SUPPLEMENTAL MATERIAL

Additional Information on Algorithmic and Computational Acceleration of Standard Genetic Models for Imaging Genomics Analyses

Introduction to standard and accelerated genetic analyses

Computational genetic analyses assume that phenotypic variability (σ^2_P) in quantitative traits can be attributed to additive genetic (σ^2_G) variance that includes variance due to polymorphic and other genetic variations and environmental causes (σ^2_E). They attempt to quantify the degrees of contribution from these sources of variation. Genetic analyses in related populations (it is wise to assume that even unrelated subjects have a non-zero degree of relatedness) use a linear mixed model to account for genetic relatedness (non-independence) among the participants. For simplicity, we will assume that effects of fixed covariates such as age, sex, batch, imaging modality, etc. were removed at the point of data processing and the data were inverse-normalized such that each imaging (in this case, "voxel-wise") trait follows a multivariate normal distribution[127]. This supplement reviews the standard computational genetic model based on maximization of likelihood and identifies the performance bottlenecks that can prevent its use in studies performing genetic analyses of rich imaging data. It then presents multiple approaches to accelerate such genetic analyses while maintaining the fidelity of the standard approach.

We start by characterizing the degree of shared genetic variance among the participants in any given sample. The information of this pair-wise relatedness among the participants is represented as Φ, an *N*x*N* matrix of coefficients of relationships (CR). Φ is also known as the pedigree, relatedness or kinship matrix. Traditionally, elements of the Φ matrix are ascertained from self-reported relationships. For example, a pair of individuals may identify themselves as parent and child, siblings, cousins, etc. Self-reported CR values are calculated as the length of the shortest ancestral path (kinship) between two individuals. Each kinship type is a fixed number: 1 for the self and a monozygotic twin; ½ for parents, full siblings and dizygotic twins; ¼ for grandparents or half-siblings; 1/8 for cousins; and 0 for unrelated individuals. The self-reported CR codes the *expected* degree of shared genomic variance for a kinship type. However, in practice, identical twins share less than 100% and no two siblings share exactly 50% of the genome-wide genetic polymorphisms[150,151]. Moreover, seemingly *unrelated* individuals often can share a significant degree of genetic variance, known as cryptic relatedness. A more practical alternative is to measure CRs empirically from the same high-throughput genome-wide scans used in genome-wide association (GWA) studies[152-155]. The empirical pedigree matrix is denser, has fewer zeros, and leads to more accurate and stable estimates of heritability and association than self-reported Φ matrix[127]. Therefore, it is recommended to use empirical relatedness whenever participants' genomic panel data are available, as opposed to CRs[127,152-155]. In the next sections, we will present the standard and accelerated computational models that use Φ for genetic analyses with the focus of making genetic analysis of high-dimensional imaging phenotypes feasible and practical.

Standard genetic model

Standard computational genetics models to measure heritability and association are based on the variance component analyses and iterative maximization of the likelihood estimation (MLE). This approach originated by Ronald Fisher's ("the father of computational genetics") foundational work and has served as the standard framework for human and animal pedigree analyses[156,157]. It uses a linear mixed effect modeling and MLE based statistical inference to compare models with fixed (e.g., the effects of relatedness within the pedigree and genetic variants on the trait) and random variance of the trait. This can be written as equation 1, where phenotype Y, such a brain volume or Hounsfield unit of a cardiac CT voxel per subject, is

coded by a vector of length of N (N is the number of subjects) and is assumed to be due to composed of genetic Yg and environmental Ye parts.

$$Y = Yg + Ye \quad (1)$$

Therefore, variance in the trait vector Y can be written as equation 2.

$$\text{Variance}[Y] = \text{Variance}[Yg + Ye] \quad (2)$$

The variance parameters can be numerically estimated by comparing the observed covariance matrix Ω of Y with the covariance matrices predicted by matrix Φ composed of CR values and identity matrix I that codes for random regimental variance[157] (equation 3).

$$\Omega = 2 \cdot \Phi \cdot \sigma_g^2 + I \cdot \sigma_e^2 \quad (3)$$

σ_e^2 is the variance due to individual-specific environmental effects under the assumption that all environmental effects are uncorrelated among family members (e.g., coded by identity matrix). The model in equation 3 can be rewritten for association analysis by including the testing of the variance from a single nucleotide variant (SNV) and its beta coefficient (equation 4), where the SNV_j is a vector (dimension N) of genetic variability for this SNV for each subject, usually coded as 0, 1, or 2 indicating allelic frequency.

