User Manual for

Fast3VmrMLM

Fast 3 Variance components **m**ulti-locus **r**andom-SNP-effect **M**ixed **L**inear **M**odel tools for genome-wide association study

(**version 1.0**)

Wang Jing-Tian and Zhang Yuan-Ming

(soyzhang@mail.hzau.edu.cn)

Last updated May, 2025

Disclaimer: The software has undergone comprehensive testing by Yuan-Ming Zhang's Lab at the College of Plant Science and Technology, Huazhong Agricultural University. The results obtained from the software are generally reliable, correct, and appropriate. However, it is important to note that these results are not guaranteed for specific datasets. We strongly recommend that users integrate Fast3VmrMLM results with those from other software packages, i.e., mrMLM, 3VmrMLM and q3VmrMLM.

Download website:

https://github.com/YuanmingZhang65/Fast3VmrMLM

Citation:

Wang J[#], Chen Y[#], Shu G[#], Zhao M[#], Zheng A, Chang X, Li G, Wang Y* and Zhang Y.M.* Machine learning assists in gene mining and breeding by design for polygenic traits using fast, efficient and large-scale genome-wide association studies. Plant Communications, Accepted.

This study was supported by the National Natural Science Foundation of China (32470657; 32270673).

1 Introduction

1.1 Why Fast3VmrMLM?

Fast3VmrMLM (**F**ast **3** **V**ariance components **m**ulti-locus **r**andom-SNP-effect **M**ixed **L**inear **M**odel) is an R package designed for fast and big data genome-wide association studies (GWAS), which consider additive and dominant effects and control for their polygenic backgrounds in a compressed variance component mixed linear model.

The current version, Fast3VmrMLM v1.0, features three modules:

1) Fast3VmrMLM: The genome-wide scanning plus machine learning framework was developed and integrated with advanced computational techniques to identify QTNs for complex traits using fast, efficient and large-scale GWAS algorithm, Fast3VmrMLM.
2) Fast3VmrMLM-Hap: Haplotypes of bins constructed by adjacent linkage disequilibrium markers are used to associate the trait of interest with the markers using the Fast3VmrMLM-Hap module.
3) Fast3VmrMLM-mQTL: To identify mQTLs, the Fast3VmrMLM-mQTL module can be used to associate the trait of interest with molecular variations, where there are two types of formats for the input file of the genotypes of molecular variations, including genes, lncRNA haplotypes and SVs. One is the "*.vcf" format with multi-allele marker, and another is the PLINK binary genotype format with the start and end positions on the genome for molecular markers (see Table 6).

Fast3VmrMLM v1.0 works only on Linux system.

1.2 Getting started

Fast3VmrMLM v1.0 (https://github.com/YuanmingZhang65/Fast3VmrMLM) is a package that runs in the R environment on the Linux system, which can be freely downloaded from the above website or requested from the maintainer, Dr Yuan-Ming Zhang at College of Plane Science and Technology, Huazhong Agricultural University (450625680@qq.com; soyzhang@hotmail.com).

1.2.1 Installation in R environment

Within R environment on the Linux system, the Fast3VmrMLM software can be installed by the following R codes:

First install the dependency packages by:

install.packages(c("Rcpp", "RcppArmadillo", "RcppParallel", "Matrix", "BH", "data.table", "openxlsx", "dplyr", "MASS", "lars", "SKAT", "KScorrect", "BEDMatrix"))

Then install the Fast3VmrMLM package from local files by:

install.packages("E:/Fast3VmrMLM_1.0.zip",repos=NULL,type="win.binary")

1.2.2 Installation on the Linux system

Within the Linux system, the Fast3VmrMLM software can be installed by the following process:

First, create a new conda environment named Fast3VmrMLM and install ependency packages by the following bash codes:

conda create -n "Fast3VmrMLM" r-essentials r-base=4.3
conda activate Fast3VmrMLM
conda install -c conda-forge r-Rcpp
conda install -c conda-forge r-RcppArmadillo
conda install -c conda-forge r-RcppParallel
conda install -c conda-forge r-data.table
conda install -c conda-forge r-MASS
conda install -c conda-forge r-dplyr
conda install -c conda-forge r-openxlsx
conda install -c conda-forge r-BH
conda install -c conda-forge boost
conda install -c conda-forge r-SKAT r-lars r-KScorrect r-BEDMatrix

Then, decompress Fast3VmrMLM.zip with bash code and install it with R code by:

unzip '/user/Fast3VmrMLM_Linux.zip' -d '/user/'
R
install.packages("/home/user/Fast3VmrMLM", repos = NULL)

User Manual file Users can decompress the Fast3VmrMLM package and find the User Manual file (name: Instruction.pdf) in the folder of "…/

Fast3VmrMLM/inst".

Example datasets Users can decompress the Fast3VmrMLM package and find the example datasets in the folder of "…/Fast3VmrMLM/extdata".

1.2.3 Run Fast3VmrMLM

Once the software Fast3VmrMLM is installed, users may run it using two commands:

library("Fast3VmrMLM")

Fast3VmrMLM(***) (please see the example of § 2.2)

Before using Fast3VmrMLM in Linux, make sure to activate the conda environment associated with Fast3VmrMLM: *conda activate Fast3VmrMLM*. Users need to run *library("Fast3VmrMLM")* every time before utilizing the Fast3VmrMLM software package.

2 Function Fast3VmrMLM

2.1 Parameter settings

Parameter	Meaning	File format	Note
fileGen	Name & path of genotypic file in your device, e.g., fileGen="D:/Users/Genotype".	PLINK binary files: **Genotype.bed+Genotype.bim+Genotype.fam**	
filePhe	Name & path of trait phenotypic file in your device, e.g., filePhe="D:/Users/Phenotype.csv".	*.csv (Phenotypic values. Row: individual; Column: traits)	**Table 1**
fileKin	Name & path of individual kinship file in your device, e.g., fileGRM="D:/Users/GRM.csv" or fileGRM=NULL.	*.csv (GRM. Row & Column: individuals)	**Table 2**
filePS	Name & path of covariates file in your device, e.g., filePS="D:/Users/covariates.csv" or filePS=NULL.	*.csv (Population structure. Row: individual; Column: sub-populations or covariates, e.g., sex and age)	**Tables 3-5**
PopStrType	The types of population structure include “structure” (*Q matrix*), “PC” (*principal components or covariates*), and “Evol” (*evolutionary population structure*).		
fileOut	Save path of the result in your device, e.g., "D:/Users/".		
genoType	Setting the algorithm as “SNP” (SNP-based Fast3VmrMLM algorithm), “Hap” (Haplotype-based Fast3VmrMLM-Hap algorithm) and “molecular” (molecular-based Fast3VmrMLM-mQTL algorithm)		
trait	Traits analyzed from number 1 to number 2, e.g., trait=1:3 indicates that users analyze the first to third traits.		
svrad	A physical distance of sliding window for removing potential candidate variants with the collinearity of the most significant one. Default value is svrad=2.0e+4 (bp). Users can obtain more potential associated variants by setting a small value of SearchRadius.		
svpal	A critical *P*-value (default svpal=1.0e-5) to select all the potentially associated variants in genome-wide single-variant scanning. The size of svpal may be changed based on sample size, such as from 1.0e-5 to 1.0e-2.		
svmlod	A critical LOD score, which is larger than 0, (default svmlod=3), is used to select suggested variants.		
SampleMarkersforGRM	A parameter indicates whether using a subset of variants to construct kinship (TRUE) or not (FALSE). Default value is SampleMarkersforGRM=FALSE.		
SampleMarkersforGRMNum	A parameter (>0) indicates the number of variants sampled to calculate kinship. Only used when SampleMarkersforGRM=TRUE. Default value is SampleMarkersforGRMNum=1.0e+4.		
c_threshold	A parameter for evaluating linkage disequilibrium of adjacent variants when constructing bin-based haplotypes, ranging from 0 to 1. Default value is c=0.7.		
numofHaplotypes	A parameter for setting the number of haplotypes for each bin genotype in the Fast3VmrMLM-Hap module. Default value is numofHaplotypes=3.		
scainngParallel	A parameter indicates whether using parallel computing in genome-wide scanning. Default value is scainngParallel=FALSE.		
nThreads	A parameter indicates the number of cores used in parallel computing. Default value is nThreads=20.		

