StocSum: Stochastic Summary Statistics Version 0.1.1

Contents

1 Introduction

StocSum is a novel reference-panel-free statistical framework for generating, managing, and analyzing stochastic summary statistics using random vectors. Summary statistical-based methods require information on the linkage disequilibrium (LD) or correlation structure between genetic variants, which is commonly derived from external reference panels. Current reference panels are usually Eurocentric biases, and it is usually difficult to find suitable external reference panels that represent the LD structure for underrepresented and admixed populations, or rare genetic variants from whole genome sequencing (WGS) studies, limiting the scope of applications for genomic summary statistics. Instead of from external reference panel, StocSum introduces a stochastic summary statistic from study samples to represent the between-variant correlation or LD matrices. StocSum can be implemented to various downstream applications, including single-variant tests, conditional association tests, gene-environment interaction tests, variant set tests, as well as meta-analysis and LD score regression tools.

2 The model

2.1 The full model

We describe StocSum under the generalized linear mixed model (GLMM) framework. It can also be applied to simpler statistical models such as generalized linear models and extended to more complex models such as generalized additive mixed models. The GLMM can be written as:

$$\mathbb{g}(\mu_i) = \boldsymbol{X}_i\boldsymbol{\alpha} + \widetilde{\boldsymbol{G}}_i\boldsymbol{\beta} + b_i \qquad (1)$$

where $\mathbb{g}(\cdot)$ is a monotonic link function of μ_i, and $\mu_i = E\left(y_i|\boldsymbol{X}_i, \widetilde{\boldsymbol{G}}_i, b_i\right)$ is the conditional mean of the phenotype y_i given p covariates $\boldsymbol{X}_i$, q genotypes $\widetilde{\boldsymbol{G}}_i$ and random effects b_i, for individual i of N samples. The phenotype y_i follows a distribution in the exponential family, such as a normal distribution for continuous phenotypes, or a Bernoulli distribution for binary phenotypes. Here $\boldsymbol{\alpha}$ is a length p column vector of fixed covariate effects including an intercept term. The genotype matrix $\widetilde{\boldsymbol{G}} = \left(\widetilde{\boldsymbol{G}}_1^T\ \widetilde{\boldsymbol{G}}_2^T\ \cdots \widetilde{\boldsymbol{G}}_N^T\right)^T$ is an $N \times q$ matrix for q ($q \geq 1$) genetic variants and $\boldsymbol{\beta}$ is a length q genotype effect vector. We assume that $\boldsymbol{b} = (b_1\ b_2\ \cdots\ b_N)^T$ is a length N column vector of random effects and $\boldsymbol{b} \sim \sum_{k=1}^{K} \tau_k \boldsymbol{\Phi}_k$, where τ_k are the variance component parameters and $\boldsymbol{\Phi}_k$ are known $N \times N$ dense or sparse relatedness matrices which account for multiple layers of correlation structures, such as genetic relatedness, hierarchical designs, shared environmental effects and repeated measures from longitudinal studies.

2.2 Stochastic summary statistics

Under the null hypothesis of no genetic fixed effects H_0: $\boldsymbol{\beta} = 0$, model (Eq.(1)) reduces to

$$\mathbb{g}(\mu_{0_i}) = \boldsymbol{X}_i\boldsymbol{\alpha} + b_i. \quad (2)$$

Here $\mathbb{g}(\cdot)$ is a monotonic link function of μ_{0_i}, and $\mu_{0_i} = E(y_i|\boldsymbol{X}_i, b_i)$ is the conditional mean of the phenotype y_i under the null hypothesis H_0: $\boldsymbol{\beta} = 0$, given p covariates $\boldsymbol{X}_i$ (including an intercept) and random effects b_i, for individual i of N samples. Let $\widehat{\boldsymbol{\mu}}_0 = (\hat{\mu}_{0_1}, \hat{\mu}_{0_2}, \dots, \hat{\mu}_{0_N})^T$ be a length N column vector for the estimated values of μ_{0_i}, $\hat{\phi}$ be an estimate of the dispersion parameter (or the residual variance for continuous traits in linear mixed models) ϕ, and $\hat{\tau}_k$ be the estimates for variance component parameters τ_k corresponding to $N \times N$ relatedness matrices $\boldsymbol{\Phi}_k$, from the null model (Eq.(2)), we define $\boldsymbol{P} = \widehat{\boldsymbol{\Sigma}}^{-1} - \widehat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{X}\left(\boldsymbol{X}^T\widehat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\widehat{\boldsymbol{\Sigma}}^{-1}$ as the projection matrix, where $\boldsymbol{X} = (\boldsymbol{X}_1^T\ \boldsymbol{X}_2^T\ \cdots\ \boldsymbol{X}_N^T)^T$ is a $N \times p$ covariate matrix, and $\widehat{\boldsymbol{\Sigma}} = \widehat{\boldsymbol{\Omega}}^{-1} + \sum_{k=1}^{K}\hat{\tau}_k\boldsymbol{\Phi}_k$ with $\widehat{\boldsymbol{\Omega}}^{-1} = \hat{\phi}\boldsymbol{I}_n$ for continuous traits in linear mixed models, and $\widehat{\boldsymbol{\Omega}}^{-1} = diag\left\{\frac{1}{\hat{\mu}_{0i}(1-\hat{\mu}_{0i})}\right\}$ for binary traits in logistic mixed models.

StocSum leverages a length N random vector $\boldsymbol{R}_b$ from a multivariate normal distribution with mean $\mathbf{0}$ and covariance matrix $\boldsymbol{P}$, repeats this simulation process B times and combines $\boldsymbol{R}_b$ $(1 \le b \le B)$ into an $N \times B$ random matrix $\boldsymbol{R} = (\boldsymbol{R}_1\ \boldsymbol{R}_2\ \cdots\ \boldsymbol{R}_B)$. In our implementation, we first decompose relatedness matrices $\boldsymbol{\Phi}_k = \boldsymbol{Z}_k\boldsymbol{Z}_k^T$, where $\boldsymbol{Z}_k$ is an $N \times L_k$ matrix $(L_k \le N)$. For low-rank relatedness matrices (such as those indicating observations from the same sample in longitudinal studies), $\boldsymbol{Z}_k$ is often known as the random effect design matrix, with L_k being the rank of $\boldsymbol{\Phi}_k$. For sparse block-diagonal relatedness matrices (such as positive definite kinship matrices), $\boldsymbol{Z}_k$ is the Cholesky decomposition of $\boldsymbol{\Phi}_k$, which is also sparse block-diagonal. We construct the $N \times B$ random matrix as $\boldsymbol{R} = \sqrt{\hat{\phi}}\boldsymbol{r}_0 + \sum_{k=1}^{K}\sqrt{\widehat{\tau_k}}\boldsymbol{Z}_k\boldsymbol{r}_k$, in which $\boldsymbol{r}_0$ is an $N \times B$ random matrix and $\boldsymbol{r}_k$ $(1 \le k \le K)$ are $L_k \times B$ random matrices, with all entries in $\boldsymbol{r}_0$ and $\boldsymbol{r}_k$ simulated from a standard normal distribution.