$$Y = SNV_j \cdot \beta_j + Yg + Ye \quad (4)$$

This can be rewritten in form of the variance as

$$\text{Var}[Y - SNV_j \cdot \beta_j] = \text{Var}[Yg + Ye] = \Phi \cdot h^2 + I \cdot (1 - h^2) \quad (5)$$

Equation 5 is a form of equation 3 that was rewritten by using h^2, which is the standard additive genetic heritability or $h^2 = \sigma^2 g/(\sigma^2 g + \sigma^2 e)$. The significance of the genetic contribution is tested by comparing the likelihood of the model in which σ_g^2 is set to zero with that of a model in which σ_g^2 is estimated. Twice the difference between the $\log_e$ likelihoods of these models yields a test statistic, which is asymptotically distributed as a 1/2:1/2 mixture of a χ^2 variable with one degree-of-freedom and a point mass at zero. The logarithmic function (l) of the likelihood (L) in equations 4 and 5 can be written as

$$l = \ln L = -\tfrac{1}{2} [N \cdot \ln 2\pi + \ln (\Phi \cdot h^2 + I \cdot (1 - h^2)) + \delta' \cdot (\Phi \cdot h^2 + I \cdot (1 - h^2)))^{-1} \cdot \delta] \quad (6)$$

where $\delta = Y - SNV_i \cdot \beta_i$. The statistical significance of association is tested by comparing the likelihoods of the model with β_i constrained to 0 to the unconstrained model.

For imaging genetic studies, this model (equation 6) needs to be evaluated for every trait and every polymorphism. A typical voxel-wise analysis involves ~100,000 traits. Heritability analyses would require 100,000 maximization of likelihood that can take between 10-50 iterations of a typical MLE algorithms. A GWA study of 1,000,000 SNVs would require 10^{11} maximization (number of traits times number of SNVs) of likelihood. This amounts to significant computational burden given the iterative nature of the MLE algorithms.

Each iteration of likelihood algorithm requires the inversion of the covariance matrix, $\Phi \cdot h^2 + I \cdot (1 - h^2)$ in equation 6. The computational effort associated with matrix inversion goes up

non-linearly ~N^2 for sparse, self-reported pedigrees and ~N^3 for dense empirical pedigrees, where N is the number of subjects. These analyses become astronomically complex when using empirical CR matrices in studies such as UK Biobank that has over N=500,000 participants. The standard computational genetics model has served us well for over six decades but it is not practical for massive computational needs required for imaging genetics applications in the big data era. In the next two sections we discuss how such analyses can be made practical using algorithmic developments and parallel computing capabilities of modern hardware.

Algorithmic acceleration of standard genetic model

Minimizing matrix inversion burden

The first step is to liberalize the N^{2-3} dependence of the computational burden associated with the inversion of covariance matrix for each likelihood calculation in equation 6. The eigen value decomposition (EVD) approach[128] performs an orthogonal transformation that diagonalizes the covariance ($\Phi \cdot h^2 + I \cdot (1- h^2)$ matrix to render the matrix inversion trivial. This transform maps the vector Y of non-independent observation to a vector Y^* of independent observations. The EVD of the covariance matrix, D_p can written as Equation 7.

$$V \cdot D_p \cdot V' = V \cdot [h^2 \cdot D_g + (1- h^2) \cdot I] \cdot V' = V \cdot [I + h^2 \cdot (D_g - I)] \cdot V' \qquad (7)$$

where V is the orthogonal matrix of eigenvectors and Dp and Dg are diagonal matrices of phenotypic and genetic eigen values λp and λg. This transformation decorrelates the data for related subjects and reduces the likelihood to the product of univariate normal densities[128]. If $\tau = V' \cdot \delta$ is the vector of residual phenotype values following the transformation to the eigen basis of the covariance matrix, then the likelihood equation becomes Equation 8, see[128] for derivations.

$$l = -\tfrac{1}{2} [N \cdot \ln 2\pi + \Sigma \ln (1+ h^2 \cdot (\lambda g_i - 1)) + \Sigma \tau_i^2 / (1+ h^2 \cdot (\lambda g_i - 1))] \qquad (8)$$

Comparing Equations 6 and 8, we can see that the likelihood calculations have been simplified to be a sum of univariate likelihoods. This comes with two benefits. The calculation of the inverse and transformed covariance is now trivial. The second benefit is that the simplified polygenic model can be reduced to simple algebraic solutions for fast approximation calculations that do not require iterative maximization of the likelihood. We present two such approximations to accelerate heritability and association calculations.

