

1 Supplementary Notes

2 Statistical model

3 Let \mathbf{y} be a length- n vector that denotes the phenotypes of n samples. Denote by $\mathbf{X} \in \mathbb{R}^{n \times p}$ the genotype
4 matrix of n individuals based on p markers or SNPs. We standardize \mathbf{X} and \mathbf{y} such that the variance of the
5 phenotype is 1 and the variance of each marker-specific genotype vector is 1. We use an additive genetic
6 model for the phenotypes as $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon}$ is a length- n vector distributed as $\boldsymbol{\varepsilon} \sim N(0, \sigma_e^2 \mathbf{I}_n)$. We
7 assume both \mathbf{X} and $\boldsymbol{\beta}$ are random, such that $\boldsymbol{\Sigma} \equiv \mathbb{E}(\mathbf{X}^\top \mathbf{X}/n)$ and $\mathbf{D} \equiv \mathbb{E}(\boldsymbol{\beta}\boldsymbol{\beta}^\top)$. The covariance of the
8 effect sizes \mathbf{D} is determined by our parameters of interest, such as partitioned heritability and enrichment
9 of functional categories, which we denote by $\boldsymbol{\theta}$. We use $\mathbf{D}(\boldsymbol{\theta})$ to emphasize the dependence of \mathbf{D} on $\boldsymbol{\theta}$.

Let $\mathbf{z} = \mathbf{X}^\top \mathbf{y}/\sqrt{n}$ be the vector of marginal association statistics from OLS. Then the distribution of \mathbf{z} ,
conditional on the true causal effect sizes can be shown as,

$$\mathbf{z}|\boldsymbol{\beta}, \mathbf{X} \sim N(n^{-1/2} \mathbf{X}^\top \mathbf{X}\boldsymbol{\beta}, n^{-1} \mathbf{X}^\top \mathbf{X}\sigma_e^2) \quad (1)$$

$$\mathbf{z}|\boldsymbol{\beta} \sim N(n^{1/2} \boldsymbol{\Sigma}\boldsymbol{\beta}, \boldsymbol{\Sigma}) \quad (2)$$

10 We emphasize that the distinction between the two distributions above lies in whether the genotype is
11 conditioned on or assumed to be fixed. Our estimator is based on (2), which has been derived and used
12 previously in the literature¹⁻³. Suppose the causal effect sizes are drawn from a normal distribution,
13 $\boldsymbol{\beta} \sim N(0, \mathbf{D}(\boldsymbol{\theta}))$. Integrating over $\boldsymbol{\beta}$ leads to the marginal distribution,

$$\mathbf{z} \sim N(0, n\boldsymbol{\Sigma}\mathbf{D}(\boldsymbol{\theta})\boldsymbol{\Sigma} + \boldsymbol{\Sigma}), \quad (3)$$

14 and our estimator of $\boldsymbol{\theta}$ comes from maximizing the likelihood function based on this marginal density.
15 To make the algorithm computationally feasible, we work with the likelihood of $\tilde{\mathbf{z}} = \hat{\mathbf{P}}\mathbf{z}$, where $\hat{\mathbf{P}}$ is an
16 estimate of $\boldsymbol{\Sigma}^{-1}$ based on LDGM, which is extremely sparse. It is easy to show that $\tilde{\mathbf{z}} \sim N(0, \mathbf{M}(\boldsymbol{\theta}))$ where
17 $\mathbf{M}(\boldsymbol{\theta}) \equiv n\mathbf{D}(\boldsymbol{\theta}) + \hat{\mathbf{P}}$. We directly maximize the likelihood,

$$\ell(\boldsymbol{\theta}) = -\frac{1}{2} \left\{ \log(2\pi) + \log|\mathbf{M}(\boldsymbol{\theta})| + \tilde{\mathbf{z}}^\top \mathbf{M}(\boldsymbol{\theta})^{-1} \tilde{\mathbf{z}} \right\}, \quad (4)$$

18 and the graphREML estimator is defined as $\hat{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}) = \arg \max_{\boldsymbol{\theta}} \log \ell(\boldsymbol{\theta})$.

19 Estimation details

20 *Quasi-Newtonian algorithm*

We use a quasi-Newtonian algorithm for parameter estimation. This algorithm relies on efficient new subroutines, leveraging the sparsity of LDGM precision matrices to compute the likelihood function (4), together with its gradient and approximate Hessian. We iteratively update our estimate of the parameters as the following,

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - (\mathbf{H}^{(k)} + e\mathbf{I})^{-1} \nabla^{(k)},$$

