Last Update: 9/28/2014

MMAP Genomic Matrix Calculations

MMAP has options to compute relationship matrices using genetic markers. The markers may be genotypes or dosages. Additive and dominant covariance matrices are available using SNP or pooled variances. Options are available for single or double precision matrices with or without adjusting for missing data. The memory footprint is determined by the size of the subject group requested and number of markers. Options to store the number of markers is available to combine matrices across genomic regions as weighted sums. Parallel calculations are controlled by the number of threads requested. PCs can be extracted from any genomic matrix.

The default option in MMAP is double precision matrix multiplication of the Z'Z where Z' is a subject-by-marker matrix of normalized genotype values. The sums in Z'Z are generally scaled by the number of observed genotypes to compute the average covariance between each pair. The matrix multiplication Z'Z default is double precision (DP). The counts of the observed genotypes between each pair is single precision (SP). DP requires twice the memory and compute time as SP. Thus, the total computational footprint is T= DP + SP = 3SP. There are options available to trade speed/memory for accuracy. First, the double precision multiplication can be replaced by single precision to reduce T to 2SP. For the range of values the genomic matrices, single precision is generally sufficient. Second, if there is no missing data (full genotyping or imputed genotypes) or little missing genotype data, the exact count between subjects can be replaced by the number of markers to eliminate the SP count multiplication. Then T=SP, which is 3 times faster than the DP and exact count model. Also the file size is smaller. The impact on downstream analysis of DP vs SP and exact count vs average may depend the size of your data (number of subjects and markers).

If genomic matrices are to be combined, the counts need to be added to the files, in which case, the two options to compute the average are available.

Options for X chromosome matrices are being added. There is no pedigree information required for calculation of autosomal matrices.

Additive matrices

1. Each SNP has own variance. Generally used in human genetics. Default setting. The average can be taken over the number of observed pairs or the number of markers.

2. SNPs have common variance. Generally used for genomic prediction. Requires –pooled_variance option. The pooled variance uses all the markers.

Dominance matrices

Genotype coding: AA=0, AB=BB=1. Similar to the additive matrix there are two forms

1. Each SNP has its own variance. Default setting. The average can be taken over the number of observed pairs or the number of markers

2. SNPs have common variance. Generally used for genomic prediction. Requires –pooled_variance option. The pooled variance uses all the markers.

KING genomic matrices

Adding the optio –king_homo option will generate the genomic covariance matrix based on Equation 5 in ref [1].

$$\varphi_{ij} = 1/2 - \sum\left(X_m^i - X_m^j\right)^2 \Big/ \sum 2p_m(1-p_m)$$

KING robust covariance matrices are being implemented.

Runtime options

--write_binary_gmatrix_file
Additive matrix calculation
--write_binary_dmatrix_file
Dominance matrix calculation

--binary_genotype_filename <SxM file>
A subject-by-marker binary genotype file
--binary_output_filename <file>
Output filename
--group_size <num>
Controls the subject-by-subject group size that is computed to fill in binary. This option impacts that memory footprint at the subject level. Generally the group_size should be as large as possible for efficiency.

--pooled_variance
Use 2nd definition of above matrices. Note that pooled variance is less impacted by MAF.

--single_pedigree
Generate pairwise covariance values for all subjects, independent of the pedigree.

--write_matrix_counts
Add counts to the binary output file. Required if genomic matrices are to be combined. Default is to write pairwise counts base on observed data.
--use_complete_data_count
Uses the number of markers for n_{ij} in the above formulas to compute the average. Can be combined with –write_matrix_counts.

--num_mkl_threads <num>
Parallelization for MKL matrix multiplication

--autosome
Extract the autosomal SNPs
--chromosome <numbers>
Extract SNP on chromosomes in <numbers>
--genomic_region <chr> <start bp> <stop bp>
Extract SNPs in the genomic region(s) specified.
--marker_set <file>
Use markers in <file>
--subject_set <file>
Use subjects in <file>
--min_minor_allele_frequency <value>
Restrict analysis to markers with MAF greater than <value>. Not used for imputed data.

PC calculations
MMAP computes the PCs then outputs to csv delimited file
--compute_pc_file
MMAP option
--binary_input_filename <file>
Binary covariance file to extract PCs
--max_pcs2print <num>
Print the <num> PCs with largest eigenvalues
--pc_output_filename <file>
Output file for the PCs. Sorted by largest to smallest eigenvalue
--subject_set <file>
Restrict the PC calculation to subjects in the file. Note that in practice PCs are computed for all subjects in binary then used for all downstream analyses independent of the number of subjects in the phenotype file. The PCs restricted to subsets of the subjects are no longer orthogonal, but the impact is minimal.
--print_scaled_pcs
Also print eigenvectors scaled by the sqrt of the eigenvalue

Extracting Values from Matrix

There are two options to extract the pairwise values from a binary covariance matrix and an option to extract the full matrix. Subject set options are being added to be able extract specified subjects.

--**variance_component_matrix_mmap2pairs**

Output each pair once. The order depends on the order of the subjects in the file. The output is the upper diagonal of the matrix

--**variance_component_matrix_mmap2pairs_all**

Output each pair of different subjects twice with opposite orders. The output is the full matrix

--**variance_component_matrix_mmap2matrix**

Rectangular output with the subject ids as the first row.

Both options require:

--binary_input_filename <file> --csv_output_filename <file>

Combining Genomic Matrices

--combine_binary_matrix_files <file list>
List of files to be combined. They must all be of the same count type, that is, have been created with the same --write_matrix_counts option with or without the --use_complete_data_count option.

--binary_output_filename <output file>
Name of the combined genomic matrix file

Example Commands

1. Double precision, pairwise adjustment to average, group size 4000 and autosome markers included. Running time 3xtime(SP).

mmap --write_binary_gmatrix_file --binary_genotype_filename <SxM file> --binary_output_filename study.0.5.bin --group_size 4000 --single_pedigree --num_mkl_threads 4 --min_minor_allele_frequency 0.05 --autosome

2. bash shell commands to create dominance matrix by chromosome then combine into single file. Group size is 1000. Single precision and constant adjustment to average, so running time is time(SP)

for chr in {1..22}
do
mmap --write_binary_dmatrix_file --binary_genotype_filename <SxM file> --binary_output_filename dom.${chr}.0.5.bin --group_size 1000 --single_pedigree --

```
write_matrix_counts --num_mkl_threads 4  --min_minor_allele_frequency 0.05 –
chromosome ${chr} –single_precision –use_complete_data_count
done

filelist=`ls dom.*.0.5.bin`
mmap --combine_binary_matrix_files $filelist  --binary_output_filename dom.auto.0.5.bin
```

3. bash shell commands to create leave-one-chromosome out (LOO) matrices

```
# get file list. Assume all files have name G.<chr>.bin
ls G.*.bin > filelist
list=`cat filelist`
for loo in {1..22}
do
# delete loo chromosome
grep -v $loo filelist >loolist
looinput=`cat loolist`
# combine the 21 gmatrices
$prog --combine_binary_matrix_files  $looinput --binary_output_filename
G.LOO.${loo}.bin
done

# combine for autosome file
$prog --combine_binary_matrix_files  $filelist --binary_output_filename G.autosome.bin
```

1. Manichaikul, A., et al., *Robust relationship inference in genome-wide association studies.* Bioinformatics, 2010. **26**(22): p. 2867-73.