# Fixed and random model Circulating Probability Unification

## User Manual for FarmCPU

(Version 1.02)

**Last updated on Dec 21, 2016**

**Disclaimer**: While extensive testing has been performed by Zhiwu Zhang Lab at Washington State University respectively. Results are, in general, reliable, correct, and appropriate. However, results are not guaranteed for any specific set of data. We strongly recommend that users validate FarmCPU results with other software packages, such as GAPIT[1], and TASSEL[2].

**Support documents**: Extensive support documents, including this user manual, source code, demo script, demo data and demo results, are available at FarmCPU website of Zhiwu Zhang Laboratory: http://zzlab.net/FarmCPU

**Questions and comments**: Users and developers are recommended to post questions and comments at FarmCPU and GAPIT joint forum: https://groups.google.com/forum/#!forum/gapit-forum. Answers from other users and developers are appreciated. The FarmCPU and GAPIT team members will periodically go through these questions and comments and address them accordingly. Requests for fixing problems can be directly sent to the first author of FarmCPU paper: Xiaolei Liu (xll19870827@hotmail.com).

**Citation**: Liu X, Huang M, Fan B, Buckler ES, Zhang Z (2016) Iterative Usage of Fixed and Random Effect Models for Powerful and Efficient Genome-Wide Association Studies. PLoS Genet 12(2): e1005767. doi:10.1371/journal.pgen.1005767

The FarmCPU project is supported by the USDA-ARS, the National Science Foundation,
the Agricultural Research Center at Washington State University, and Washington Grain Commission.

# Contents

# 1 INTRODUCTION

## 1.1 Why FarmCPU?

With substantial success in finding genes underlying human diseases and agriculturally economic traits, Genome-Wide Association Study (GWAS) has been suffering two controversial problems: overwhelmed false positives and painful false negatives. Population structure and Kinship among individuals can be incorporated as covariates in a Mixed Linear Model (MLM) to control false positives. But the confounding problem between the covariates and test marker also weakens the signals of QTNs (Quantitative Trait Nucleotides), resulting false negatives. Here we present a user-friendly R GWAS package called FarmCPU (Fixed and random model Circulating Probability Unification), it implemented a method we recently developed to address the “confounding problem” and used several mathematical or programming strategies to increase the speed and save memory in order to make FarmCPU adapted for big data. Now, FarmCPU can deal with a data with 500,000 samples and 500,000 markers in three days.

For further help or report problems, please contact:
Xiaolei Liu: xll19870827@hotmail.com

## 1.2 How to use FarmCPU user manual?

The chapter 1 is a brief introduction to FarmCPU and how to use FarmCPU’s user manual. The chapter 2 describes how to install FarmCPU; the chapter 3 describes the format of input data; the chapter 4 describes how to run FarmCPU; the chapter 5 describes the results list; the chapter 6 lists frequently asked questions and answers.

# 2 Installing FarmCPU and a quick start

R software and several libraries should be installed before using FarmCPU.

R software can be freely downloaded from http://www.r-project.org/.

FarmCPU uses two libraries: bigmemory, biganalytics and GAPIT R package.

Library ‘bigmemory’ can be installed by typing these command lines: (you are required to choose a cite to download and you only need to install the packages once in your computer):

```
install.packages("bigmemory")
install.packages("biganalytics")
```

Once the above packages are installed, these libraries can be imported to R environment by typing these commands:

```
library("bigmemory")
library("biganalytics")
```

The GAPIT package and an initial R library can be installed by typing this command line:

```
library("compiler") #this library is already installed in R
source("http://zzlab.net/GAPIT/gapit_functions.txt")
```

The FarmCPU package can be installed by typing this command line:

```
source("http://zzlab.net/FarmCPU/FarmCPU_functions.txt")
```

