User Manual for REMMA

Chao Ning

China Agricultural University

ningchao@cau.edu.cn; ningchao91@gmail.com

Introduction

A Rapid Epistatic Mixed-model Association Analysis by Linear Retransformations of Genomic Estimated Values

REMMA is written by Python and C and is a rapid method to perform genome-wide epistatic association analysis in the presence of population structure and cryptic relatedness. This method first estimates individuals' epistatic effects by an extended genomic best linear unbiased prediction (EG-BLUP) model with additive and epistatic kinship matrix, then pairwise interaction effects are obtained by linear retransformations of individuals' epistatic effects. For detailed theory, please refer to our paper *A rapid epistatic mixed-model association analysis by linear retransformations of genomic estimated values* (Ning, et al., 2018).

Note: No missing genotypes are allowed by in the program. The software BEAGLE or IMPUTE2 can be used to fill the missing genotype. When the accuracy of the filled-in calls isn't important, the PLINK command *--fill-missing-a2* can be used to simply replace all missing calls with homozygous A2 calls, which may have little influence for relative low missing rate (eg. Less than 0.05 for each SNP).

Dependencies

Anaconda python (Python 2.7)

https://www.anaconda.com/download/

Intel® Math Kernel Library (Intel MKL)

Using the binary of REMMA for Linux, users do need to install Intel MKL (https://software.intel.com/en-us/intel-mkl). However, it is recommended to install Intel® Parallel Studio XE (https://software.intel.com/en-us/intel-parallel-studio-xe), which simplifies the progress of compiling the source code.

Gmat64 https://github.com/chaoning/Gmat64

Input files

PLINK binary file,

Three files: *.bed, *.fam and *bim files.

Phenotypic file

No header and missing phenotypes are allowed. At least 4 columns. The first two columns are family ID and individual ID, which are the same to the *.fam file. The third column is intercept (population mean) is recommended to be 1 always. Other covariates can included from the fourth column. The last column is phenotypic values. Here is an example

```
12659 14462 1 0 126 0 0.58
12659 14463 1 0 91  1 0.39
12659 14464 1 1 126 0 0.37
12659 14465 1 0 91  1 0.9
12659 14466 1 0 91  1 0.84
12659 14467 1 0 91  1 0.61
12659 14468 1 1 91  1 0.84
12659 14469 1 1 91  1 0.59
12659 14470 1 1 124 0 0.13
12659 14471 1 0 124 0 0.72
```

Run

Download the REMMA files.

$path_to_remma = /../../REMMA

cd $path_to_remma/fun

Building dynamic links

icc REMMA_epi_C.c -fPIC -shared -mkl -fopenmp –o REMMA_epi_C.so

icc REMMA_epi_scan_C.c -fPIC -shared -mkl -fopenmp -o REMMA_epi_scan_C.so

icc REMMA_epi_select_C.c -fPIC -shared -mkl -fopenmp -o REMMA_epi_select_C.so

Build the kinship matrix

cd $path_to_remma/example

Gmat64 --bfile plink --inv 1 --normMat 1 --outformat 0 --Gmat addGmat

Gmat64 --bfile plink --inv 1 --normMat 1 --outformat 0 --Gmat epiGmatAA

REMMA_add

Test the significance of additive effects

Script: REMMA_add.py

python REMMA_add.py --prefix plink --pheno pheno --out res_file

--prefix: the prefix for PLINK binary file and kinship file

--pheno: phenotypic file

--out: output file

The output file contain 6 columns. Here is an example

```
SNPID chro chrPos effect chi_val Pvalue
mCV25266528 1.0 0.0 2.275798693808782 0.9842058171465128 0.3211626588393375
gnf01.037.906 1.0 0.0 -1.2971445984179688 0.2044862957435381 0.6511237487167159
petAF067836-350A-1 1.0 0.0 1.3382547684612516 0.24186215632472485 0.6228643906437092
mCV23057534 1.0 0.0 -2.2454208343654694 1.196019783920078 0.2741186489133145
rs13475932 1.0 0.0 1.2470493875357127 0.4237709279731882 0.5150613968139525
gnf01.075.385 1.0 0.0 1.727440789765992 0.80228242782791 0.37041183608100425
mCV23431007 1.0 0.0 -0.6634826682368744 0.11085920527063482 0.73916805528462
mCV23433457 1.0 0.0 -1.4699431911989198 0.5368029287269004 0.463760724911889
UT-1-92.862916 1.0 0.0 1.9644664977881314 0.7215014815051601 0.39565183725021036
CEL-1-111503693 1.0 0.0 1.3873210068104087 0.7388303006599399 0.3900358323255806
mCV22824651 1.0 0.0 0.9385099065575597 0.3339150155822963 0.5633628282308284
mCV24201027 1.0 0.0 3.1700291961925333 3.634303542896059 0.056600281066701015
```

REMMA_pre

The program will prepare the necessary files for subsequent epistatic analysis.