21 where $\nabla^{(k)}$ and $\mathbf{H}^{(k)}$ are the gradient and the approximate Hessian of the likelihood function evaluated at
 22 the current estimate of the parameters $\boldsymbol{\theta}^{(k)}$. e is some small-valued number that is added to the diagonal of
 23 the Hessian matrix to prevent singularity in estimation. Let $\mathbf{M}(\boldsymbol{\theta}^{(k)}) = n\mathbf{D}(\boldsymbol{\theta}^{(k)}) + \hat{\mathbf{P}}$. At each iteration,
 24 we first perform a Cholesky factorization of the matrix $\mathbf{M}(\boldsymbol{\theta}^{(k)})$, which is feasible and computationally
 25 tractable due to the sparsity of $\hat{\mathbf{P}}$. Denote by $\frac{\partial \mathbf{D}_{\mathbf{a}}(\boldsymbol{\theta})}{\partial \theta_i}$ a diagonal matrix where the diagonal elements are
 26 the partial derivatives of the per-SNP heritability with respect to the parameters, $\left(\frac{\partial g_{\theta}(a_1)}{\partial \theta_i}, \dots, \frac{\partial g_{\theta}(a_p)}{\partial \theta_i} \right)$. We
 27 estimate the gradient using chain rule as the following,

$$\nabla_i^{(k+1)} = \frac{1}{2} n \left(\tilde{\mathbf{z}}^\top (\mathbf{M}^{(k)})^{-1} \frac{\partial \mathbf{D}_{\mathbf{a}}(\boldsymbol{\theta})}{\partial \theta_i} (\mathbf{M}^{(k)})^{-1} \tilde{\mathbf{z}} - \text{Tr} \left(\frac{\partial \mathbf{D}_{\mathbf{a}}(\boldsymbol{\theta})}{\partial \theta_i} (\mathbf{M}^{(k)})^{-1} \right) \right), \quad (5)$$

28 where we have used $\mathbf{M}^{(k)}$ to denote $\mathbf{M}(\boldsymbol{\theta}^{(k)})$ for simplicity of notation, and i indexes the parameters.
 29 Importantly, this equation is evaluated without computing $\mathbf{M}^{(k)}$ explicitly. In particular, the second term
 30 is evaluated by computing the sparse inverse subset of $\mathbf{M}^{(k)}$, as implemented in *suitesparse*⁴. We have
 31 four sparse matrix operations – matrix multiplication, division, log-determinant and matrix inverse. All of
 32 these functions were added to the LDGM package (see URL).

33 **Approximation of the Hessian matrix**

34 We use a second-order method to solve for the maximum likelihood estimators, which requires computing
 35 the Hessian matrix. The first derivative of (4) with respect to the i -th parameter at the k -th iteration is,

$$\nabla_i^{(k+1)} = \frac{1}{2}n \left(\tilde{\mathbf{z}}^\top (\mathbf{M}^{(k)})^{-1} \frac{\partial \mathbf{D}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_i} (\mathbf{M}^{(k)})^{-1} \tilde{\mathbf{z}} - \text{Tr} \left(\frac{\partial \mathbf{D}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_i} (\mathbf{M}^{(k)})^{-1} \right) \right). \quad (6)$$

The exact form of the second derivative of the log-likelihood with respect to the i -th and l -th parameters is,

$$\begin{aligned} \mathbf{H}_{il}^{(k+1)} &= \frac{1}{2}n \left[\text{Tr} \left(\frac{\partial^2 \mathbf{D}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_i \partial \boldsymbol{\theta}_l} (\mathbf{M}^{(k)})^{-1} \right) - n \text{Tr} \left(\frac{\partial \mathbf{D}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_i} (\mathbf{M}^{(k)})^{-1} \frac{\partial \mathbf{D}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_l} (\mathbf{M}^{(k)})^{-1} \right) \right] \\ &\quad - \frac{1}{2}n \left[\tilde{\mathbf{z}}^\top \left((\mathbf{M}^{(k)})^{-1} \frac{\partial^2 \mathbf{D}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_i \partial \boldsymbol{\theta}_l} (\mathbf{M}^{(k)})^{-1} - 2n (\mathbf{M}^{(k)})^{-1} \frac{\partial \mathbf{D}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_i} (\mathbf{M}^{(k)})^{-1} \frac{\partial \mathbf{D}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_l} (\mathbf{M}^{(k)})^{-1} \right) \tilde{\mathbf{z}} \right]. \end{aligned} \quad (7)$$

The second line is easy to evaluate, but not the first line, which has two terms. Recall that $\tilde{\mathbf{z}} \sim N(0, \mathbf{M})$. Thus, we can use the trace trick for the expectation of a quadratic form and re-write the second term as the following,

$$-n^2 \text{Tr} \left(\frac{\partial \mathbf{D}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_i} (\mathbf{M}^{(k)})^{-1} \frac{\partial \mathbf{D}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_l} (\mathbf{M}^{(k)})^{-1} \right) = -n^2 \mathbb{E} \left(\tilde{\mathbf{z}}^\top (\mathbf{M}^{(k)})^{-1} \frac{\partial \mathbf{D}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_i} (\mathbf{M}^{(k)})^{-1} \frac{\partial \mathbf{D}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_l} (\mathbf{M}^{(k)})^{-1} \tilde{\mathbf{z}} \right).$$