DrawPlot	A parameter indicates whether drawing and outputting the Manhattan plot based on the GWAS results. Default value is DrawPlot=FALSE.
Plotformat	A parameter indicates the format of the Manhattan plot. Default value is Plotformat=*.tiff.
MGinputClass	Setting the input file format "bed" (PLINK binary file) and "vcf" (.vcf format file), when variableType="molecular" in the Fast3VmrMLM-mQTL module. Default value is inputClass="vcf".
filegeneRegion	Setting the input file that indicates the genome intervals of each molecular marker when inputClass="bed" in the Fast3VmrMLM-mQTL module.

2.2 Parameter settings

The running code for Fast3VmrMLM is as follows:

Fast3VmrMLM(fileGen="D:/Users/Genotype",filePhe="Kin=NULL,filePS="D:/Users/PopStr.csv",PopStrType="PC",fileOut="D:/Users/",genoType="SNP",trait=1,svrad=2e+4,svpal=1e-5,svmlod=3,scainngParallel=TRUE,nThreads=20,DrawPlot=FALSE,Plotformat=".tiff")*

The running code for Fast3VmrMLM-Hap algorithms:

Fast3VmrMLM(fileGen="D:/Users/Genotype",filePhe="Kin=NULL,filePS="D:/Users/PopStr.csv",PopStrType="PC",fileOut="D:/Users/",genoType="Hap",trait=1,svrad=2e+4,svpal=1e-5,svmlod=3,c_threshold=0.7,numofHaplotypes=3,DrawPlot=FALSE,Plotformat=".tiff")*

The running code for Fast3VmrMLM-mQTL algorithms:

Fast3VmrMLM(fileGen="D:/Users/Genotype",filePhe="Kin=NULL,filePS="D:/Users/PopStr.csv",PopStrType="PC",fileOut="D:/Users/",genoType="molecular",trait=1,svrad=2e+4,svpal=1e-5,svmlod=3,MGinputClass="vcf")

Users must set "fileGen", "filePhe", "trait", and "fileOut", while the other parameters may be default in function Fast3VmrMLM (see § 2.1).

2.2.1 Data input format

Format for genotypic dataset "fileGen"

The file type of genotypes is "plink binary format" (Genotype.bed + Genotype.bim + Genotype.fam), which can be found the introduction from PLINK v1.9 (https://www.cog-genomics.org/plink2/.

Format for genotypic dataset "filePhe" (Table 1)

The file type of phenotypes for complex trait is *.csv, following the format outlined in Table 1. The first row in the first column: "<Phenotype>"; the second to nth rows in the first column: individual IDs or names, such as Ind46. The first row in other columns: trait names, such as “trait1”, and the second to nth rows in other columns: values of phenotypic traits. The phenotypes missed: “NA”.

Table 1. The format of phenotypic dataset

<Phenotype>	trait1	trait2	trait3	…
Ind46	91.03	88.32	87.67	…
Ind52	103.11	103.10	98.36	…
Ind57	92.07	116.30	NA	…
Ind64	128.5	101.20	101.02	…
Ind68	95.84	91.74	94.50	…
⋮	⋮	⋮	⋮	…

Format for genotypic dataset “fileKin” (Table 2)

Table 2. The format of knship dataset

Ind_1	Ind_2	Ind_3	Ind_4	Ind_5	Ind_6
1	0.700361011	0.599277978	0.675090253	0.620938628	…
0.700361011	1	0.620938628	0.666064982	0.653429603	…
0.599277978	0.620938628	1	0.561371841	0.5433213	…
0.675090253	0.666064982	0.561371841	1	0.615523466	…
0.620938628	0.653429603	0.5433213	0.615523466	1	…
⋮	⋮	⋮	⋮	⋮	…