For an $N \times M$ genotype matrix $\boldsymbol{G}$ for M variants on the whole genome (or on one chromosome), the $M \times B$ stochastic summary statistic matrix $\boldsymbol{U}$ can be calculated as $\boldsymbol{U} = \boldsymbol{G}^T\boldsymbol{R}$. In the next sections, we describe how the stochastic summary statistics can be used in various downstream genetic analysis applications.

2.3 Single-variant tests

We are interested in conducting single-variant tests for the null hypothesis H_0: $\beta = 0$, using the score test. The GMMAT single-variant score is $S = \frac{\boldsymbol{g}^T(\boldsymbol{y}-\hat{\boldsymbol{\mu}}_0)}{\hat{\phi}}$, where $\boldsymbol{g} = (g_1\ g_2\ ...\ g_N)^T$ is a length N column genotype vector for the variant of interest, $\boldsymbol{y} = (y_1\ y_2\ ...\ y_N)^T$ is a length N column vector for the phenotype (Chen et al., 2016). The variance of the score is $Var(S|H_0) = \boldsymbol{g}^T\boldsymbol{P}\boldsymbol{g}$.

Denote the jth row of the stochastic summary statistic matrix $\boldsymbol{U}$ (for variant j, $1 \le j \le M$) by a length B row vector $\boldsymbol{U}_{j.}$, we can show that the variance $Var(S|H_0)$ of single-variant score S for variant j can be estimated as $\frac{1}{B}\boldsymbol{U}_{j.}\boldsymbol{U}_{j.}{}^T$, without using any individual-level data. The asymptotic P value is then computed using the single-variant score S[55] and its variance estimated from the stochastic summary statistic matrix $\boldsymbol{U}$, for each variant of interest.

2.4 Conditional association tests

Assume $\dot{G}$ is an $N \times c$ genotype matrix for $c \ge 1$ association genetic variants to be conditioned on and $\boldsymbol{g}$ is a length N column genotype vector for the variant of interest in the conditional association test. The single-variant score conditional on the variant set $\dot{\boldsymbol{G}}$ is $S_{\boldsymbol{g}|\dot{\boldsymbol{G}}} = S_{\boldsymbol{g}} - \boldsymbol{g}^T\boldsymbol{P}\dot{\boldsymbol{G}}\left(\dot{\boldsymbol{G}}^T\boldsymbol{P}\dot{\boldsymbol{G}}\right)^{-1}S_{\dot{\boldsymbol{G}}}$.

The variance of the conditional score is $Var\left(S_{\boldsymbol{g}|\dot{\boldsymbol{G}}}\right) = \boldsymbol{g}^T\boldsymbol{P}\boldsymbol{g} - \boldsymbol{g}^T\boldsymbol{P}\dot{\boldsymbol{G}}\left(\dot{\boldsymbol{G}}^T\boldsymbol{P}\dot{\boldsymbol{G}}\right)^{-1}\dot{\boldsymbol{G}}^T\boldsymbol{P}\boldsymbol{g}$[17].

In the StocSum framework, $S_{\boldsymbol{g}}$ and $\boldsymbol{U}_{\boldsymbol{g}}$ are the single-variant score and stochastic summary statistics corresponding to the variant of interest in the conditional association test and $\boldsymbol{S}_{\dot{\boldsymbol{G}}}$ (a length c vector) and $\boldsymbol{U}_{\dot{\boldsymbol{G}}}$ (a $c \times B$ matrix) are the single-variant score and stochastic summary statistics corresponding to the association variants to be conditioned on. The conditional score can be computed as

$S_{\boldsymbol{g}|\dot{\boldsymbol{G}}} = S_{\boldsymbol{g}} - \boldsymbol{U}_{\boldsymbol{g}}\boldsymbol{U}_{\dot{\boldsymbol{G}}}^T\left(\boldsymbol{U}_{\dot{\boldsymbol{G}}}\boldsymbol{U}_{\dot{\boldsymbol{G}}}^T\right)^{-1}\boldsymbol{S}_{\dot{\boldsymbol{G}}}$,

and the conditional stochastic summary statistics can be computed as

$\boldsymbol{U}_{\boldsymbol{g}|\dot{\boldsymbol{G}}} = \boldsymbol{U}_{\boldsymbol{g}} - \boldsymbol{U}_{\boldsymbol{g}}\boldsymbol{U}_{\dot{\boldsymbol{G}}}^T\left(\boldsymbol{U}_{\dot{\boldsymbol{G}}}\boldsymbol{U}_{\dot{\boldsymbol{G}}}^T\right)^{-1}\boldsymbol{U}_{\dot{\boldsymbol{G}}}$.

The variance $Var\left(S_{\boldsymbol{g}|\dot{\boldsymbol{G}}}\right)$ of the conditional score $S_{\boldsymbol{g}|\dot{\boldsymbol{G}}}$ can be estimated as $\frac{1}{B}\boldsymbol{U}_{\boldsymbol{g}|\dot{\boldsymbol{G}}}\boldsymbol{U}_{\boldsymbol{g}|\dot{\boldsymbol{G}}}{}^T$. The asymptotic P value is computed using the conditional score $S_{\boldsymbol{g}|\dot{\boldsymbol{G}}}$ and its variance estimated from the stochastic summary statistics $\boldsymbol{U}_{\boldsymbol{g}|\dot{\boldsymbol{G}}}$, for each variant of interest in the conditional association test.

2.5 Gene-environment interaction tests

We introduce a general model for testing m gene-environment interaction (GEI) terms in the GLMM framework. The full model including the genetic main effect and GEI effects is

$$\mathbb{g}(\mu_i) = \boldsymbol{X_i}\boldsymbol{\alpha} + g_i\beta + \boldsymbol{H_i}\boldsymbol{\gamma} + b_i, \quad (3)$$

where g_i is the genotype for the variant of interest for individual i, β is a scalar of the genetic main effect, $\boldsymbol{H}_i$ is a length m row vector for the GEI terms, which include the products of g_i and m environmental factors (a subset from p covariates in $\boldsymbol{X}_i$), and $\boldsymbol{\gamma}$ is a length m column vector for GEI effects. We note that under the constraint $\boldsymbol{\gamma} = 0$, β also represents the marginal genetic effect. Other notations follow the null model (Eq.(2)).