Similarly, we can approximate the first term in as the following,

$$n \text{Tr} \left(\frac{\partial^2 \mathbf{D}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_i \partial \boldsymbol{\theta}_l} (\mathbf{M}^{(k)})^{-1} \right) = n \mathbb{E} \left(\tilde{\mathbf{z}}^\top (\mathbf{M}^{(k)})^{-1} \frac{\partial^2 \mathbf{D}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_i \partial \boldsymbol{\theta}_l} (\mathbf{M}^{(k)})^{-1} \tilde{\mathbf{z}} \right)$$

36 Next, we adopt the same trick as used in BOLT-REML⁵, replacing the expected information with the
 37 observed information. With this approximation, the terms for the first and second line of equation (7) can
 38 be canceled out, which leads to the following,

$$\mathbf{H}_{il}^{(k+1)} \approx \frac{1}{2}n^2 \tilde{\mathbf{z}}^\top \left((\mathbf{M}^{(k)})^{-1} \frac{\partial \mathbf{D}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_i} (\mathbf{M}^{(k)})^{-1} \frac{\partial \mathbf{D}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_l} (\mathbf{M}^{(k)})^{-1} \right) \tilde{\mathbf{z}}, \quad (8)$$

39 ***Trust-region algorithm***

40 We use the trust-region algorithm to control the step size of each update in a principled way, and employ an
41 adaptive bound on the maximum change at each iteration (**Algorithm 1**). This allows us to balance between
42 convergence speed and robustness of the updates. We referenced the trust region iterative optimization in
43 Loh et al.⁶ in developing our own algorithm, but modified several aspects of the procedure for graphREML.
44 The main differences are that: 1) we compute the actual log-likelihood for the decision rule to accept or
45 reject the step size; 2) we do not impose the constraint related to parameter domain, because our link
46 function automatically leads to non-negative per-SNP heritability and thus a valid covariance matrix (for
47 the effect sizes); 3) we do not explicitly *optimize* the step size by maximizing the predicted change in log
48 likelihood according to the local quadratic model since this optimization can incur further computational
49 cost; instead, we simply use the closed-form step size similar to that used in Newton Raphson.

50 A key quantity that is central to both step size acceptance and trust region radius update is the
51 ratio between the actual and the predicted change of log likelihood, $\rho = \Delta_{actual}/\Delta_{pred}$. We use the
52 hyperparameter values recommended by Gould et al.⁷ to set up the lower and upper bounds, denoted by $\underline{\rho}$
53 and $\bar{\rho}$ respectively. These parameters, along with the radius change rate μ_{TR} , determine the updating of
54 the trust region radius adjustment.

55 The trust region algorithm is embedded within the Newton Raphson algorithm, such that each update
56 involves a step size adaptation. Therefore, each trust-region algorithm takes the current values of the
57 parameters as input and output the updated parameter values, along with the selected step size. Note that
58 the trust region radius is also adjustable, which is passed down from one iteration to the next. In addition,
59 we adopt the safeguard procedure proposed in Loh et al.⁶, rejecting step sizes that lead to an updated
60 gradient whose norm is more than double the norm of the gradient evaluated at the current parameter
61 values.

62 ***Convergence Criterion***

63 We stop the Newton Raphson algorithm either when the maximum number of updates have reached or
64 if the convergence criterion is triggered. We use a rather stringent threshold to determine convergence:

Algorithm 1: Trust region algorithm for step size selection

```
Preset :  $\mu_{TR} = 10$ ;  $\rho = 10^{-4}$ ;  $\bar{\rho} = 0.99$ ;  $K = 20$ 
Input :  $\theta^k$ 
Init :  $\lambda_{TR} = 10^{-3}$ ;  $k = 0$ ; accept=0
1 while  $k < K$  and accept is 0 do
2    $k \leftarrow k + 1$ ;
3    $\tilde{\mathbf{H}} \leftarrow \mathbf{H}(\theta^k) + 0.01 \cdot \lambda_{TR}(\mathbf{I} \odot \mathbf{H}(\theta^k))$ ;
4    $\mathbf{s} \leftarrow \tilde{\mathbf{H}}^{-1} \nabla(\theta^k)$ ; /* Compute the proposed step size */
5    $\tilde{\theta} \leftarrow \theta^k - \mathbf{s}$ ;
6    $\Delta_{actual} \leftarrow \ell(\theta^k) - \ell(\tilde{\theta})$ ;
7    $\Delta_{pred} \leftarrow \mathbf{s}^\top \nabla(\theta^k) - \frac{1}{2} \mathbf{s}^\top \mathbf{H}(\theta^k) \mathbf{s}$ 
8    $\rho = \Delta_{actual} / \Delta_{pred}$ ; /* Evaluate the step size */
9   if  $\rho > \rho$  then
10    |  $\theta^{k+1} \leftarrow \tilde{\theta}$ ; accept = 1;
11  else
12    |  $\theta^{k+1} \leftarrow \tilde{\theta}$ ; /* Accept or reject the step size */
13  end
14  if  $\rho < \rho$  then
15    |  $\lambda_{TR} = \lambda_{TR} \cdot \mu_{TR}$ ;
16  else
17    | if  $\rho > \bar{\rho}$  then
18      |  $\lambda_{TR} = \lambda_{TR} / \mu_{TR}$ ; /* Update the trust region parameters */
19    | end
20  end
21 end
Output :  $\theta^{k+1}$ ;  $\mathbf{s}$ ;  $\lambda_{TR}$ ;
```

65 we set the maximum number of iterations to 50, and declares convergence of the algorithm when the
66 change of log-likelihood averaged over three consecutive iterations is less than 10^{-3} . The sensitivity of
67 the estimation results with respect to these hyperparameters can vary depending on the dataset and the size
68 of the problem (*i.e.*, number of parameters). These parameters can be adjusted easily if needed.