The easiest way to use FarmCPU is to COPY/PASTE following codes after you successfully installed bigmemory and biganalytics packages.

```
library("bigmemory")
library("biganalytics")
library("compiler") #this library is already installed in R
source("http://zzlab.net/GAPIT/gapit_functions.txt")
source("http://zzlab.net/FarmCPU/FarmCPU_functions.txt")
```

Now create a directory and set it as your working directory in R. Download and decompress the myFarmCPU zip file:

```
setwd("C:\\myFarmCPU")  #example for windows; if you use linux or mac, change the file path in the quote
```

A quick start of FarmCPU is to run following codes.

```
#Step 1: Set working directory and import data
setwd("C:\\myFarmCPU")
myY  <- read.table("mdp_traits_validation.txt", head = TRUE)
myGM <- read.table("mdp_SNP_information.txt", head = TRUE)
myGD <- read.big.matrix("mdp_numeric.txt", type="char", sep="\t", head = TRUE)

#Step 2: Run FarmCPU
myFarmCPU <- FarmCPU(
                        Y=myY[,c(1,8)],
                        GD=myGD,
                        GM=myGM
                        )
```

After a few seconds, GWAS results will be saved in your working directory. The results include one or two .csv files ("GWAS.Results.csv", "CVeffect.csv") and two .pdf files ("Manhattan-Plot.Genomewise.pdf", "QQ-Plot.pdf").

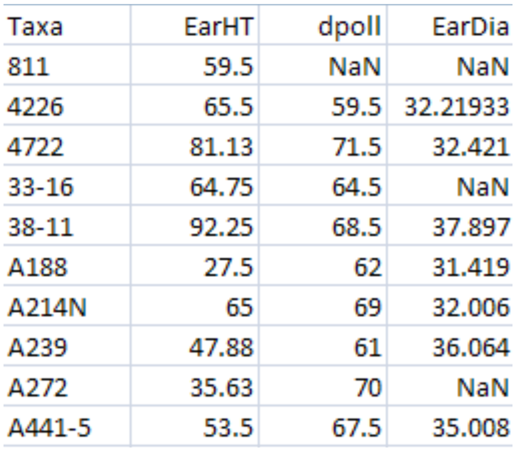
# 3 Data

FarmCPU needs genotypic data (GD), phenotypic data (Y) and genotypic map (GM) data at least. Covariate variables, such as principle components or Q matrix and other fixed effects are optional. So, we have to prepare three files (genotypic data + phenotypic data + genotypic map data) or four files (genotypic data + phenotypic data + genotypic map data + covariate data).

*Notice: All files should be sorted in a fixed order based on the taxa names and all files should be saved in "tab" delimited.*

## *3.1 Phenotypic Data*

FarmCPU use the same phenotypic data format with GAPIT. The phenotypic data includes one taxa column and one or multiple columns of phenotypes. Missing phenotypic data should be marked as “NA” or “NaN”. The first ten rows and four columns in a tutorial data (mdp_traits.txt) are displayed as follows:



| Taxa | EarHT | dpoll | EarDia |
|---|---|---|---|
| 811 | 59.5 | NaN | NaN |
| 4226 | 65.5 | 59.5 | 32.21933 |
| 4722 | 81.13 | 71.5 | 32.421 |
| 33-16 | 64.75 | 64.5 | NaN |
| 38-11 | 92.25 | 68.5 | 37.897 |
| A188 | 27.5 | 62 | 31.419 |
| A214N | 65 | 69 | 32.006 |
| A239 | 47.88 | 61 | 36.064 |
| A272 | 35.63 | 70 | NaN |
| A441-5 | 53.5 | 67.5 | 35.008 |

The file is “Tab” delimited. The first row consists of column labels (i.e., headers). The column labels indicate the phenotype name, which is used for the remainder of the analysis.

The phenotypic file can be input to R by typing command line:

```
myY  <- read.table("mdp_traits_validation.txt", head = TRUE)
```

## 3.2 Genotypic Data

FarmCPU accepts genotypic data in two numeric formats (Numeric format in columns and Numeric format in rows) and we also provide a function that can transform Hapmap format to Numeric format in column (Details see Tutorials part, we will update the codes soon).