Non-iterative approximations: Two-step Fast and Powerful Heritability Inference (FPHI)

The solution in Equation 8 provides a precise estimate of model parameters while greatly reducing ~N^{2-3} burden associated with inversion of pedigree matrix. However, this solution is still iterative and requires recalculation of likelihood 10-50 times during the maximization convergence. We developed the FPHI solution for a two-step estimation of heritability. If we don't consider the variance associated with measured genotypes (β_i=0) then the functional form in equation 8 can be simplified as equation 9

$$l (\sigma^2_A, \sigma^2_E) = -\tfrac{1}{2}[N \cdot \ln 2\pi + \Sigma \ln (\sigma^2_g \cdot \lambda g_i + \sigma^2_e)) + \Sigma \tau_i^2 / (\sigma^2_g \cdot \lambda g_i + \sigma^2_e))] \qquad (9)$$

where, we again use the definition $h^2 = \sigma^2_g / (\sigma^2_g + \sigma^2_e)$, and the covariance matrix becomes $\sigma^2_g \cdot Dg + \sigma^2_e I$ where Dg is a diagonalized matrix of eigenvalues. We define θ as a 2-D vector = (σ^2_g, σ^2_e). In the standard MLE approach maximization of likelihood is achieved by solving for

the root value θ_{ML} where $l'(\theta_{ML}) = 0$ using the iterative Newton's method to achieve convergence to θ_{ML}

$$\theta_{n+1}=\theta_n - l'(\theta_n) \cdot l''(\theta_n)^{-1} \qquad (10)$$

where $l'(\theta_n)$ is the first and $l''(\theta_n)$ is the second derivatives of the log-likelihood function and n denotes the iteration number. Equation 10 requires inversion of 2x2 $l''(\theta_n)$ Hessian matrix at every iteration. While trivial for a single trait analysis, this can lead to significant computational effort for imaging genetic studies that utilize thousands to hundreds-of-thousands traits.

FPHI method uses a two-step Ordinary Linear Squares (OLS) followed by Weighted Linear Squares (WLS) approximation to solve equation 9 non-iteratively[125]. If we define a Nx2 matrix U as [1, λgi], this amounts to solving equation $U \cdot\theta=\varepsilon^{2\cdot}$ The OLS solution $\theta_{OLS}= (\sigma^2_{gOLS}, \sigma^2_{eOLS})$ is given by equation 11,

$$\theta_{OLS}= \max \{0 \text{ or } (U'\cdot U)^{-1}\cdot U'\cdot f_{OLS} \} \qquad (11)$$

where U is, [1, λgi], an Nx2 matrix, and ε^2 is the square of the residual $Y^*=V'\cdot Y$. We set the θ_{OLS} as the OLS $((U'\cdot U)^{-1}\cdot U'\cdot f_{OLS})$, unless it is negative, in which case θ_{OLS} is set to zero to ensure the non-negativity of the solution. This provides a vector and the corresponding h^2_{OLS} but it is not recommended as a final estimate[125]. Instead, the WLS solution given in equation 12 is preferred.

$$\theta_{WLS}= \max \{ 0 \text{ or } (U'\cdot((\sigma^2_{gOLS}Dg + \sigma^2_{eOLS}I)^2)^{-1}\cdot U'\cdot((\sigma^2_{gOLS}Dg + \sigma^2_{eOLS}I)^2)^{-1} \cdot \varepsilon^2_{OLS}) \} \quad (12)$$

This WLS estimator is asymptotically normal and unbiased[125]. The corresponding heritability estimate is given by equation 13,

$$h^2_{WLS} = \sigma^2_{AWLS}/(\sigma^2_{gWLS} + \sigma^2_{eWLS}) \qquad (13)$$

In evaluation in both simulated and real data, h^2_{WLS} provides an excellent approximation for h^2_{ML} as long as the data follows multivariate normal distribution and proposed harmonization strategies to improve agreement[125,158].