69 **Modeling the per-SNP heritability**

70 S-LDSC assumes an unrealistic linear relationship between the heritability of a SNP and its annotations,
71 leading for example to negative per-SNP heritability estimates. In contrast, graphREML can fit essentially
72 any heritability model, incorporating a flexible link function to map between the annotations of a SNP to
73 its heritability. We assume that \mathbf{D} is a diagonal matrix with non-negative diagonal elements that represent
74 the per-SNP heritability of SNPs in the model. In other words, we assume that the covariance of $\tilde{\mathbf{z}}$ has the
75 form,

$$\mathbf{M}(\boldsymbol{\theta}) = n \cdot \text{diag}(\sigma_1^2, \dots, \sigma_p^2) + \widehat{\mathbf{P}}, \quad \text{with} \quad \sigma_j^2 = g_{\boldsymbol{\theta}}(\mathbf{a}_j), \quad (9)$$

76 where \mathbf{a}_j denotes the vector of annotation values for SNP j , and $\boldsymbol{\theta}$ denotes the conditional enrichment
77 coefficients. $g_{\boldsymbol{\theta}}$ is a non-negative scalar-valued link function that we choose for estimation. graphREML
78 by default uses the softmax link function, $g(\mathbf{a}_j) = \log(1 + \exp(\mathbf{a}_j^{\top} \boldsymbol{\theta}))$, but a more general form of \mathbf{D} and
79 other options for the link function are possible.

80 Notably, the existing methods can be viewed as special cases of (9), in terms of their assumptions
81 about the covariance structure of the causal effect sizes. For example, S-LDSC assumes $\sigma_j^2 = \mathbf{a}_j^{\top} \boldsymbol{\theta}$, where
82 $\boldsymbol{\theta}$ is a vector of coefficients that determine the partitioned heritability and enrichment^{8,9}. SumHer varies
83 from S-LDSC in modeling σ_j^2 as $\mathbf{a}_j^{\top} \boldsymbol{\theta} q_j / Q$, where q_j is a weight that explicitly accounts for the frequency-
84 dependent and LD-dependent architecture, and Q is the normalizing constant such that $Q = \sum_{j=1}^p q_j$ ¹⁰.
85 It is evident from the modeling of σ_j^2 that the per-SNP heritability estimates from S-LDSC or SumHer
86 may be negative and thus invalid. In contrast, graphREML is guaranteed to produce valid non-negative
87 per-SNP heritability estimates, as long as a non-negative link $g(\cdot)$ is used.

88 **Numerical overflow**

One implementation detail associated with the softmax link function is that we observed an numerical
overflow issue when applying the softmax function. To avoid this problem, we rewrote the link as the

following:

$$\begin{aligned}
g_{\theta}(\mathbf{a}_j) &= \log(1 + \exp(\mathbf{a}_j^{\top} \boldsymbol{\theta})) \\
&= \log(1 + \exp(\mathbf{a}_{j[-]} \cdot \boldsymbol{\theta}[-] + \mathbf{a}_{j[+]} \cdot \boldsymbol{\theta}[+])) \\
&= \log\left(\frac{\exp(\mathbf{a}_{j[-]} \cdot \boldsymbol{\theta}[-]) + \exp(-\mathbf{a}_{j[+]} \cdot \boldsymbol{\theta}[+])}{\exp(-\mathbf{a}_{j[+]} \cdot \boldsymbol{\theta}[+])}\right) \\
&= \mathbf{a}_{j[+]} \cdot \boldsymbol{\theta}[+] + \log[\exp(\mathbf{a}_{j[-]} \cdot \boldsymbol{\theta}[-]) + \exp(-\mathbf{a}_{j[+]} \cdot \boldsymbol{\theta}[+])]. \tag{10}
\end{aligned}$$

where $[+]$ and $[-]$ indicate the subset of the parameters which are positive and negative, respectively. We implemented a version of the softmax function using equation (10), which guards against numerical overflow issues. Analogously, we implemented a more robust version of the link derivative as the following,

$$g'_{\theta}(\mathbf{a}_j) = \frac{\mathbf{a}_{j[-]} \cdot \exp(\mathbf{a}_{j[-]} \cdot \boldsymbol{\theta}[-])}{1 + \exp(\mathbf{a}_{j[-]} \cdot \boldsymbol{\theta}[-])} + \frac{\mathbf{a}_{j[+]}}{1 + \exp(-\mathbf{a}_{j[+]} \cdot \boldsymbol{\theta}[+])}. \tag{11}$$

89 We scale the link and the link gradient functions by one over the number of SNPs. This ensures that any
90 non-linear relationships between the SNP's annotation values and its heritability can be properly captured
91 (the softmax function becomes effectively linear when $\mathbf{a}_j^{\top} \boldsymbol{\theta}$ is very large).