### 3.2.1 Numeric format in rows

FarmCPU accepts the numeric format used by EMMA and GAPIT. Following description of “Numeric format in rows” is from GAPIT manual (GAPIT Manual: 2.2.2 Numeric format): Columns are used for SNPs and rows are used for taxa. This format is problematic in Excel because the number of SNPs used in a typical analysis exceeds the Excel column limit.

Homozygotes are denoted by “0” and “2” and heterozygotes are denoted by “1” in the “GD” file. Any numeric value between “0” and “2” can represent imputed SNP genotypes. The first row is a header file with SNP names, and the first column is the taxa name.

Example file (mdp_numeric.txt from tutorial data set):

| taxa | PZB00859.1 | PZA01271.1 | PZA03613.2 | PZA03613.1 | PZA03614.2 | PZA03614.1 | PZA00258.3 | PZA02962.13 |
|---|---|---|---|---|---|---|---|---|
| 33-16 | 2 | 0 | 0 | 2 | 2 | 2 | 2 | 2 |
| 38-11 | 2 | 2 | 0 | 2 | 2 | 2 | 0 | 2 |
| 4226 | 2 | 0 | 0 | 2 | 2 | 2 | 0 | 2 |
| 4722 | 2 | 2 | 0 | 2 | 2 | 2 | 1 | 2 |
| A188 | 0 | 0 | 0 | 2 | 2 | 2 | 0 | 2 |

This file is read into R by typing the following command line:

```
myGD <- read.big.matrix("mdp_numeric.txt", type="char", sep="\t", head = TRUE)
```

### 3.2.2 Numeric format in columns

For Numeric format in columns, Columns are used for taxa and rows are used for SNPs. Here is an example that contains 5 samples and each sample has 8 markers.

*Note: The SNPs in the genotypic data and* genotypic map *data files NEED to be in the same order.*

| 33-16 | 38-11 | 4226 | 4722 | A188 |
|---|---|---|---|---|
| 2 | 2 | 2 | 2 | 0 |
| 0 | 2 | 0 | 2 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 |
| 2 | 0 | 0 | 1 | 0 |
| 2 | 2 | 2 | 2 | 2 |

The genotypic file can be input to R by typing command line:

```
myGD <- read.big.matrix("mdp_numeric.txt", type="char", sep="\t", head = TRUE)
```

## *3.3 Genotypic map data*

FarmCPU accept the genotypic map data format used by GAPIT ("GM" data in GAPIT). Following description of "SNP information data" is from GAPIT manual (GAPIT Manual 2.2.2 Numeric format):

The "GM" file contains the name and location of each SNP. The first column is the SNP id, the second column is the chromosome, and the third column is the base pair position. As seen in the example, the first row is a header file.

Example file (mdp_SNP_information.txt from tutorial data set):

| SNP | Chromosome | Position |
|---|---|---|
| PZB00859.1 | 1 | 157104 |
| PZA01271.1 | 1 | 1947984 |
| PZA03613.2 | 1 | 2914066 |
| PZA03613.1 | 1 | 2914171 |
| PZA03614.2 | 1 | 2915078 |
| PZA03614.1 | 1 | 2915242 |
| PZA00258.3 | 1 | 2973508 |
| PZA02962.13 | 1 | 3205252 |

The genotypic map file is read into R by typing the following command line:

```
myGM <- read.table("mdp_SNP_information.txt", head = TRUE)
```

## 3.4 Covariate variables

Covariate variable file usually includes some fixed effects such as age, sex and population structure matrix[3]. This file is similar to phenotype file but covariate variables file do not have the first taxa column.