The single-variant score for the marginal genetic effect is $S_{\boldsymbol{g}} = \frac{\boldsymbol{g}^T(\boldsymbol{y}-\widehat{\boldsymbol{\mu}}_{\boldsymbol{0}})}{\widehat{\phi}}$ and its variance is $Var(S_{\boldsymbol{g}}) = \boldsymbol{g}^T\boldsymbol{P}\boldsymbol{g}$. The single-variant score for the GEI effects is $S_{\boldsymbol{H}} = \frac{\boldsymbol{H}^T(\boldsymbol{y}-\widehat{\boldsymbol{\mu}}_{\boldsymbol{0}})}{\widehat{\phi}}$ and its $m \times m$ covariance matrix is $Var(S_{\boldsymbol{H}}) = \boldsymbol{H}^T\boldsymbol{P}\boldsymbol{H}$, where $\boldsymbol{H} = (\boldsymbol{H}_1^T\ \boldsymbol{H}_2^T\ \cdots\ \boldsymbol{H}_N^T)^T$ is a $N \times m$ matrix for the GEI terms. The score for GEI effects adjusting for the marginal genetic effect can be approximated by $S_{\boldsymbol{H}|\boldsymbol{g}} = S_{\boldsymbol{H}} - \boldsymbol{H}^T\boldsymbol{P}\boldsymbol{g}(\boldsymbol{g}^T\boldsymbol{P}\boldsymbol{g})^{-1}S_{\boldsymbol{g}}$ [61], with a covariance matrix $Var(S_{\boldsymbol{H}|\boldsymbol{g}}) = \boldsymbol{H}^T\boldsymbol{P}\boldsymbol{H} - \boldsymbol{H}^T\boldsymbol{P}\boldsymbol{g}(\boldsymbol{g}^T\boldsymbol{P}\boldsymbol{g})^{-1}\boldsymbol{g}^T\boldsymbol{P}\boldsymbol{H}$. The marginal genetic effect can be tested using the quadratic form ${S_{\boldsymbol{g}}}^T Var(S_{\boldsymbol{g}})^{-1} S_{\boldsymbol{g}}$, which follows a chi-square distribution with 1 degree of freedom under the null hypothesis of no marginal genetic effects. The GEI effects can be tested using ${S_{\boldsymbol{H}|\boldsymbol{g}}}^T Var(S_{\boldsymbol{H}|\boldsymbol{g}})^{-1} S_{\boldsymbol{H}|\boldsymbol{g}}$, which follows a chi-square distribution with m degrees of freedom under the null hypothesis of no gene-environment interactions. The joint test, which evaluates both marginal genetic effects and GEI effects, can be constructed by the sum of these two chi-square statistics, since $S_{\boldsymbol{H}}$ and $S_{\boldsymbol{H}|\boldsymbol{g}}$ are asymptotically independent. The joint test statistic follows a chi-square distribution with $1 + m$ degrees of freedom under the null hypothesis of no marginal genetic effects or gene-environment interactions.

In the StocSum framework, we first compute stochastic summary statistics for the marginal genetic effect $\boldsymbol{U_g} = \boldsymbol{g}^T\boldsymbol{R}$ and GEI effects $\boldsymbol{U_H} = \boldsymbol{H}^T\boldsymbol{R}$ using individual-level data. We can use $\frac{1}{B}\boldsymbol{U_g}\boldsymbol{U_g}^T$, $\frac{1}{B}\boldsymbol{U_H}\boldsymbol{U_H}^T$, and $\frac{1}{B}\boldsymbol{U_g}\boldsymbol{U_H}^T$ to estimate the variance of the marginal genetic effect score $Var(S_{\boldsymbol{g}})$, the covariance matrix of the GEI effect score $Var(S_{\boldsymbol{H}})$, and the covariance of $S_{\boldsymbol{g}}$ and $S_{\boldsymbol{H}}$, respectively. The adjusted scores can be constructed as $S_{\boldsymbol{H}|\boldsymbol{g}} = S_{\boldsymbol{H}} - \boldsymbol{U_H}\boldsymbol{U_g}^T\left(\boldsymbol{U_g}\boldsymbol{U_g}^T\right)^{-1}S_{\boldsymbol{g}}$, and its variance $Var(S_{\boldsymbol{H}|\boldsymbol{g}})$ can be approximated as $\frac{1}{B}\left\{\boldsymbol{U_H}\boldsymbol{U_H}^T - \boldsymbol{U_H}\boldsymbol{U_g}^T\left(\boldsymbol{U_g}\boldsymbol{U_g}^T\right)^{-1}\boldsymbol{U_g}\boldsymbol{U_H}^T\right\}$.

2.6 Variant set tests

We include four variant set tests: the burden test[34–37], SKAT[38], SKAT-O[83], and the efficient hybrid test of burden and SKAT[21,39], in the StocSum framework. Here we consider a variant set including q genetic variants ($q > 1$) and denote $\tilde{\boldsymbol{S}}$ as a length q single-variant score vector, and $\tilde{\boldsymbol{G}}$ as an $N \times q$ genotype matrix (a subset of the $N \times M$ genotype matrix $\boldsymbol{G}$ on the whole genome, or on one chromosome). We note that our examples are not a complete list of all variant set tests that are commonly used, but any other variant set tests that would require $q \times q$ covariance matrices could also be implemented using stochastic summary statistics.

The burden test statistic can be constructed as

$T_{Burden} = \tilde{\boldsymbol{S}}^T \boldsymbol{W} \mathbf{1}_q \mathbf{1}_q^T \boldsymbol{W} \tilde{\boldsymbol{S}}$,

where $\boldsymbol{W} = diag\{w_j\}$ is a pre-specified $q \times q$ diagonal weight matrix, and $\mathbf{1}_q$ is a length q vector of 1's. The weights can be a function of the MAF[36,38], or functional annotation scores such as CADD[84,85], FATHMM-XF[86], and annotation principal components from STAAR[87]. Under the null hypothesis, the statistic T_{Burden} asymptotically follows $\xi_{Burden}\chi_1^2$, where the scaling factor $\xi_{Burden} = \mathbf{1}_q^T \boldsymbol{W} \tilde{\boldsymbol{G}}^T \boldsymbol{P} \tilde{\boldsymbol{G}} \boldsymbol{W} \mathbf{1}_q = \mathbf{1}_q^T \boldsymbol{W} \tilde{\boldsymbol{V}} \boldsymbol{W} \mathbf{1}_q$ (where $\tilde{\boldsymbol{V}}$ is a $q \times q$ covariance matrix for the single-variant score vector $\tilde{\boldsymbol{S}}$), and χ_1^2 is a chi-square distribution with 1 df. In the StocSum framework, ξ_{Burden} can be estimated as $\frac{1}{B}\mathbf{1}_q^T \boldsymbol{W} \tilde{\boldsymbol{U}} \tilde{\boldsymbol{U}}^T \boldsymbol{W} \mathbf{1}_q = \frac{1}{B}\tilde{\boldsymbol{u}}^T \tilde{\boldsymbol{u}}$, where $\tilde{\boldsymbol{U}}$ is a $q \times B$ matrix (a subset of the $M \times B$ stochastic summary statistic matrix $\boldsymbol{U}$), and $\tilde{\boldsymbol{u}} = \tilde{\boldsymbol{U}}^T \boldsymbol{W} \mathbf{1}_q$ is a length B vector (i.e., column sum of $\boldsymbol{W}\tilde{\boldsymbol{U}}$).