92 **Differentiability**

93 We note that it is important for the link function to be differentiable everywhere, because our estimation
94 algorithm, the gradient of the likelihood function in particular, entails the first derivative of the link
95 function. As we will discuss below, performing a score test for the conditional enrichment of an annotation
96 requires taking the second derivative of the link function. For our default link function – the softmax – we
97 have $\frac{\partial g(\mathbf{a}_j)}{\partial \theta_k} = \frac{\mathbf{a}_{j,k} \exp(\mathbf{a}_j^{\top} \boldsymbol{\theta})}{1 + \exp(\mathbf{a}_j^{\top} \boldsymbol{\theta})}$, and $\frac{\partial^2 g(\mathbf{a}_j)}{\partial \theta_k \partial \theta_l} = \frac{\mathbf{a}_{j,k} \mathbf{a}_{j,l} \exp(\mathbf{a}_j^{\top} \boldsymbol{\theta})}{(1 + \exp(\mathbf{a}_j^{\top} \boldsymbol{\theta}))^2}$. Other smooth candidate link functions include the
98 logistic function, $g(\mathbf{a}_j) = \frac{1}{1 + \exp(-\mathbf{a}_j^{\top} \boldsymbol{\theta})}$.

99 **Score test for inference on the joint enrichment**

100 Here we provide more details on the score test procedure we use for inference on the joint enrichment of a
 101 new annotation, conditional on a set of baseline annotations. The key observation that enables us to derive
 102 this test is that the score contributed from a given SNP can be re-written via chain rule into two parts –
 103 one part is the derivative of the likelihood with respect to the per-SNP heritability; the other part is the
 104 derivative of per-SNP heritability with respect to the parameters. In particular, when evaluated at the null,
 105 the score involves the new annotation values only through the second part.

106 Let $\hat{\theta} = (\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_K)$ denote the set of K parameter estimates in the baseline model. Let $\theta^* = (\hat{\theta}, 0)$
 107 denote the parameters fitted under the null *i.e.*, with baseline annotations included and constraining the
 108 conditional enrichment of the new $K + 1$ -th annotation to be zero. The score contributed from SNP j can
 109 be written as the following,

$$\begin{aligned}
 U_{j,K+1}(\theta^*) &= \frac{\partial \ell}{\partial \sigma_j^2} \cdot \frac{\partial \sigma_j^2}{\partial \theta_{K+1}} \Big|_{\theta=\theta^*} \\
 &= \mathbf{a}_{j,K+1} \left[g'(\mathbf{a}_j^\top \theta) \frac{\partial \ell}{\partial g(\mathbf{a}_j^\top \theta)} \Big|_{\theta=\theta^*} \right]_j && \text{(chain rule)} \\
 &= \mathbf{a}_{j,K+1} \underbrace{\left[g'(\mathbf{a}_j^\top \theta) \frac{\partial \ell}{\partial g(\mathbf{a}_j^\top \theta)} \Big|_{\theta=\hat{\theta}} \right]_j}_{\text{SNP-specific gradient obtained from the null fit}} && (\theta_{K+1}^* = 0 \text{ and thus } \mathbf{a}_j^\top \theta^* = \mathbf{a}_j^\top \hat{\theta}) \\
 &= \mathbf{a}_{j,K+1} \nabla_j(\hat{\theta}) && (12)
 \end{aligned}$$

110 Re-writting the score as equation (12) enables us to separate out the part that solely relies on the null fit
 111 and the part that entails the new annotation, which is the basis for the score test we developed.

112 The procedure we propose is as follows:

- 113 1. Fit graphREML under the null model (with K baseline annotations) to obtain the SNP-specific
 114 gradients, $\nabla_j(\hat{\theta})$ for $j = 1, 2, \dots, p$.

115 2. For any new annotation \mathbf{a}_{K+1} , construct the following statistics,

$$S_{K+1} = \frac{U_{K+1}(\boldsymbol{\theta}^*)^2}{\text{Var}(U_{K+1}(\boldsymbol{\theta}^*))} \quad (13)$$

116 where $U_{K+1}(\boldsymbol{\theta}^*) = \sum_j \mathbf{a}_{j,K+1} \nabla_j(\hat{\boldsymbol{\theta}})$ is the score aggregated from all markers.

117 3. Compare the score statistics against the null distribution $S_{K+1} \stackrel{H_0}{\sim} \chi^2(1)$ to compute the p-value for
118 the enrichment of this new annotation.