Example file (mdp_population_structure.txt from tutorial data set. *Notice: (1) the first column of example file is taxa column, FarmCPU need a covariate variables file without taxa column; (2) Covariate variables file and phenotype file should in the same order*):

| PC1 | PC2 | PC3 |
|---|---|---|
| -24.61260272 | 2.929758052 | 2.378974655 |
| 19.01517623 | 35.99961149 | -3.534235921 |
| -18.75445031 | 5.755455338 | 1.626373633 |
| -21.25806895 | 1.568326634 | -0.075703399 |
| -21.22114006 | -8.02247851 | 2.041586307 |
| 19.77723042 | 37.5455684 | -3.211485814 |
| 14.40888008 | 34.14099124 | -1.297498297 |
| 26.343461 | 40.85397514 | -4.159001978 |
| 27.63342477 | 42.8150698 | -4.218617699 |
| -20.87368399 | 5.337665874 | 4.869219547 |

This file is read into R by typing the following command line:

```
myCV  <- read.table("Copy of Q_First_Three_Principal_Components.txt", head = TRUE)
```

## 3.5 Prior data

Prior data is used to add prior knowledge into FarmCPU model. The format is to add a probability column after genotypic map file. When FarmCPU selects the possible QTNs, p values of prior SNPs are calculated by original p value multiply by probability.
Example file:

| SNP | Chromosome | Position | Probability |
|---|---|---|---|
| PZB00859.1 | 1 | 157104 | 1.00E-30 |
| PZA02841.3 | 2 | 167759219 | 1.00E-30 |
| PZA03054.5 | 3 | 31695534 | 1.00E-30 |
| PZA00627.1 | 3 | 56747948 | 1.00E-30 |

This file is read into R by typing the following command line:

```
myPrior  <- read.table("prior.txt", head = TRUE)
```

## 3.6 Other FarmCPU Input Parameters

In addition to the above parameters, FarmCPU has more parameters to provide more options to optimize the results. (See Gallery of FarmCPU Input Parameters).

## Gallery of FarmCPU input parameters

| Parameter | Default | Options | Description |
|---|---|---|---|
| **Y** | NULL | Users | Phenotype |
| **GD** | NULL | Users | Genotype |
| **GM** | NULL | Users | Genotypic map |
| **CV** | NULL | Users | Covariate variables |
| **GP** | NULL | Users | A P value vector used for select possible QTNs at first iteration |
| **Yt** | NULL | Users | A phenotype used for cross validation test that has the same taxa with Y, but some taxa have a value which is NA in Y |
| **DPP** | 100000000 | Users | How many points will be selected for Manhattan plots |
| **kinship.algorithm** | FARM-CPU | FARM-CPU | Choose FarmCPU method |
| **file.output** | TRUE | TRUE/FALSE | Decide whether output plots and tables or not |
| **cutOff** | 0.01 | Users | Bonferroni test threshold = cutOff / number of markers, this will set a significant level in Manhattan plots |
| **method.GLM** | FarmCPU.LM | FarmCPU.LM/lm/fast.lm | Function used to solve a fixed effect model |
| **method.sub** | reward | reward/mean/median/penalty | Substitution method |
| **method.sub.final** | reward | reward/mean/median/penalty | Substitution method in last iteration |
| **method.bin** | static | static/optimum | Default is "static", it uses a fixed combination of bin.size and bin.selection; If set to “optimum”, users can set bin.size and bin.selection or use the default set of bin.size and bin.selection to do optimization of possible QTN window size and number of possible QTNs selected into FarmCPU model. |
| **bin.size** | c(5e5,5e6,5e7) | Users | The window size used to select one possible QTN |
| **bin.selection** | seq(10,100,10) | Users | Number of possible QTNs will be selected into the model |
| **memo** | NULL | Users | Add memo in the name of output files |
| **Prior** | NULL | Users | Add Possible QTNs into model |
| **ncpus** | 1 | Users | Number of cpus selected for calculation |
| **maxLoop** | 10 | Users | Maximum number of iterations allowed |
| **threshold.output** | 0.01 | Users | P value smaller than threshold.output will be output in GWAS table |
| **WS** | c(1e0,1e3,1e4,1e5,1e6,1e7) | Users | For power-fdr test, a QTN was detected if a SNP within WS on either side was detected |
| **alpha** | c(.01,.05,.1,.2,.3,.4,.5,.6,.7,.8,.9,1) | Users | Calculate power in different type 1 error |
| **maxOut** | 100 | Users | How many rows in the output power-fdr file |
| **QTN.position** | NULL | Users | The rank number of QTN position, only used for mark the QTNs in Manhattan plot |
| **converge** | 1 | 0<&<1 | Decide the percentage of overlapped possible QTNs in two neighbor iterations |
| **iteration.output** | FALSE | TRUE/FALSE | Decide whether ouput results of iterations |
| **model** | A | A | Choose additive model |
| **MAF.calculate** | FALSE | FALSE/TRUE | Calculate minor allele frequency (MAF) or not, if set to TRUE, the SNPs with a lower MAF (<maf.threshold) will be deleted |
| **plot.style** | FarmCPU | FarmCPU | Style of output figures |
| **p.threshold** | NA | NA/a threshold number | Only p value smaller than p.threshold can be selected into the model for the first iteration; the default (NA) means bonferroni test threshold |
| **QTN.threshold** | 0.01 | 0<&<1 | Only p value smaller than p.threshold can be selected into the model since the second iteration |
| **maf.threshold** | 0.05 | Users | When MAF.calculate=TRUE, the SNPswith a lower MAF (<maf.threshold) will be deleted |
| **bound** | NULL | Users | bound is used to set the maximum number of possible QTNs selected for optimization, When bound = NULL,  the maximum number is set to sqrt(n)/sqrt(log10(n)), where n is sample size |

