Last update: 6/1/2014

Estimating Variance Components in MMAP

MMAP implements routines to estimate variance components within the mixed model. These estimates can be used for likelihood ratio tests to compare model fits or as null model estimates for score tests. Estimates of individual random effects are provided for analysis or as residuals for additional analyses such as score tests. Any number of random effects can be modeled. These can include any combination of pedigree, genomic, omic and environment effects. Non-heteroscedastic errors can be modeled to allow group-specific variances such as by-gender (gQTLs) or by-genotype (vQTLS).

See the documentation on covariance matrices and genomic matrices for constructing matrices to model the random effects and the score test for rare variant testing.

Command line options:

The pedigree, phenotype and trait options are as described previously.

--ped <pedigree filename>

--phenotype_filename <filename>

--trait <trait>

--covariates <covariates>

Variance Component Likelihood Estimation

--estimate variance components

Specifies that a variance component model is to be run.

--binary_covariance_filename <filename 1> <filename 2> …. < filename N>

<filename J> corresponds to the Jth variance component to be estimated in the model. The residual error is assumed homoscedastic and is the N+1st variance term (see below for more general residual error models). The format of the variance component file is specific to MMAP. MMAP has options to create a variety standard covariance matrices based on pedigree that include additive, dominance, epistasis, maternal and paternal imprinting, X chromosome. Covariance matrices using genetic data include additive, dominance and epistasis (instructions on how to create them are in the **Genomic Matrices** document.

--variance_component_label <label 1> <label 2> … <label N>

Label is used in the output file to distinguish the variance component names. If no label is supplied the default label VAR is used. For example, the label for the kinship matrix might be A, for epistasis AA and dominance D and genomic matrix G.

The default model assume heteroscedastic error (identity matrix), but more general residuals errors can be modeled, for example, multiple measurements, sexual dimorphism, variance-heterogeneity QTLs (vQTLs). Currently these error matrices are assumed diagonal.

--error_variance_componen_filename <filename 1> <filename 2> …. < filename N>

<filename J> corresponds to the Jth error variance component to be estimated in the model. These matrices are diagonal, so represented by a csv file with header with first column the subject id and second the diagonal value. For example, to estimate male and female variances separately there would be one file with for males containing 0 for each female and 1 for each male. The female file would be dual to this. Alternately, one could model the error using a file with all 1's, then add either the female or male file. This model can then be compared to the homoscedastic model as a nested model. For vQTLs the matrices would have 1's for the genotype class, 0's otherwise.

--error_variance_component_label <label 1> <label 2> … <label N>

Label is used in the output file to distinguish the error variance component names.

Likelihood Options

MMAP implements 4 different options for REML likelihood estimation: EM-REML, AI-REML, Fisher information and Newton-Raphson. Both Fisher information and Netwon-Raphson involve a trace of matrix products that is computationally expensive, so **NOT** recommended. AI-REML uses the average information of these two methods, which eliminates the trace term, leading to faster computational algorithm. EM-REML if often combined with AI-REML to provide initial estimates.

--use_em_ai_reml

Use EM-REML then switch AI-REML after number iterations

--num_em_reml_burnin <num iterations>

The number of EM –REML iterations prior to switching to AI-REML.

--use_ai_reml

Use AI-REML.

--use_newton_raphson

Use the Hessian matrix of second derivatives of the likelihood function in the information matrix. Option is more computationally more expensive than AI-REML. RECOMMEND DO NOT USE

--use_fisher_information

Use Fisher information matrix, which is the expected value of the Hessian matrix. Option is more computationally expensive than AI-REML. RECOMMEND DO NOT USE

--add_likelihood_constant

The likelihood has a constant term that does not impact maximization. The constant can be dropped or retained in the final output. The default in MMAP is to drop, but this option will retain the constant. The option is used if comparing MMAP results to another program that retains the constant.

--null_likelihood <val>

If this option is present a likelihood ratio test will be performed using the likelihood of the current model and the <val>. The output file <trait>.<suffix>.variance.components.T.csv (see below) will have the p-value of the test in LRT_PVAL. This option is useful for comparing nested models, for example, in IBD or IBS linkage analysis, or constructing null models for score tests, where <val> is LN_LIKE from the baseline model. For example, for linkage a model with the A matrix is run, LN_LIKE extracted, then added to the model with A and the IBD matrix with this option.

Matrix operations

MMAP uses the Intel Math Kernel Library (MKL) http://software.intel.com/en-us/intel-mkl/ for matrix operations. The likelihood equation involves inverting the dense marginal covariance matrix, which is the primary computational challenge. MKL has two main routines for matrix inversion: DPOTRS and DPOTRI. Performance between the two routines seems to depend on the number of threads. MMAP has options to compute the likelihood using single or double precision matrices or a combination of each. Single precision takes have the memory and is twic as fast, but convergence of the maximization routines might be impacted.

--use_dpotrs

Use single precision BLAS routine DPOTRS for matrix inversion.