119 **Test of multiple new annotations**

120 More generally, we can adopt a similar procedure to test the significance of multiple new annotations at the
121 same time, conditional on the baseline annotations. To jointly test the significance of L new annotations
122 conditional on the K baseline annotations, we can construct the following test statistics,

$$S = U(\boldsymbol{\theta}^*)^\top \text{Cov}(U(\boldsymbol{\theta}^*)) U(\boldsymbol{\theta}^*) \quad (14)$$

123 where $U(\boldsymbol{\theta}^*) = [U_{K+1}(\boldsymbol{\theta}^*), \dots, U_{K+L}(\boldsymbol{\theta}^*)]^\top$. Under the null, $S \stackrel{H_0}{\sim} \chi^2(L)$, with which we can compute the
124 p-value for the joint enrichment of these L annotations.

125 **Jackknife covariance estimator**

We adopt a similar procedure as that used in the Wald test to derive a jackknife estimator of the SE. We
take advantage of the LD block structure to calculate the empirical covariance of the scores as the plug-in
variance for the score statistic, First, we compute a set of leave-one-LD-block scores,

$$J_{K+1}^b(\boldsymbol{\theta}^*) = U_{K+1}(\boldsymbol{\theta}^*) - \sum_{j \in b} \mathbf{a}_{j,K+1} \nabla_j(\hat{\boldsymbol{\theta}}), \quad \forall b = 1, 2, \dots, B$$

Then, we use the empirical distribution of the jackknife scores to construct the score test,

$$\frac{1}{B-2} \frac{(\sum_b J_{K+1}^b(\boldsymbol{\theta}^*)/B)^2}{\text{Var}(J_{K+1}^b(\boldsymbol{\theta}^*))} \stackrel{H_0}{\sim} \chi^2(1).$$

126 **Accounting for the uncertainty in $\hat{\theta}$**

127 We note that $\hat{\theta}$ is not perfectly estimated, which can affect the score test in two ways – one is that the
 128 estimates upon termination of the Newton updates may not be the actual solution to the score equations or
 129 the maximizer of the likelihood; the other is that the estimation noise needs to be accounted for through
 130 the plug-in variance estimator of the score statistic.

First, we need to account for the fact that the likelihood function evaluated at $\hat{\theta}$ is not the actual maximum even though it may be sufficiently close, because we stop the Newton updates using a pre-specified convergence criterion. Denote by $\tilde{\theta}$ the true solutions to the score equations under the null. Then by definition,

$$U_k(\theta)|_{\theta=\tilde{\theta}} = \sum_{j \in [p]} \mathbf{a}_{j,k} \nabla_j(\tilde{\theta}) = 0, \forall k = 1, 2, \dots, K \quad (15)$$

which is equivalent to $\mathbf{a}_k \perp \nabla(\tilde{\theta})$, for all $k = 1, 2, \dots, K$. In practice, the estimate we obtain from the null fit $\hat{\theta}$ may be arbitrarily close but is not exactly $\tilde{\theta}$. We adjust the SNP-specific gradients by projecting $\nabla(\hat{\theta})$ onto the null space expanded by the baseline annotation matrix \mathbf{A} ,

$$\nabla(\tilde{\theta}) \approx (\mathbf{I} - \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top) \nabla(\hat{\theta}) \quad (16)$$

131 Using the right-hand side of equation (16) ensures that the estimating equations under the null (15)
 132 indeed hold. In practice, we observed that this adjustment leads to different degrees of modification
 133 to the gradients. We note that it may be useful and of future interest to develop a statistic based on
 134 $\mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \nabla(\hat{\theta})$ for diagnostics of convergence.

135 Second, we want to account for the variance of $\hat{\theta}$ in order to develop an *efficient* score test. This
 136 affects the denominator of the score statistics in equation (13). Denote by $\hat{\theta}^{-b}$ the jackknife estimate of
 137 the parameters with the b -th block excluded. Recall that these values are readily available as they are used

138 to compute the jackknife estimator of the SE. Applying the jackknife variance leads to

$$\text{Var}(U_{K+1}(\boldsymbol{\theta}^*)) = (B - 1)\text{Var}(U_{K+1}(\hat{\boldsymbol{\theta}}^{-b})), \quad (17)$$

where B is the number of LD blocks or jackknife estimates, and $U_{K+1}(\hat{\boldsymbol{\theta}}^{-b}) = \sum_{j \notin b} \mathbf{a}_{j,K+1} \nabla_j(\hat{\boldsymbol{\theta}}^{-b})$ is the score aggregated from all except the SNPs on block b . However, computing $\nabla_j(\hat{\boldsymbol{\theta}}^{-b})$ for every single LD block is costly and unwieldy. We propose approximating the jackknife gradient as,

$$\nabla_j(\hat{\boldsymbol{\theta}}^{-b}) \approx \nabla_j(\hat{\boldsymbol{\theta}}) + (\hat{\boldsymbol{\theta}}^{-b} - \hat{\boldsymbol{\theta}}) \cdot \left. \frac{\partial \nabla_j(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}, \quad (18)$$