The SKAT statistic can be constructed as

$$T_{SKAT} = \tilde{\boldsymbol{S}}^T \boldsymbol{W} \boldsymbol{W} \tilde{\boldsymbol{S}}.$$

Under the null hypothesis, T_{SKAT} asymptotically follows $\sum_{j=1}^{q} \xi_{SKAT_j} \chi_{1,j}^2$, where $\chi_{1,j}^2$ are independent chi-square distributions with 1 df, and ξ_{SKAT_j} are the eigenvalues of $\boldsymbol{\Xi}_{SKAT} = \boldsymbol{W} \tilde{\boldsymbol{G}}^T \boldsymbol{P} \tilde{\boldsymbol{G}} \boldsymbol{W} = \boldsymbol{W} \tilde{\boldsymbol{V}} \boldsymbol{W}$. In the StocSum framework, ξ_{SKAT_j} can be estimated as the square of the singular values of $\frac{1}{\sqrt{B}} \boldsymbol{W} \tilde{\boldsymbol{U}}$ (**Supplementary Note 1**).

In SKAT-O, the variance component statistic T_ρ given a weight parameter ρ ($0 \le \rho \le 1$) is

$T_\rho = \rho T_{Burden} + (1 - \rho) T_{SKAT}$.

If $\rho = 1$, T_ρ becomes the burden test statistic T_{Burden}; if $\rho = 0$, T_ρ becomes the SKAT statistic T_{SKAT}. SKAT-O searches for an optimal ρ by minimizing the P value of T_ρ. Specifically, the $q \times q$ weighted covariance matrix $\boldsymbol{\Xi}_{SKAT} = \boldsymbol{W} \tilde{\boldsymbol{V}} \boldsymbol{W}$ is decomposed into two parts $\boldsymbol{\Xi}_{Burden} = \boldsymbol{\Xi}_{SKAT} \mathbf{1}_q (\mathbf{1}_q^T \boldsymbol{\Xi}_{SKAT} \mathbf{1}_q)^{-1} \mathbf{1}_q^T \boldsymbol{\Xi}_{SKAT}$ and $\boldsymbol{\Xi}_{SKAT|Burden} = \boldsymbol{\Xi}_{SKAT} -$

$\boldsymbol{\Xi}_{Burden}$, used in subsequent one-dimensional numerical integration to compute the SKAT-O *P* value. In the StocSum framework, $\boldsymbol{\Xi}_{Burden}$ can be estimated as $\frac{1}{B}\widetilde{\boldsymbol{U}}_{Burden}\widetilde{\boldsymbol{U}}_{Burden}^{T}$, where $\widetilde{\boldsymbol{U}}_{Burden} = \boldsymbol{W}\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{u}}(\widetilde{\boldsymbol{u}}^T\widetilde{\boldsymbol{u}})^{-1}\widetilde{\boldsymbol{u}}^T$, and $\boldsymbol{\Xi}_{SKAT|Burden}$ can be estimated as $\frac{1}{B}\widetilde{\boldsymbol{U}}_{SKAT|Burden}\widetilde{\boldsymbol{U}}_{SKAT|Burden}^{T}$, where $\widetilde{\boldsymbol{U}}_{SKAT|Burden} = \boldsymbol{W}\widetilde{\boldsymbol{U}} - \widetilde{\boldsymbol{U}}_{Burden}$.

In the efficient hybrid test to combine the burden test and SKAT, the adjusted SKAT statistic $T_{SKAT|Burden}$ can be approximated by

$$T_{SKAT|Burden} = \tilde{\boldsymbol{S}}^T\boldsymbol{W}\left\{\boldsymbol{I}_q - \mathbf{1}_q\left(\mathbf{1}_q^T\boldsymbol{\Xi}_{SKAT}\mathbf{1}_q\right)^{-1}\mathbf{1}_q^T\boldsymbol{\Xi}_{SKAT}\right\}\left\{\boldsymbol{I}_q - \boldsymbol{\Xi}_{SKAT}\mathbf{1}_q\left(\mathbf{1}_q^T\boldsymbol{\Xi}_{SKAT}\mathbf{1}_q\right)^{-1}\mathbf{1}_q^T\right\}\boldsymbol{W}\tilde{\boldsymbol{S}}.$$

Under the null hypothesis, $T_{SKAT|Burden}$ asymptotically follows $\sum_{j=1}^{q}\xi_{SKAT|Burden_j}\chi_{1,j}^2$, where $\chi_{1,j}^2$ are independent chi-square distributions with 1 df and $\xi_{SKAT|Burden_j}$ are the eigenvalues of $\boldsymbol{\Xi}_{SKAT|Burden}$. In the StocSum framework, these eigenvalues can be estimated as the square of the singular values of $\frac{1}{\sqrt{B}}\widetilde{\boldsymbol{U}}_{SKAT|Burden}$ (**Supplementary Note 2**).