Non-iterative approximations: Two-step Fast and Powerful Genome-wide Association (FPGA)

The FPHI provided a two-step decomposition of the variance within a trait into additive genetic and environmental components. We expand this model for genotype association analysis of the full model (equation 14) that included measured genotypes[126]. The significance of the association model estimated by including a measured genotype term $SNP_j\cdot \beta_j$. into equation 11.

$$l(\beta_j, \sigma^2_A, \sigma^2_E) = -\tfrac{1}{2}[N\cdot\ln 2\pi + \Sigma \ln (\sigma^2_A \cdot \lambda g_i + \sigma^2_e)) + \Sigma\tau_i^2/(\sigma^2_A \cdot \lambda g_i +\sigma^2_e))] \quad (14)$$

where τ_I now equals to $V'(Y- SNP_j\cdot \beta_j)$, the eigenvalue transformation of the residual of the phenotype vector. The significance is evaluated using a likelihood test that evaluates two models. In the null model the β_j is set to 0. This is followed by evaluation of the unconstrained model and evaluation of significance using the log-likelihood ratio[126]. However, even two-step

approximation method becomes a considerable computation burden given the large number of association tests in a voxel-wise GWA studies given a large number of measured genotypes.

Non-iterative approximations: Single-step FPGA-Wald

Additional computational performance can be achieved by testing the significance of association in equation 14 using the Wald test. The Wald test is a classical hypothesis testing approach and is an alternative to MLE. Wald test is based on an asymptotic $\chi 2$-distribution to evaluate a distance between the unrestricted estimate and its hypothesized value under the null hypothesis. It provides a significant reduction in computation burden because the variance is calculated once per trait as opposed to once for each test of significance (e.g., each SNV) thus reducing it to a single step evaluation. Under the Wald test, the square of parameter estimate is divided by the variance of the effect of the SNV (β) under the unrestricted model. If we define Σ as a covariance matrix obtained by multiplying Nx2 vector of eigen values of the kinship matrix [1, λg] by 2x1 vector additive genetic and environmental variances obtained by FPHI [σ^2_A, σ^2_E]. Then we can obtain the Wald score (T_{wald}) using Equation 15.

$$T_{wald} = \beta_a (X' \Sigma^{-1} X)^{-1} \beta'_a \qquad (15)$$

where β_a is the SNV effect, X is the vector of SNV values obtained under the alternative hypothesis. We consider FPGI-Wald test as an "screening" technique because the Wald test uses an asymptomatic approximation, yet, empirical evaluations show excellent agreements for the results obtained using MLE and Wald tests.

In summary, the algorithmic approximation of the standard genetic model (i.e., FPHI, FPGI and FPGI-Wald) provides significant improvement in computational efficiency versus classical approaches. These approaches reduce computational complexity (from N^{2-3} to N^1), which make them especially valuable for big data studies. However, algorithmic approximations by themselves still do not make imaging genetic analyses practical.

Hardware acceleration of imaging genetics computations

The highly parallel and non-iterative nature of the FPHI and FPGI algorithms calls for efficient implementation using modern hardware optimized for massively parallel computations. Contemporary computational clusters are built of nodes equipped with central processing and graphics processing units (CPU/GPU) that offer multiple computational cores (typically 2-64 for CPUs and 1000-8000 for GPU). Each core can act as an independent computational unit that can access memory and perform calculations in parallel with other cores. GPUs make the parallel computing especially cost effective by offering thousands of computational cores on a single board that is equipped with dedicated high-speed memory. This provides higher computation power per unit cost ratio (a few cents vs. $100-500 per GPU vs CPU core, respectively). The CPU and GPU version of FPHI and FPGI can be implemented using linear algebra software libraries that optimize the code for parallel scientific computing in CPU and GPU environment. However, there are important caveats that makes CPU and GPU implementations different due to hardware differences between two computational devices.

