - \frac{(q_i)}{\sigma_\beta^2}.$$

- requires O(M^2) computations per iteration.
- Other methods: O(kMN) for SuSiE, O(kM^2) for SuSiE RSS, O(k^{2}M) for FINEMAP
- we can reduce the complexity from O(M^2) to O(p$_c$M) by preserving accuracy, where p$_c$ << M. This improvement is achieved by using Principal Component Analysis (PCA)

Computational Complexity



(A) Varying M values for h^2=0.04 and k=10

(B) Varying h^2 values for M=16000 and k=10

(C) Varying k values for M=16000 and h^2=0.04

Fig 4. Run time comparison (in seconds) of the three methods, with their variants. Note that we have also included SuSiE RSS which uses SuSiE method with summary statistics, and Finemap-MiXeR PCA which reduces computational complexity by applying PCA. All tools are run in HPC with Intel Xeon CPU E5-2698 v4 @2.20GHZ. Unlike other methods, Finemap-MiXeR is only scalable with M and its computational complexity increases by M^2. When we apply Finemap-MiXeR PCA, the computational complexity further reduces to the order of M.

Conclusion

- We proposed a Finemap method that can outperform other existing methods in different aspects

- Can be both used to finemap a locus & estimate phenotypes

- Preprint has been submitted: https://www.biorxiv.org/content/10.1101/2022.11.30.518509v1

- Next steps: Using annotations (from GSA MiXeR) to prioritize SNPs (for now we assumed equal priors!!)

REFERENCES

- Beyond SNP Heritability: Polygenicity and Discoverability Estimated for Multiple Phenotypes with a Univariate Gaussian Mixture Model
- A. A. Shadrin, O. Frei, O. B. Smeland, F. Bettella, K. S. O’Connell, O. Gani, S. Bahrami, T. K. Uggen, S. Djurovic, D. Holland et al., “Phenotype-specific differences in polygenicity and effect size distribution across functional annotation categories revealed by ai-mixer,” Bioinformatics, vol. 36, no. 18, pp. 4749–4756, 2020.
- D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.
- M. Titsias and M. L´azaro-Gredilla, “Doubly stochastic variational bayes for nonconjugate inference,” in International conference on machine learning. PMLR, 2014, pp. 1971–1979.
- Schaid, D. J., Chen, W., & Larson, N. B. (2018). From genome-wide associations to candidate causal variants by statistical fine-mapping. Nature Reviews Genetics, 19(8), 491-504.