The “fileKin” should be a file with *.csv format. All the kinship coefficients are listed as an n × n matrix. Both rows and columns represent individuals that arranged in the order of the *.fam file. The parameter “fileKin=NULL” indicates that the kinship matrix is calculated by the “Fast3VmrMLM” software. When fileKin="D:/Users/kinship.csv", the kinship matrix with name kinship.csv is uploaded from the folder "D:/Users". If the number and order of individuals in the "kinship.csv" file do not match those in the phenotypic files, our software will attempt to match them.

Q matrix format for dataset “filePS” (Table 3)

The Q matrix dataset in Table 3 consists of a (n+1) × (k+1) matrix, where n is sample size (the number of the above common individuals), and k is the

number of sub-populations. The first column is "<Structure>" and individual IDs or names. In the 2nd to (*k*+1)-th columns, "Q1" to "Qk" indicate sub-populations. In the second row, "0.014", "0.972" and "0.014" are posterior probabilities that the individual "33-16" is belong to the 1st, 2nd, and 3rd subpopulations, respectively.

Table 3. The Q matrix format of dataset filePS

<Structure>	Q1	Q2	Q3
33-16	0.014	0.972	0.014
Nov-38	0.003	0.993	0.004
A4226	0.071	0.917	0.012
A4722	0.035	0.854	0.111
⋮	⋮	⋮	⋮

Principal components or covariates for dataset "filePS" (Table 4)

The principal components or covariates dataset in Table 4 consists of a (*n*+1) × *k* matrix, where *n* is sample size (the number of the above common individuals), and *k* is the number of principal components and/or covariates. The first column is "<PCA>" and individual IDs or names. The 2nd to *k*-th columns indicate principal components and/or covariates.

Table 4. The principal components or covariates format of dataset filePS

<PCA>	PC1	PC2	PC3	…
33-16	0.306	0.029	0.226	…
Nov-38	-0.708	-0.271	1.413	…
A4226	-2.330	0.116	-0.824	…
A4722	1.059	0.470	-0.135	…
⋮	⋮	⋮	⋮	

The evolutionary population structure for dataset "filePS" (Table 5)

The evolutionary population structure dataset in Table 5 consists of a (*n*+1) × 2 matrix, where n is sample size and the number of categories for variables is the number of sub-populations. The first column is "<EvolPopStr>" and individual IDs or names. The 2nd column indicates the evolutionary sub-population. Other population structure described by character type of variables are supported in this format.

Table 5. The evolutionary population structure format of dataset filePS

<EvolPopStr>	EvolType
33-16	A
Nov-38	B
A4226	A
A4722	C
⋮	⋮

The format of the chromosome intervals of each molecular marker for the parameter filegeneRegion in § 2.1 (Table 6)

When genoType="molecular" and MGinputClass="bed", the genome intervals of each molecular marker should be set by users via the file in Table 6. The first column is its ID, the second column is its chromosome, the third and the fourth columns are the left and right physical positions, respectively.

Table 6. The chromosome intervals of each molecular marker

Zm00001eb000010	1	34617	40204
Zm00001eb000020	1	41214	46762
⋮	⋮	⋮	⋮

2.2.1 Result

The results include three files: *_intermediate.csv (intermediate results), *_result.csv (final results) and a Manhattan plot file.