# 4 Analysis

FarmCPU is designed for GWAS and GS on large data. This section illustrates the important parameters in FarmCPU and how to optimize the results.

## 4.1 Introduction of FarmCPU Algorithm

Start: Set of pseudo QTNs (pQTNs) (pQTNs=empty);
Step 1: (GLM) Test marker one at a time with pQTNs as co-factors in a fixed model;
Step 2: (Substitution) P values substitution for the markers corresponding to pQTNs by the minimum P values of each pQTN;
Step 3: (find bins) Find a set of bins such that a random model has the restricted maximum likelihood by fitting a random effect of individual with variance structure defined by the representatives of these bins (strongest associated markers in each bin);
Step 4: Use the representatives of optimized bins as pQTNs;
Step 5: Go back to step 1 unless no change on pQTNs or iterations reaching the maximum allowed.

## 4.2 *Important parameters*

There are eight important parameters in FarmCPU that effect the results: “CV”, “p.threshold”, “maf.threshold”, “method.bin”, “bin.size”, “bin.selection”, “maxLoop” and “Prior”. In each specific analysis, optimizing the parameters can lead to a better result, details see tutorials.

# 5 Results

FarmCPU uses GAPIT function to produce results, such as Manhattan plot, QQ plot, GWAS results table and Effect table of user-provide covariates. Following description of “Manhattan Plot”, “QQ Plot” and “Association Table” are from GAPIT manual (GAPIT Manual 5 Results).

## *5.1 Manhattan Plot*

The Manhattan plot is a scatter plot that summarizes GWAS results. The X-axis is the genomic position of each SNP, and the Y-axis is the negative logarithm of the *P*-value obtained from the GWAS model (specifically from the *F*-test for testing $H_0$: No association between the SNP and trait). Large peaks in the Manhattan plot (i.e., “skyscrapers”) suggest that the surrounding genomic region has a strong association with the trait.