Script: REMMA_pre.py

python REMMA_pre.py --plink_prefix plink --pheno pheno --out_prefix epi --add_test 0/1

--plink_prefix: the prefix for PLINK binary file and kinship file;

--pheno: phenotypic file

--out_prefix: the prefix for the output files, which is used for subsequent epistatic analysis.

--add_test: whether to test the significance of additive effects. 0 means No, while 1 means Yes.

The parameters can be set to 0. If you want to test the significance of additive effects,

REMMA_add.py is OK. The two programs have little difference for additive effects.

Note: Three variances will be estimated in this step. For population with about 1000 individuals, this step may need several minutes. However, for population with more than 10 000 individuals, this step may need more than 2 days depend on your computational condition.

Full REMMA

The program will perform the exhaustive genome-wide epistatic association analysis.

Script: REMMA_epi.py

```
python REMMA_epi.py --plink_prefix plink --out_prefix epi --parallel total i    \
--num_test 50000 --p_threshold 1e-5
```

--plink_prefix: the prefix for PLINK binary file and kinship file;

--out_prefix: the prefix for the output files. This is the same to the REMMA_pre. This program will read the output files from REMMA_pre and generate epistaitic results based on the prefix defined by --out_prefix;

--parallel total i: This program is very time-consuming. So the analysis can be split into tens or even hundreds of parts. For example, --parallel 20 1, --parallel 20 2,….Then merge the results.

--num_test 50000. The number of test stored in the memory during the running. Set depend on the computational memory.

--p_threshold: 1e-5. It is not necessary and be unable to store all the results. The parameters will help to filter the SNP pairs with p values more than the defined.

The program will generate the epistatic test results, with names out_prefix.epires.total-i. The chi-square values not the p-values are generated.

If you want to merge the results and p values, please use the merge.py script.

Eg. python merge.py --prefix out_prefix. epires --p_val 1

The out_prefix.epires.merge file will be generated.

```
order1 order2 chro1 SNP_ID1 pos1 chro2 SNP_ID2 pos2 epi_effect chi_value p_value
1 2 1 mCV25266528 0 1 gnf01.037.906 0 0.00324406 0.654849 0.4183843517280651
1 3 1 mCV25266528 0 1 petAF067836-350A-1 0 -0.00427741 1.00451 0.3162216752150252
1 4 1 mCV25266528 0 1 mCV23057534 0 -1.47746e-06 1.31697e-07 0.9997104470506436
1 5 1 mCV25266528 0 1 rs13475932 0 0.00198325 0.238019 0.6256405726605232
1 6 1 mCV25266528 0 1 gnf01.075.385 0 0.00237819 0.346759 0.555953540342445
1 7 1 mCV25266528 0 1 mCV23431007 0 -0.00121322 0.0910549 0.7628403280703615
1 8 1 mCV25266528 0 1 mCV23433457 0 -0.000422463 0.0106799 0.9176902717434668
1 9 1 mCV25266528 0 1 UT-1-92.862916 0 0.000402117 0.0101944 0.9195764012832617
```

REMMA-scan

REMMA-scan first scans the genome-wide epistatic effects with running time that is linear in the cohort size, and then select top interactions to calculate their P-values.

Script: REMMA_epi_random.py, REMMA_epi_scan.py, REMMA_epi_select.py

(1) Randomly select *S* (e.g. 10^6) SNP pairs and calculate their epistatic effects. Normal distribution is fitted with sample mean and standard deviation of the *S* epistatic effects, and different quantiles are calculated.

python REMMA_epi_random.py --plink_prefix plink --output_prefix pre_file \

--quantile 1e-5 --num_test 50000 --num_random_pair 1000000

--plink_prefix: the prefix for PLINK binary file and kinship file;

--out_prefix: the prefix for the output files. This is the same to the REMMA_pre. This program will read the output files from REMMA_pre and generate epistaitic results based on the prefix defined by --out_prefix;

--quantile 1e-5. Quantiles of empirical normal distribution. Smaller values means less computational time, but may miss some true epistatic effects. Recommend, 100K SNPs, 1e-5; 10K, 1e-4; 1k, 1e-3.