2.7 Meta-analysis

In a traditional meta-analysis on a region with q genetic variants from L studies, we use the single-variant scores $\tilde{\boldsymbol{S}}_l$ and the covariance matrix $\widetilde{\boldsymbol{V}}_l$ from each study l ($1 \le l \le L$). The variant set meta-analysis can be performed using the summary scores $\tilde{\boldsymbol{S}} = \sum_{l=1}^{L}\tilde{\boldsymbol{S}}_l$ and the summary covariance matrix $\widetilde{\boldsymbol{V}} = \sum_{l=1}^{L}\widetilde{\boldsymbol{V}}_l$[18,19,21,31,33]. The single-variant meta-analysis only requires $\tilde{\boldsymbol{S}}$ and the diagonal elements of $\widetilde{\boldsymbol{V}}$. In the StocSum framework, we compute $\widetilde{\boldsymbol{U}} = \sum_{l=1}^{L}\widetilde{\boldsymbol{U}}_l$ instead of $\widetilde{\boldsymbol{V}}$. Assuming $q < B$, each column of $\widetilde{\boldsymbol{U}}_l$ follows a multivariate normal distribution with mean $\mathbf{0}$ and covariance matrix $\widetilde{\boldsymbol{V}}_l$, and $\widetilde{\boldsymbol{U}}_l$ are independent across L studies assuming no sample overlaps or between-study relatedness. Therefore, each column of $\widetilde{\boldsymbol{U}}$ follows a multivariate normal distribution with mean $\mathbf{0}$ and covariance matrix $\widetilde{\boldsymbol{V}}$. In our implementation, we first compute the stochastic summary statistic matrix $\boldsymbol{U} = \sum_{l=1}^{L}\boldsymbol{U}_l$ for all M genetic variants on the whole genome (or one chromosome), regardless of how variants should be grouped, and then extract q genetic variants by taking a subset of $\boldsymbol{U}$ only when computing *P* values, for both single-variant meta-analysis and variant set meta-analysis.

2.8 LD score regression

LD Score Regression (LDSC) has been widely applied to GWAS summary statistics to estimate confounding bias, heritability explained by genotyped variants, heritability enrichments of functional categories, and genetic correlations[14,15,88]. The classical LDSC model can be written as

$$E\left[{\chi^2}_j | l_j\right] = \frac{Nh^2 l_j}{M} + Na + 1,$$

where ${\chi^2}_j$ denotes the χ^2 statistic of variant j from GWAS summary statistics; $l_j = \sum_k r_{jk}^2$ is the LD score of variant j with r_{jk}^2 being the squared Pearson correlation coefficient of genotypes between variants j and k, N is the sample size, M is the total number of variants, a is a measure of confounding bias, and h^2 is the heritability of the phenotype. In practice, LDSC calculates l_j by summing up $\hat{r}_{adj_{jk}}^2$ for all variants k in specific window around the index variant j. The adjusted correlation estimate $\hat{r}_{adj_{jk}}$ can be computed from the sample correlation estimate $\hat{r}_{jk}$ using

$\hat{r}_{adj_{jk}}^2 = {\hat{r}_{jk}}^2 - \frac{1-{\hat{r}_{jk}}^2}{N-2}$.

Sample correlation coefficients $\hat{r}_{jk}$ can be estimated as $\frac{w_j \boldsymbol{G}_{.j}{}^T \boldsymbol{L} \boldsymbol{G}_{.k} w_k}{N-1}$, where $\boldsymbol{G}_{.j}$ is the jth column of the genotype matrix $\boldsymbol{G}$, representing variant j, $\boldsymbol{L} = \left(\boldsymbol{I}_N - \mathbf{1}_N\left(\mathbf{1}_N{}^T \mathbf{1}_N\right)^{-1}\mathbf{1}_N{}^T\right)$ is an $N \times N$ idempotent projection matrix, and $w_j = \frac{1}{\sqrt{2f_j(1-f_j)}}$ (f_j is the MAF of variant j) is a weight that standardizes $\boldsymbol{G}_{.j}$ to a unit variance.

In the StocSum framework, we construct the $N \times B$ random matrix as $\boldsymbol{R} = \boldsymbol{L}\boldsymbol{r}_0$, where $\boldsymbol{r}_0$ is an $N \times B$ random matrix with all entries simulated from a standard normal distribution. For an $N \times M$ genotype matrix $\boldsymbol{G}$ for all M genetic variants on the whole genome (or one chromosome), we compute the $M \times B$ stochastic summary statistic matrix $\boldsymbol{U} = \boldsymbol{W}\boldsymbol{G}^T\boldsymbol{R}$, where $\boldsymbol{W} = diag\{w_j\}$ is an $M \times M$ diagonal weight matrix. For variant j, we subset M_j variants within the flanking region (with a default window width of 1000 Kb) to get the corresponding $M_j \times B$ subset $\widetilde{\boldsymbol{U}}$. The adjusted correlation coefficient $\tilde{r}_{adj_{jk}}$ for $\tilde{r}_{jk}$ from StocSum is computed as (**Supplementary Note 3**)

$$\tilde{r}_{adj_{jk}}^2 = {\tilde{r}_{jk}}^2 - \frac{1-{\tilde{r}_{jk}}^2}{B-2} - \frac{1-{\tilde{r}_{jk}}^2}{N-2}.$$

The LD score l_j of variant j could be estimated by summarizing stochastic summary statistics of M_j variants in flanking region,

$$l_j = \sum_{k=1}^{M_j} \tilde{r}_{adj_{jk}}^2 = \left\{\sum_{k=1}^{M_j}\left(1 + \frac{1}{B-2} + \frac{1}{N-2}\right){\tilde{r}_{jk}}^2\right\} - \frac{M_j}{B-2} - \frac{M_j}{N-2}$$

$$= \left(1 + \frac{1}{B-2} + \frac{1}{N-2}\right)\left(\frac{\widetilde{\boldsymbol{U}}\,\widetilde{\boldsymbol{U}}_{j.}{}^T}{B(N-1)} \circ \frac{\widetilde{\boldsymbol{U}}\,\widetilde{\boldsymbol{U}}_{j.}{}^T}{B(N-1)}\right)^T 1_{M_j} - \frac{M_j}{B-2} - \frac{M_j}{N-2}.$$

in which $\circ$ denotes the Hadamard product, and $\widetilde{\boldsymbol{U}}_{j.}$ is the jth row of $\widetilde{\boldsymbol{U}}$.

3 Getting started

3.1 Downloading StocSum

StocSum is an open source project and is freely available for download at https://github.com/NWang-hub/StocSum.

3.2 Installing StocSum

The following R packages are required before installing StocSum: Rcpp and RcppArmadillo for R and C++ integration and testthat to run code checks during development. Additionally, StocSum imports from Rcpp, CompQuadForm, Foreach, parallel, Matrix, methods, GMMAT, and Bioconductor packages SeqArray and SeqVarTools. The R package doMC is required to run parallel computing in StocSum.stat, StocSum_GE.stat, StocSum_LDSC.stat (doMC is not available on Windows and these functions will switch to a single computer thread).

For optimal computational performance, it is recommended to use an R version configured with the Intel Math Kernel Library (or other fast BLAS/LAPACK libraries). See the instructions on building R with Intel MKL (https://software.intel.com/en-us/articles/using-intel-mkl-with-r).