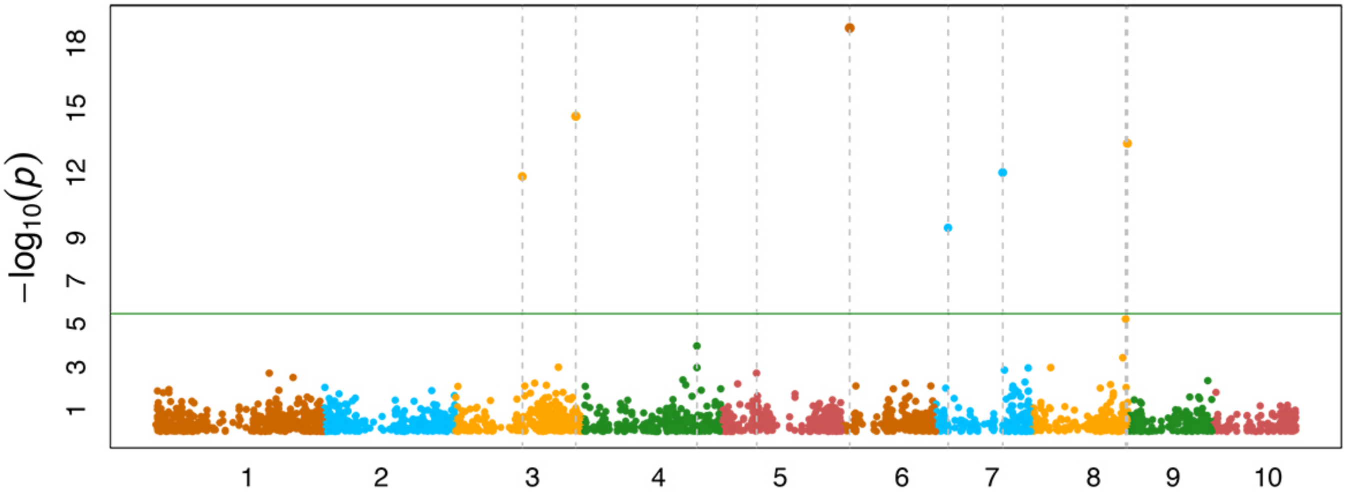


**Figure 5.1.** Manhattan plot. The X-axis is the genomic position of the SNPs in the genome, and the Y-axis is the negative log base 10 of the *P*-values. Each chromosome is colored differently. SNPs with stronger associations with the trait will have a larger Y-coordinate value.

## *5.2 QQ plot*

The quantile-quantile (QQ) –plot is a useful tool for assessing how well the model used in GWAS accounts for population structure and familial relatedness. In this plot, the negative logarithms of the *P*-values from the models fitted in GWAS are plotted against their expected value under the null hypothesis of no association with the trait. Because most of the SNPs tested are probably not associated with the trait, the majority of the points in the QQ-plot should lie on the diagonal line. Deviations from this line suggest the presence of spurious associations due to population structure and familial relatedness, and that the GWAS model does not sufficiently account for these spurious associations. It is expected that the SNPs on the upper right section of the graph deviate from the diagonal. These SNPs are most likely associated with the trait under study. By default, the QQ-plots in FarmCPU show only a subset of the larger *P*-values (i.e., less significant *P*-values) to reduce the file size of the graph.