--num_test 50000. The number of test stored in the memory during the running. Set depend on the computational memory.

--num_random_pair 1000000. How many SNP pair generated to build the empirical normal distribution.

The analysis results from the random SNP pairs will be printed in the screen.

```
Information from the  1000000  SNP pairs:
The  1e-05  quantile:  0.0180823344807
The SNP-SNP epistatic sample mean and variance:  -7.73834394956e-05 1.79760044312e-05
The coefficient of the epistatic effects and chi-square value:  0.933615852471
```

In this example, “The 1e-05 quantile: 0.0180823344807” will be used in the subsequent analysis.

(2) Scan the genome-wide epistatic effects and select the top SNP–SNP pairs passing the quantiles.

python REMMA_epi_scan.py --plink_prefix plink --output_prefix pre_file

--parallel total i --num_test 50000 --eff_threshold 0.0180823344807

--eff_threshold: the absolute value of epistatic below the threshold will be filtered.

Merge the results.

python merge.py –prefix epi.epieff --p_val 0

We get the epi.epieff.merge file now.

(3) Examine these top SNP pairs.

python --plink_prefix plink --out_prefix pre_file --SNP_select_file epi.epieff.merge \

--num_test 50000 --p_val 1e-4

--SNP_select_file: The merged file generated at last step.

--p_val 1e-4: the threshold for p value.

Here is an example for result file (epi.epieff.merge_res.addP)

order1	order2	chro1	SNP_ID1	pos1	chro2	SNP_ID2	pos2	epi_effect	chi_value	p_value
120	812	1	rs13476095	125585192	10	CEL-10-113177617	0	-0.0164752	17.7995	2.454477356274718e-05
122	812	1	rs3716165	125938666	10	CEL-10-113177617	0	-0.0164752	17.7995	2.454477356274718e-05
118	812	1	rs3687720	125011741	10	CEL-10-113177617	0	-0.0164682	17.7818	2.4774176224189575e-05
124	812	1	rs3691374	126933528	10	CEL-10-113177617	0	-0.0160834	17.2067	3.3525189852621405e-05
18	1185	1	mCV23509126	0	15	rs13482580	53022566	-0.0168908	15.1862	9.74127161091933e-05
123	812	1	rs6228473	126627262	10	CEL-10-113177617	0	-0.0156884	16.4577	4.974771270051981e-05
121	812	1	rs3090765	125864462	10	CEL-10-113177617	0	-0.0165836	17.9663	2.248506461108792e-05
119	812	1	rs13476094	125295821	10	CEL-10-113177617	0	-0.0164682	17.7818	2.4774176224189575e-05
117	812	1	rs6189020	124619922	10	CEL-10-113177617	0	-0.0164598	17.7615	2.5039930068172687e-05
125	812	1	rs13476100	126946240	10	CEL-10-113177617	0	-0.0160834	17.2067	3.3525189852621405e-05
19	1182	1	CEL-1-182091301	0	15	rs3660290	50857204	-0.0170473	15.9344	6.557599536836168e-05
19	1183	1	CEL-1-182091301	0	15	rs3692040	51075548	-0.0170467	15.9305	6.571124805788694e-05
19	1184	1	CEL-1-182091301	0	15	rs6165881	51817647	-0.0178486	17.1895	3.383010557659132e-05
19	1185	1	CEL-1-182091301	0	15	rs13482580	53022566	-0.0184392	18.4714	1.7247317777375085e-05
392	1253	4	rs13477694	49588593	16	rs4198737	69011459	-0.0171499	16.9288	3.880829795745827e-05
454	806	5	rs3709946	6789766	9	rs6316481	118974445	0.0181275	16.2504	5.549787645067868e-05
468	707	5	rs13478223	45356297	8	rs6287472	13314069	0.0171233	15.3826	8.779301408221543e-05

Reference

Ning, C.*, et al.* (2018) A Rapid Epistatic Mixed-model Association Analysis by Linear Retransformations of Genomic Estimated Values, *Bioinformatics*, bty017-bty017.