Here is an example for installing StocSum and all its dependencies in an R session (assuming none of the R packages other than the default has been installed):

> install.packages(c("devtools", "RcppArmadillo", "CompQuadForm", "doMC", "foreach", "Matrix", "data.table", "GMMAT", "BiocManager", "testthat"), repos="https://cran.r-project.org/")
> BiocManager::install(c("SeqArray", "SeqVarTools"))
> devtools::install_github("https://github.com/NWang-hub/StocSum")

4 Input

StocSum requires an object from fitting the null model using the glmm.kin function from the GMMAT package, and a genotype file in a GDS format. For variant set test, a user-defined marker group file is required. Specified formats of these files are described as follows

4.1 Object

StocSum can perform various applications including single-variant tests, conditional association tests, gene-environment interaction tests, variant set tests, as well as meta-analysis and LD score regression tools. To fit the null model, the phenotype and covariates (include the environmental factors of interest) should be saved in a data frame. If the

samples are related, the relatedness should be known positive semidefinite matrices V_k as an R matrix (in the case of a single matrix) or an R list (in the case of multiple matrices). Refer to the GMMAT user manual (https://cran.r-project.org/web/packages/GMMAT/vignettes/GMMAT.pdf) to learn the method of fitting the null model. The class of the object should be "glmmkin".

4.2 Genotypes

StocSum can take genotype files in the GDS format. Genotypes in Variant Call Format (VCF) and PLINK binary PED format can be converted to the GDS format using seqVCF2GDS and seqBED2GDS functions from the SeqArray package:

> SeqArray::seqVCF2GDS("VCF_file_name","GDS_file_name")
>SeqArray::seqBED2GDS("BED_file_name","FAM_file_name","BIM_file_name","GDS_file_name")

4.3 Group definition file

For variant set test, a group definition file with no header and 6 columns (variant set id, variant chromosome, variant position, variant reference allele, variant alternate allele, weight) is required. For example, here we show the first 6 rows of the example group definition file "SetID.withweights.txt":

Set1	1	1	T	A	1
Set1	1	2	A	C	4
Set1	1	3	C	A	3
Set1	1	4	G	A	6
Set1	1	5	A	G	9
Set1	1	6	C	A1	9

Note that each variant in the group definition file is matched by chromosome, position, reference allele and alternate allele with variants from the GDS file. One genetic variant can be included in different groups with possibly different weights. If no external weights are needed in the analysis, simply replace the 6^{th} column by all 1's.

5 Running StocSum

If StocSum has been successfully installed, you can load it in an R session using

> library (StocSum)

We provide three functions in StocSum: StocSum.R to perform single-variant tests, conditional association tests, variant set tests, as well as meta-analysis; StocSum.GE.R to

perform gene-environment interaction tests; StocSum.LDSC.R to perform LD score regression.

In StocSum.R, the nested function glmmin2randomvec is for generating the random vectors, the nested function G.stat is for calculating the stochastic summary statistics, the nested function svt.pval is for running single variant score tests, the nested function svt.meta is perform meta-analysis on score test results, the next functions G.prep and G.pval is two-step low-memory of performing variant set tests.

In StocSum.GE.R, the nested function glmmin2randomvec is for generating the random vectors from multivariate normal distribution with mean 0 and covariance matrix P; the nested function GE.stat is for calculating the stochastic summary statistics; the nested function GE.svt.pval is for running single variant score tests,

In StocSum.LDSC.R, the nested function LDSC.glmmin2randomvec is for generating the random vectors from multivariate normal distrubution with mean 0 and covariance matrix $P = I - 1(1'1)^{-1}1'$; the nested function G.stat is for calculating the stochastic summary statistics; the nested function LDSC.win is for calculating the LD scores.

Details about how to use these functions, their arguments and returned values can be found in the R help document of StocSum. For example, to learn more about glmmin2randomvec in an R session you can type

> ?glmmin2randomvec

5.1 Fitting GLMM

StocSum requires a "glmmkin" class object that contains a fitted GLMM null model. The object can be obtained from the glmmkin function from the R package GMMAT. For more examples and details about the glmmkin function, see the GMMAT manual (https://cran.r-project.org/web/packages/GMMAT/vignettes/GMMAT.pdf). Below is an example of fitting a GLMM using the glmmkin function from GMMAT:

> library(StocSum)
> data(example)
> attach(example)
> GRM.file <- system.file("extdata", "GRM.txt.bz2", package = "StocSum")
> GRM <- as.matrix(read.table(GRM.file, check.names = FALSE))
> nullmod <- glmmkin(disease ~ age + sex, data = pheno, kins = GRM, id = "id", family = binomial(link = "logit"))

5.2 Single-variant tests

5.2.1 Generate random vectors

To run StocSum.R, the user needs to generate the random vectors that having covariance approximate to the projection matrix from above fitted null model.

```
> obj <- glmmkin2randomvec(nullmod)
```

The function glmmkin2randomvec returns a list. The element *random.vectors* stores the generated random vectors. The remaining elements *theta*, *scaled.residuals*, and *id_include* are inherited from the null model generated in 5.1.

If the returned nullmod object in 5.1 does no include the projection matrix **P**, the user needs to set the relation matrix in glmkin2randomvec function. The following is an example showing the setting when only the kinship is considered as the relation matrix.

```
> kinship.chol <- chol(GRM)
> obj <- glmmkin2randomvec(nullmod, Z = list(t(kinship.chol)))
```

5.2.2 Calculate summary statistics

The next step is to calculate the summary statistics and stochastic summary statistics. The genotype file in the GDS format is as input. The output of G.stat is intermediates files containing single variant scores and the stochastic summary statistics. An example is as following:

```
> out.prefix <- "test"
> gdsfile <- system.file("extdata", "geno.gds", package = "StocSum")
> G.stat(obj, geno.file = gdsfile, meta.file.prefix = out.prefix, MAF.range=c(0,0.5), miss.cutoff = 1)
```

The first argument “obj” in G.stat is the object returned by glmmkin2randomvec in section 5.2.2. The argument “geno.file” is the genotype file. The argument “meta.file.prefix” specifies the prefix of output files. In the example above, a space-delimited file “test.sample.1” will be generated to save the single variant scores, and a binary file “test.ressample.1” will be generated to save the stochastic summary statistics. Note that this binary file is not human-readable, but can be loaded by downstream modules/functions. The argument “ncores” in G.stat is to specify how many cores you would like to use on a computing node. It also determines the intermediate files. For example, if “ncores=2”, there will be four intermediate files, i.e., “test.sample.1”, “test.sample.2”, “test.resample.1”, “test.resample.2”.

5.2.3 Calculating P-values

When the intermediate files are generated by “G.stat”, the function “svt.pval” can be used to calculate P-values for each variants. An example is as following:

```
> out1<-svt.pval(out.prefix, n.file=ncores, MAF.range=c(0,0.5), miss.cutoff = 1, auto.flip=F)
```

The first argument “out.prefix” is the prefix of output files specified in “G.stat”.