Implementation using parallel CPU computing

CPU parallelization is based on the simultaneous multi-threading (SMT) processing where each CPU core acts as an independent processing unit with a rich instruction set capable of executing a sequence of instructions. It can be visualized by imaging running two instances of software independent of each other. The FPHI and FPGI algorithms were implemented for parallel analysis by using one trait and one SNP per one computational thread. That each thread performed a single FPHI analysis of additive genetic variance or a single FPGI/FPGI-

Wald analysis of association between a trait and SNP. The OpenMP (https://www.openmp.org) software library was used to implement thread-level parallelization. This library automatically handles the number of threads and spreads the load across available CPUs/core. Within a thread, linear algebra operation used by FPHI and FPGI algorithms, including vector-vector, matrix-vector and matrix-matrix operations were coded using Basic Linear Algebra Subprograms implemented in the Intel Math Kernel Library (https://software.intel.com/en-us/mkl). All algorithmic, software and hardware approaches discussed here are implemented in the solar-eclipse software (solar-eclipse-genetics.org) and are freely available for download, use and distribution.

Implementation using parallel GPU computing

The parallelization of scientific calculations for GPU differs from these used in CPU. A GPU card is optimized for single-instruction, multiple thread (SIMT) processing. SIMT was specifically developed for parallelization of vector operation. This includes such tasks as ray tracing and texture mapping where each GPU core performs a relatively simple task using a limited instruction set. To give an example of the difference between CPU and GPU approaches to parallel computing, let us consider a task of performing an addition operation of two vectors (arrays of data). The SMT approach is to execute a thread that contains the code for a loop-based addition the elements of two vectors, one element at a time. The code within that thread will be executed on a single core and parallel threads would be launched when additions of multiple vectors are required. The SIMT places the emphasis on the operational level parallelization. The GPU cores do not have a rich instruction set used in OpenMP operation and furthermore the memory exchange between CPU and GPU memory banks becomes a processing bottleneck. Instead, the GPU code for addition of two vectors would subdivide the operations to the level of singular operations, e.g., a summation of two elements of an array and issue this task in a form of a kernel of a GPU board. The addition of multiple vectors will therefore be performed sequentially, unless the kernel size allows for the full array command to be executed within the kernel. This level of planning and coordination is difficult to achieve for scientific programing. Typically, scientific algorithms must be redesigned from the ground up to adhere to SIMT architecture of GPU computing, thus making efficient porting of scientific algorithms challenging[159]. Fortunately, cuBLAS library provides a convenient alternative for porting code written for OpenMP environment for execution using the GPUs. It also handles parallelization across multiple GPU cards and manages the allocation of global (accessible across GPU), shared (accessible to all threads within a thread block) and register (accessible only to one thread) memory for the developer (https://developer.nvidia.com/cublas). The disadvantage of this approach is the vendor specific nature of cuBLAS that is only available for the devices that support Compute Unified Device Architecture.

Example of the performance evaluation in high-dimensional imaging data

The performance improvement provided by the aforementioned approaches was recently demonstrated through a collaboration with the Human Connectome Project (HCP) and UK Bio Bank (UKBB) projects. These voxel-wise analyses were performed in neuroimaging data, however, the underlying methods are applicable to any high-density phenotyping approaches including x-ray, computed tomography or 2-D expression arrays. The linearized computational burden was reduced from ~N^3 to N^1, leading to a ~10^6 improvement against standard MLE approaches and 10^{4-6} improvement against accelerated software like Scalable and Accurate Implementation of GEneralized (SAIGE). For example, the calculation of voxel-wise heritability map in approximately 117,000 voxels in N=1024 participants in HCP was reduced from ~100 hours for MLE approach to ~8 seconds using the FPHI-GPU algorithm executed on a single Tesla P100 GPU card (~180,000 times faster). In N=19,257 UKBB subjects, the standard MLE computation approach would take ~10^7 hours, while FPHI method can perform it in under 10

hours due to $N^3 \rightarrow N^1$ reduction in calculation burden (longer than predicted due to disk i/o of large datasets). A voxel-wise GWAS of ~$60 \cdot 10^3$ voxels and $6 \cdot 10^6$ SNVs that showed significant heritability in HCP voxelwise diffusion tensor imaging data, took 32,000 CPU hours using SAIGE software vs. only 80 CPU hours vs 3 hours using Tesla P100 GPU and Fast Permutation Genetic Inference (FPGI) approach with Wald significance testing (~10^3 improvement). In N=19K UKBB subjects, the performance gap vs SAIGE has increased to ~10^6. Together, these improvements made voxel-wise GWA studies practical in samples with complex *empirical* pedigrees including HCP and UKBB (N=10,000-50,000).