where the second term serves to explicitly account for the uncertainty in estimating $\hat{\boldsymbol{\theta}}$. Importantly, both terms in equation 18 can be obtained from the null fit without involving new annotation we want to test. Now consider the derivative in the second term. Without loss of generality, consider the partial derivative with respect to the first coefficient, which is the first element of the vector $\frac{\partial \nabla_j(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$. We have

$$\frac{\partial \nabla_j(\boldsymbol{\theta})}{\partial \theta_1} = \mathbf{a}_{j,1} \left[g''(\mathbf{a}_j^\top \boldsymbol{\theta}) \ell'_j + \left(g'(\mathbf{a}_j^\top \boldsymbol{\theta}) \right)^2 \ell''_j \right]$$

where we use ℓ'_j, ℓ''_j short for $\frac{\partial \ell}{\partial g(\mathbf{a}_j^\top \boldsymbol{\theta})}$ and $\frac{\partial^2 \ell}{\partial (g(\mathbf{a}_j^\top \boldsymbol{\theta}))^2}$, respectively. This enables us to compute a jackknife

score (aggregated at the LD level) that incorporates uncertainty in the estimation of $\hat{\theta}$.

$$\begin{aligned}
\widehat{Var}(U_{K+1}(\hat{\theta}^{-b})) &= \widehat{Var}\left(\sum_{j \notin b} \mathbf{a}_{j,K+1} \nabla_j(\hat{\theta}^{-b})\right) \\
&\approx \widehat{Var}\left(\sum_{j \notin b} \mathbf{a}_{j,K+1} \left[\nabla_j(\hat{\theta}) + (\hat{\theta}^{-b} - \hat{\theta}) \cdot \frac{\partial \nabla_j(\theta)}{\partial \theta} \Big|_{\theta=\hat{\theta}} \right]\right) \\
&\quad \text{(first-order approximation by equation 18)} \\
&= \frac{1}{B-2} \sum_{b=1}^B (\mathbf{V}_b - \overline{\mathbf{V}_b})^2, \quad \text{(sample variance of } \mathbf{V}_b \text{ across } B \text{ blocks)} \\
\text{where } \mathbf{V}_b &= \left(\sum_{j \notin b} \mathbf{a}_{j,K+1} \left[\nabla_j(\hat{\theta}) + (\hat{\theta}^{-b} - \hat{\theta})^\top \mathbf{a}_j \underbrace{\left[g''(\mathbf{a}_j^\top \hat{\theta}) \ell'_j + (g'(\mathbf{a}_j^\top \hat{\theta}))^2 \ell''_j \right]}_{\text{denote by } \mathbf{H}_{jj}, \text{ which is a scalar}} \right] \right) \quad (19)
\end{aligned}$$

139 where we have used the empirical variance across the jackknife scores as the plug-in variance.

To facilitate the computation of the scores, we re-write the leave-one-LD-block score in equation (19) using the block-specific scores, such that each SNP-specific score gets aggregated once.

$$\begin{aligned}
U_{K+1}(\hat{\theta}^{-b}) &= \sum_{j \notin b} \mathbf{a}_{j,K+1} \left(\nabla_j(\hat{\theta}) + \mathbf{H}_{jj}(\hat{\theta}^{-b} - \hat{\theta})^\top \mathbf{a}_j \right) \\
&= U_{K+1} - \sum_{j \in b} \mathbf{a}_{j,K+1} \nabla_j(\hat{\theta}) + \sum_{j \notin b} \mathbf{a}_{j,K+1} \mathbf{H}_{jj}(\hat{\theta}^{-b} - \hat{\theta})^\top \mathbf{a}_j \\
&\quad \text{(Define } U_{K+1} = \sum_{j \in [p]} \mathbf{a}_{j,K+1} \nabla_j(\hat{\theta})) \\
&= U_{K+1} - \sum_{j \in b} \mathbf{a}_{j,K+1} \nabla_j(\hat{\theta}) + (\hat{\theta}^{-b} - \hat{\theta})^\top \left[\sum_{j \in [p]} \mathbf{a}_{j,K+1} \mathbf{H}_{jj} \mathbf{a}_j - \sum_{j \in b} \mathbf{a}_{j,K+1} \mathbf{H}_{jj} \mathbf{a}_j \right], \quad (20)
\end{aligned}$$

140 where U_{K+1} is the overall score based on the original parameter estimate (*i.e.*, no jackknife).

141 **Memory cost**

142 In order to perform the score test, we save the following quantities from the null fit: $\nabla_j(\hat{\theta})$, \mathbf{H}_{jj} , both of
143 which are vectors of a length that equals to the number of markers. To perform the score test (using the
144 variance estimator that accounts for the uncertainty in $\hat{\theta}$, we also need the annotation matrix for both the
145 baseline annotations and the new annotation for testing.