5.2.4 Output Files

	SNP	chr	pos	ref	alt	N	missrate	altfreq	SCORE	VAR	PVAL
1	SNP1	1	1	T	A	393	0.0175	0.9745547	-1.9849977	4.588055	0.3540751
2	SNP2	1	2	A	C	400	0.0000	0.5000000	3.5103164	49.685470	0.6184822
3	SNP3	1	3	C	A	400	0.0000	0.7925000	0.5334004	33.398425	0.9264616
4	SNP4	1	4	G	A	400	0.0000	0.7012500	3.1149410	41.128852	0.6271732
5	SNP5	1	5	A	G	400	0.0000	0.5937500	-4.0013505	43.791006	0.5454023
6	SNP6	1	6	C	A	400	0.0000	0.8887500	-1.6920412	17.296781	0.6841223

The 11 columns are: SNP (“annotation/id”), chr (“chromosome”), pos (“position”), reference and alternate alleles, the sample size N, the genotype missing rate missrate, the allele frequency of ALT allele, the score statistic SCORE of ALT allele, the variance of the score VAR, the score test P value PVAL.

5.3 Variant set tests

5.3.1 Calculating P-values

Same as single-variant tests, the variant set tests also need generating random vectors (section 5.2.1) and calculating summary statistics (section 5.2.2). After these two steps, G.prep and G.pval is used to calculate P-vaules for variant sets. A group definition file with no header and 6 columns (variant set id, variant chromosome, variant position, variant reference allele, variant alternative allele, weight) is required, as described in section 4.3. Here we perform variant set test in single study with an example shown as following:

```
> group.file <- system.file("extdata", "SetID.withweights.txt", package = "GMMAT")
> obj.prep <- G.prep(out.prefix, n.files = 1, group.file = group.file, auto.flip=F)
> save(obj.prep,file=paste0(out.prefix,".prepobj.Rdata"))
> obj.prep <- get(load(paste0(out.prefix,".prepobj.Rdata")))
> out <- G.pval(obj.prep, MAF.range = c(0,0.5), miss.cutoff = 1, method = "davies")
> print(out)
```

5.3.2 Output Files

	group	n.variants	B.score	B.var	B.pval	E.pval

```
1  Set1       20  194.05011  84243.93 0.50377208 0.2001607
2  Set2       20  -82.55532 255018.97 0.87014224 0.9580280
3  Set3       20  184.18465 229741.44 0.70078013 0.4804642
4  Set4       20  296.38607  25970.19 0.06589123 0.1020998
5  Set5       20  446.62340  77028.37 0.10756768 0.3252381
6  Set6       20  260.94738 127859.95 0.46553112 0.5403710
7  Set7       20  186.76450 142536.96 0.62082101 0.5535321
8  Set8       20 -217.12052 119423.47 0.52981787 0.0473496
9  Set9       20   32.51345 185369.47 0.93980348 0.5839853
```

It returns a data frame with the first 2 columns showing the group (variant set) name, number of variants in each group. For Burden, 3 columns will be included to show the burden test score, variance of the score, and its P value. For the efficient hybrid test, the P value column will be included in the last column.

5.4 Meta-analysis

5.4.1 single-variant meta-analysis

Score test results from multiple studies can be combined in meta-analysis. The intermediate files for each study from G.stat can be used as input to the function svt.meta.

```
> library(StocSum)
> data(example)
> attach(example)
> seed <- 12345
> set.seed(seed)
> GRM.file <- system.file("extdata", "GRM.txt.bz2", package = "StocSum")
> GRM <- as.matrix(read.table(GRM.file, check.names = FALSE))
> nullmod <- glmmkin(disease ~ age + sex, data = pheno, kins = GRM, id = "id", family =
binomial(link = "logit"))
> if(!is.null(nullmod$P)){
+obj <- glmmkin2randomvec(nullmod)
+}else{
+kinship.chol <- chol(GRM)
+obj<-glmmkin2randomvec(nullmod, Z = list(t(kinship.chol)))
+}
> out.prefix <- "test"
> gdsfile <- system.file("extdata", "geno.gds", package = "StocSum")
> G.stat(obj, geno.file = gdsfile, meta.file.prefix = out.prefix, MAF.range=c(0,0.5),
miss.cutoff = 1)
> GRM1.file <- system.file("extdata", "GRM1.txt.bz2", package = "StocSum")
> GRM1 <- as.matrix(read.table(GRM1.file, check.names = FALSE))
> nullmod1 <- glmmkin(disease ~ age + sex, data = pheno1, kins = GRM1, id = "id", family
= binomial(link = "logit"))
> if(!is.null(nullmod$P)){
+obj1 <- glmmkin2randomvec(nullmod1)
```

```
+}else{
+kinship1.chol <- chol(GRM1)
+obj1 <- glmmkin2randomvec(nullmod1, Z = list(t(kinship1.chol)))
+}
> out.prefix1 <- "test1"
> gdsfile1 <- system.file("extdata", "geno1.gds", package = "GMMAT")
> G.stat(obj1, geno.file = gdsfile1, meta.file.prefix = out.prefix1, MAF.range=c(0,0.5), miss.cutoff = 1)
> outMeta.prefix <- "comp.meta"
> svt.meta(c("test", "test1"), n.files = rep(1, 2), outfile.prefix=outMeta.prefix, MAF.range=c(0,0.5), miss.cutoff = 1)
```