--use_spotrs

Use single precision BLAS routine SPOTRS for matrix inversion. Requires half the memory and is approximately twice as fast as DPORTS.

--use_spotrs_dpotrs

Use single precision BLAS routine SPOTRS until convergence reaches a pre-specified tolerance, and then switches to double precision BLAS routine DPOTRS until final convergence. The hybrid option is useful for large datasets.

--use_dpotri

Use single precision BLAS routine DPOTRI for matrix inversion.

--use_spotri

Use single precision BLAS routine SPOTRI for matrix inversion. Requires half the memory and is approximately twice as fast as DPORTI.

--use_spotri_dpotri

Use single precision BLAS routine SPOTRI until convergence reaches a pre-specified tolerance, and then switches to double precision BLAS routine DPOTRI until final convergence. The hybrid option is useful for large datasets.

--num_mkl_threads <num threads>

Specifies the number of threads to use for the matrix operations. Default is 1 one thread. Parallelization can provide significant performance increase for computing the variance estimates under the null with large data sets with dense covariance matrices.

--num_iterations <val>

Sets the maximum number of iterations for the variance component estimation. The default is 500 which would only potentially be needed if the optimization were EM-REML only. AI-REML converges generally in 10-20 iterations. A single iteration can be used to generate estimates of the random effects at a user-specified variance estimates.

Setting Initial Estimates

Setting the initial estimates of the variances or the ratio of the variance estimates in general does not have a significant impact on the number of AI-REML iterations. However, one application would be to set the values to external variance estimates to generate estimates of random effects using the –num_iterations 1 option to be used for additional analysis such as prediction.

--initial_variance_component_values <val 1> <val 2> … <val N>

The values of the variances to seed the estimation procedure. Includes the error term

--initial_variance_component_lambdas <val 1> <val 2> … <val N>

The values of the variance component/total variance ratios. Easier to set than the actual variances. For example, if the heritability is assume 0.2 then an initial values would be 0.2 0.8 with a model with the kinship coefficient.

--write_projection_matrix_file <filename>

Write the projection matrix P at the REML estimate under the null to a binary file. Useful for multiple runs where the null does not change to avoid recomputing P. For example, running a single chromosome at a time or changing the definition of snp set to model different pathways. Note that if the null models changes, for example, adding SNPs in conditional analysis, a new P needs to be computed. *Not yet implemented*.

--read_projection_matrix_file <filename>

Read in a projection matrix <filename>. *Not yet implemented*.

Handling Large Data

The memory footprint of the analysis depends on the number subjects and the number of variance components in the model. For example, with 100,000 subjects, the memory needed for each covariance matrix is 100K x 100K x 8 bytes = 80 Gb. If the analysis was estimating the contribution of each human autosome and the X chromosome using genetic data, then the requirement would 80 Gb x 23 = 1.7 Tb. We have implemented options to trade memory for disk reads to be able to reduce the memory footprint to that required by a single variance component independent of the number of components in the model. Thus, the above example with 23 components and error would require 80 Gb. An analysis of 250,000 subjects would require 500 Gb. The running time depends on the number of cores available. Our analysis of additive and dominance components in 90,000 cows using 20 cores required under 5 hours.

--create_inverse_from_disk --write_disk_variance_component_filename <filename 1> <filename 2> …. < filename N>

There must be the same number of filenames as used in **--binary_covariance_filename** option. These files are temporary and can be deleted after the run.

Output files

<trait>.<suffix>.variance.components.T.csv

Contains the estimates and standard errors of the variance components and fixed effects. P-values of the fixed effects and likelihood are also provided.

<trait>.<suffix>.variance.components.model.csv

Contains the estimates of the fixed and random effects for each individual used in the analysis. Zscores of the random effects are also provide for ranking by standard deviations from the population mean. The <trait>_ERROR term can be used as a residual in linear regression. If multiple residual error terms are modeled then the term <trait>_COMBINED_ERROR is the sum of the error terms and can be used as the residual.

Example MMAP commands

Family data: Command run score test with null model BMI = mean + sex + age + sex*age+ a + x + mt + e, where sex, age, and sex-by-age interaction are fixed effects, g is the additive random effect with label A and covariance matrix kinship.bin, x is the X chromosome effect with label X and covariance matrix Xchr.bin and mt is the mitochondrial random effect with label MT and covariance matrix MT.bin. The MT.bin is constructed as a 0-1 matrix grouping maternal lineages. Estimation of the variances use EM-REML for 2 iterations, then AI-REML using DPOTRS and two threads.

```
mmap  --ped pedigree.csv --phenotype_filename phenotype.csv --trait BMI  --
estimate_variance_components --variance_component_filename kinship.bin
Xchr.bin MT.bin --num_em_reml_burnin 2 --use_em_ai_reml  --use_dpotrs  --
variance_component_label A X MT --covariates sex age --interaction age* --
file_suffix G.X --num_mkl_threads 2
```

Simulated test data

Under development