146 Note that the cost of calculating the score statistics is linear in the number of markers and in the
 147 number of LD blocks, so this test procedure is fast. In terms of memory requirement, $\nabla_j(\hat{\theta})$ is a vector of
 148 the same length of as a single annotation, and $\left. \frac{\partial \nabla_j(\theta)}{\partial \theta} \right|_{\theta=\hat{\theta}}$ has the same size as the annotation matrix (*i.e.*,
 149 number of markers by number of baseline annotations), which is equivalent to 2 annotations. Therefore,
 150 our inference procedure does not increase the memory requirement of graphREML drastically.

151 **Application of graphREML to AMM**

152 **Link functions**

153 To explicitly model the mediated heritability by the nearest genes in a given gene set, we write per-SNP
 154 heritability as the following,

$$\sigma_j^2 = f(\boldsymbol{\theta}^\top \mathbf{b}_j) \circ \left(1 + f(\boldsymbol{\gamma})^\top \mathbf{a}_j \right), \quad (21)$$

where we use $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$ to denote the vectors of parameters for the baseline and nearest gene annotations. We
 compute the first derivative of the link with respect to the baseline parameters and the AMM parameters
 separately as the following:

$$\begin{aligned} \frac{\partial \sigma_j^2}{\partial \theta_l} &= \left(1 + f(\boldsymbol{\gamma})^\top \mathbf{a}_j \right) f'(\boldsymbol{\theta}^\top \mathbf{b}_j) \mathbf{b}_j^l \\ \frac{\partial \sigma_j^2}{\partial \gamma_k} &= f(\boldsymbol{\theta}^\top \mathbf{b}_j) f'(\boldsymbol{\gamma}) \mathbf{a}_j^k \end{aligned}$$

155 **Comparison to the original AMM**

The original AMM work models the per-SNP heritability as the following,

$$\mathbb{E}(\beta_j^2) = \tau(0) + \tau(A) \sum_k p^{(k)} \mathbf{a}_j^{(k)} = \tau(0) \left(1 + \sum_k \frac{\tau(A)}{\tau(0)} p^{(k)} \mathbf{a}_j^{(k)} \right) = \tau(0) \left(1 + \sum_k \frac{\tau^{(k)}}{\tau(0)} \mathbf{a}_j^{(k)} \right)$$

156 To estimate p_k , one first obtains estimates of $\tau^{(k)}$; then we estimate p_k as $\frac{\tau^{(k)}}{\sum_l \tau^{(l)}}$.

With the new links defined in the application of graphREML to AMM, we have,

$$\mathbb{E}(\beta_j^2) = f(\boldsymbol{\theta}^\top \mathbf{b}_j) \left(1 + \sum_k f(\gamma_k) \mathbf{a}_j^{(k)} \right).$$

157 Analogously, we first obtain estimates of γ_k ; then estimate p_k with the link applied as $\frac{f(\gamma_k)}{\sum_k f(\gamma_k)}$. The key
158 differences between the two models are that 1) graphREML enables the baseline per-SNP heritability to
159 be variant-specific; 2) graphREML incorporates a non-negative mapping $f(\cdot)$ to ensure the non-negativity
160 of the heritability and to enable a non-linear relationship with the annotations.

References

- 161 **1.** Conneely, K. N. & Boehnke, M. So many correlated tests, so little time! rapid adjustment of p values
162 for multiple correlated tests. *The Am. J. Hum. Genet.* **81**, 1158–1168 (2007).
163
- 164 **2.** Zhu, X. & Stephens, M. Bayesian large-scale multiple regression with summary statistics from
165 genome-wide association studies. *The annals applied statistics* **11**, 1561 (2017).
- 166 **3.** Ning, Z., Pawitan, Y. & Shen, X. High-definition likelihood inference of genetic correlations across
167 human complex traits. *Nat. genetics* **52**, 859–864 (2020).
- 168 **4.** Campbell, Y. E. & Davis, T. A. Computing the sparse inverse subset: an inverse multifrontal approach.
169 *Univ. Florida, Tech. Rep. TR-95-021* (1995).
- 170 **5.** Loh, P.-R. *et al.* Contrasting genetic architectures of schizophrenia and other complex diseases using
171 fast variance-components analysis. *Nat. genetics* **47**, 1385 (2015).
- 172 **6.** Loh, P.-R. *et al.* Efficient bayesian mixed-model analysis increases association power in large cohorts.
173 *Nat. genetics* **47**, 284–290 (2015).
- 174 **7.** Gould, N. I., Orban, D., Sartenaer, A. & Toint, P. L. Sensitivity of trust-region algorithms to their
175 parameters. *4OR* **3**, 227–241 (2005).
- 176 **8.** Bulik-Sullivan, B. K. *et al.* Ld score regression distinguishes confounding from polygenicity in
177 genome-wide association studies. *Nat. genetics* **47**, 291–295 (2015).
- 178 **9.** Finucane, H. K. *et al.* Partitioning heritability by functional annotation using genome-wide association
179 summary statistics. *Nat. genetics* **47**, 1228–1235 (2015).
- 180 **10.** Speed, D. & Balding, D. J. Sumher better estimates the snp heritability of complex traits from
181 summary statistics. *Nat. genetics* **51**, 277–284 (2019).