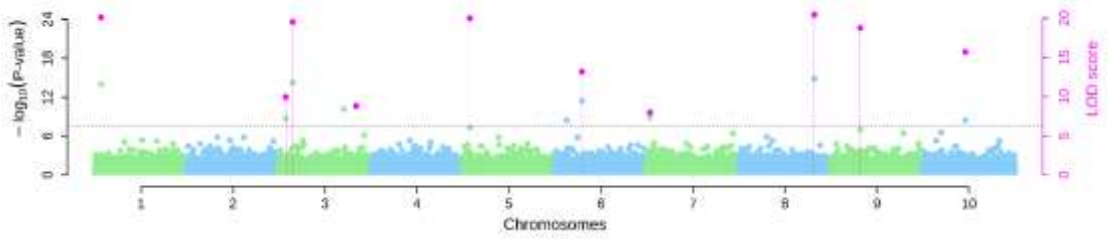
*_intermediate.csv: This file contains the results of genome-wide single-variants scanning in the first step. In this file, all the columns are named as "MarkerID" (variants name), "CHR" (Chromosome), "POS" (variants position (bp) on the genome), and "pval" (the *P*-value for main-effect variants).

MarkerID	CHR	POS	pval
PZB00859.1	1	157104	0.292043111
PZA01271.1	1	1947984	0.185246808
PZA03613.2	1	2914066	0.99208603
PZA03613.1	1	2914171	0.999987108
⋮	⋮	⋮	⋮

*_result.csv: The final results for significant and suggested variants. In this file, all the columns are named as “Marker” (marker name); “Chromosome”; “Position (bp)” (markers position (bp) on the genome); “add” (additive effect); “dom” (dominance effect); “LOD” (LOD score); variance (the variance of each variants); “r2(%)” (the proportion of total liability variance explained by each variants); “*P*-value” (calculated from LOD score using χ^2 distribution); “significance” (significant (SIG) variants are based on Bonferroni correction, that is, critical *P*-value is 0.05/*m*, where *m* is the number of tests or variants, while suggested (SUG) variants are based on LOD ≥ 3.0, default).

Trait ID	Trait name	Marker	Chromosome	Position (bp)	LOD	add	dom	variance	r2(%)	P-value	signific
1	phe_1	null_653	1	654	31.5242	-0.4295	5.6711	7.9444	5.0209	3.00e-32	SIG
1	phe_1	null_20728	3	20729	9.9457	1.6284	2.7827	4.3464	2.7469	1.13e-10	SIG
1	phe_1	null_21489	3	21490	19.5296	1.2742	4.3724	6.1609	3.8937	2.96e-20	SIG
1	phe_1	null_28367	3	28368	8.7913	2.5226	0.91	3.4508	2.1809	1.62e-09	SIG
1	phe_1	null_40745	5	40746	20.4689	3.7475	-1.434	2.3409	1.4795	3.40e-21	SIG
1	phe_1	null_52928	6	52929	13.1912	3.5793	-0.163	6.4797	4.0952	6.45e-14	SIG
1	phe_1	null_60322	7	60323	7.9802	1.0273	2.853	3.4603	2.1869	1.05e-08	SIG
1	phe_1	null_78191	8	78192	65.6504	6.5965	-1.579	5.054	3.1942	2.25e-66	SIG

*_Manhattan plot: Y-axis on the left-side reports -log10 *P*-values of variants, which are obtained from single-variant genome-wide scanning for all the variants in the first step of Fast3VmrMLM, while Y-axis on the right-side reports LOD scores, which are obtained from likelihood ratio test for suggested and significant variants, with the suggested threshold of LOD = 3.0 (dashed line), in the second step of Fast3VmrMLM. Users can set different LOD threshold by setting svmlod (see § 2.1). These LOD scores are shown in points with straight lines. If LOD score ≥ 20, the LOD scores obtained are transformed as LOD’ = 20 + (LOD – 20)/100 in order that the Manhattan plot is more beautiful.



We recommend that all the significantly and suggested associated markers with the traits of interest are listed in a supplemental table, while all the all the significant and suggested QTNs with known and candidate genes are marked in the Manhattan plot.

3 Reference

1 Wang J, Chen Y, Shu G, Zhao M, Zheng A, Chang X, Li G, Wang YB and Zhang YM. Machine learning assists in gene mining and breeding by design for polygenic traits using fast, efficient and large-scale genome-wide association studies. Plant Communications 2025, Accepted.