5.4.2 variant set meta-analysis

For variant set meta-analysis, the intermediate files for each study from G.stat can be used as input to the function G.pval. For example,

```
> library(StocSum)
> data(example)
> attach(example)
> seed <- 12345
> set.seed(seed)
> GRM.file <- system.file("extdata", "GRM.txt.bz2", package = "StocSum")
> GRM <- as.matrix(read.table(GRM.file, check.names = FALSE))
> nullmod <- glmmkin(disease ~ age + sex, data = pheno, kins = GRM, id = "id", family = binomial(link = "logit"))
> if(!is.null(nullmod$P)){
+obj <- glmmkin2randomvec(nullmod)
+}else{
+kinship.chol <- chol(GRM)
+obj<-glmmkin2randomvec(nullmod, Z = list(t(kinship.chol)))
+}
> out.prefix <- "test"
> gdsfile <- system.file("extdata", "geno.gds", package = "StocSum")
> G.stat(obj, geno.file = gdsfile, meta.file.prefix = out.prefix, MAF.range=c(0,0.5), miss.cutoff = 1)
> GRM1.file <- system.file("extdata", "GRM1.txt.bz2", package = "StocSum")
> GRM1 <- as.matrix(read.table(GRM1.file, check.names = FALSE))
> nullmod1 <- glmmkin(disease ~ age + sex, data = pheno1, kins = GRM1, id = "id", family = binomial(link = "logit"))
> if(!is.null(nullmod$P)){
+obj1 <- glmmkin2randomvec(nullmod1)
+}else{
+kinship1.chol <- chol(GRM1)
+obj1 <- glmmkin2randomvec(nullmod1, Z = list(t(kinship1.chol)))
+}
```

```
> out.prefix1 <- "test1"
> gdsfile1 <- system.file("extdata", "geno1.gds", package = "GMMAT")
> G.stat(obj1, geno.file = gdsfile1, meta.file.prefix = out.prefix1, MAF.range=c(0,0.5), miss.cutoff = 1)
> group.file <- system.file("extdata", "SetID.withweights.txt", package = "GMMAT")
> obj.prep <- G.prep(meta.files.prefix= c("test", "test1"), n.files = rep(1, 2), group.file = group.file, auto.flip=F)
> outMeta.prefix <- “test.Meta”
> save(obj.prep,file=paste0(outMeta.prefix,".prepobj.Rdata"))
> obj.prep <- get(load(paste0(outMeta.prefix,".prepobj.Rdata")))
> out <- G.pval(obj.prep, MAF.range = c(0,0.5), miss.cutoff = 1, method = "davies")
> print(out)
```

5.5 Conditional association tests

Same as single-variant tests, the variant set tests also need generating random vectors (section 5.2.1) and calculating summary statistics (section 5.2.2). After these two steps, Cond.svt.pval is used to calculate P-vaules.

```
library(StocSum)
> data(example)
> attach(example)
> seed <- 12345
> set.seed(seed)
> GRM.file <- system.file("extdata", "GRM.txt.bz2", package = "StocSum")
> GRM <- as.matrix(read.table(GRM.file, check.names = FALSE))
> nullmod <- glmmkin(disease ~ age + sex, data = pheno, kins = GRM, id = "id", family = binomial(link = "logit"))
> if(!is.null(nullmod$P)){
+   obj <- glmmkin2randomvec(nullmod)
+ }else{
+   kinship.chol <- chol(GRM)
+   obj<-glmmkin2randomvec(nullmod, Z = list(t(kinship.chol)))
+ }
> out.prefix <- "test"
> gdsfile <- system.file("extdata", "geno.gds", package = "GMMAT")
> G.stat(obj, geno.file = gdsfile, meta.file.prefix = out.prefix, MAF.range=c(0,0.5), miss.cutoff = 1)
> out <- Cond.svt.pval(out.prefix, n.files = 1, tagChr = 1, StartPos = 1, EndPos = 100, tagPos = 82, MAF.range=c(0,0.5), miss.cutoff = 1, auto.flip=F)
> print(out)
```

5.6 Gene-environment tests

Same as single-variant tests, the variant set tests also need generating random vectors (section 5.2.1).

5.6.1 Calculate summary statistics

Different from G.state, the "GE.stat" needs the arguments "interaction" and "intearaction.covariates" as input. An example is as following:

```
> out.prefix <- "test.GE"
> gdsfile <- system.file("extdata", "geno.gds", package = "StocSum")
> GE.stat(obj, interaction='sex', geno.file = gdsfile, meta.file.prefix = out.prefix)
```

5.6.2 Calculating P-values

When the intermediate files are generated by "GE.stat", the function "GE.svt.pval" can be used to calculate P-values for each variants. An example is as following:

```
> out.file<-paste0(out.prefix,".out")
> GE.svt.pval(meta.files.prefix = out.prefix, out.file, n.files = 1, n.pheno = 1, interaction=interaction, MAF.range=c(1e-7,0.5), miss.cutoff = 1, auto.flip=F)
```

5.7 LD score regression

5.7.1 Generate random vectors

To run StocSum.LDSC.R, the user needs to generate the random vectors from multivariate normal distrubution with mean 0 and covariance matrix $P = I - 1(1'1)^{-1}1'$.

```
> obj <- LDSC.glmmkin2randomvec(nullmod)
```

The function LDSC.glmmkin2randomvec returns a list. The element *random.vectors* stores the generated random vectors. The remaining elements *theta*, *scaled.residuals*, and *id_include* are inherited from the null model generated in 5.1.

5.7.2 Calculate summary statistics

Same function G.stat as in secton 5.2.2.

5.7.3 Calculate LD scores

The function LDSC.win is used to calculate LD scores. The argument "wind.b" defines the window size in kilobases to estimate LD Scores.

```
> library(StocSum)
> data(example)
```

```
> attach(example)
> seed <- 12345
> set.seed(seed)
> GRM.file <- system.file("extdata", "GRM.txt.bz2", package = "GMMAT")
> GRM <- as.matrix(read.table(GRM.file, check.names = FALSE))
> nullmod <- glmmkin(disease ~ age + sex, data = pheno, kins = GRM, id = "id", family
= binomial(link = "logit"))
> obj <- LDSC.glmmkin2randomvec(nullmod)
> out.prefix <- "test"
> gdsfile <- system.file("extdata", "geno.gds", package = "GMMAT")
> G.stat(obj, geno.file = gdsfile, meta.file.prefix = out.prefix, MAF.range=c(0,0.5),
miss.cutoff = 1)
> out<-LDSC.win(out.prefix, use.minor.allele = FALSE, auto.flip = FALSE, wind.b =
1000000, nperbatch = 10000))
```

5.7.4 Output

The 6 columns are: chr (“chromosome”), pos (“position”), the sample size N, the genotype missing rate missrate, the allele frequency of ALT allele, LD Scores.

```
   chr pos   N missrate   altfreq   LDscore
1    1   1 393   0.0175 0.9745547 1.7255943
2    1   2 400   0.0000 0.5000000 1.2154295
3    1   3 400   0.0000 0.7925000 1.3473072
4    1   4 400   0.0000 0.7012500 1.8789483
5    1   5 400   0.0000 0.5937500 1.0940140
6    1   6 400   0.0000 0.8887500 1.0354738
7    1   7 400   0.0000 0.9687500 0.9296832
8    1   8 400   0.0000 0.8837500 0.8144483
9    1   9 400   0.0000 0.6475000 4.1155792
10   1  10 400   0.0000 0.7675000 1.2189952
```

6 Advanced options

7 Version

8 Contact

9 Acknowledgments

References

Breslow, Norman E., and David G. Clayton. "Approximate inference in generalized linear mixed models." Journal of the American statistical Association 88.421 (1993): 9-25.