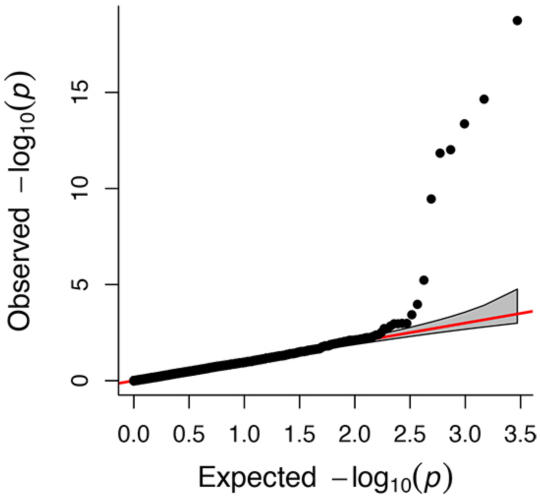


**Figure 5.2.** Quantile-Quantile (QQ) –plot of *P*-values. The Y-axis is the observed negative base 10 logarithm of the *P*-values, and the X-axis is the expected observed negative base 10 logarithm of the *P*-values under the assumption that the *P*-values follow a uniform [0,1] distribution. The dotted lines show the 95% confidence interval for the QQ-plot under the null hypothesis of no association between the SNP and the trait.

## 5.3 *Association Table*

The GWAS result table provides a detailed summary of appropriate GWAS results. The rows display the results for each SNP above the user-specified minor allele frequency threshold.

**Table 5.3**. GWAS results for all SNPs that were analyzed.

| SNP | Chromosome | Position | P.value | maf | effect |
|---|---|---|---|---|---|
| PZA02962.14 | 1 | 3205262 | 0.002457044 | 0.096085409 | 0.836330408 |
| PZA02032.1 | 1 | 4490461 | 1.06E-05 | 0.174377224 | 0.900445081 |
| PZA03520.1 | 1 | 10067544 | 0.003947552 | 0.307829181 | 0.600141218 |
| PZA03520.2 | 1 | 10067604 | 0.0012993 | 0.140569395 | 0.905302504 |
| PZA03521.1 | 1 | 10068726 | 0.002491113 | 0.192170819 | 0.718196009 |

This table provides the SNP id, Chromosome, phsical position, *P*-value, Minor allele frequency (maf), and SNP effect.

## *5.4 Effect Table of User-provide Covariates*

The Effect table provides the estimated effect of user-provided covariates, such as PCs, sex.

**Table 5.4**. Effect of user-provide covariates.

| CV | Effect |
|---|---|
| 1 | 0.033248735 |
| 2 | 0.357262046 |
| 3 | -0.184175048 |

# 6 Tutorials

FarmCPU function in these tutorials adds more input based on the function that used in the second section. It is assumed that FarmCPU package and required libraries have been installed before running these tutorials. We use the same genotypic data, phenotypic data and genotypic map data in the second section.

*Notice*: Before running the codes in following tutorials, make sure the correct paths to the required data. Please note that two backward slashes ("\\") are necessary when you are using windows system and it is one slash ("/") in mac OS and Linux systems.

## *6.1 User-provide Covariates*

The user needs to provide one data set (myCV) and one input parameter (CV=myCV). User needs to organize the covariates data using the format of 3.4. **We strongly recommended users to calculate PCs using a third software like GAPIT, TASSEL, PLINK, or GCTA and then add top PCs (3-5 PCs) as covariates in FarmCPU to reduce false positives caused by non-genetic effect such as environmental effect.** The analysis can be performed by typing following command lines:

```
#Step 1: Set data directory and import files
myY  <- read.table("mdp_traits_validation.txt", head = TRUE)
myGM <- read.table("mdp_SNP_information.txt", head = TRUE)
myGD <- read.big.matrix("mdp_numeric.txt", type="char", sep="\t", head = TRUE)
myCV <- read.table("Copy of Q_First_Three_Principal_Components.txt", head = TRUE)

#Step 2: Run FarmCPU
myFarmCPU <- FarmCPU(
                    Y=myY[,c(1,2)],
                    GD=myGD,
                    GM=myGM,
                    CV=myCV
                    )
```

## *6.2 Optimizing bin.size and bin.selection*

In this scenario, users can optimize the combinations of two parameters: bin.size and bin.selection. Set "method.bin" parameter to "optimum" can start the optimization functions. "bin.size" and "bin.selection" are related to the LD (Linkage Disequilibrium) distance. We recommend users to use the default set of bin.size and bin.selection. The analysis can be performed by typing following command lines:

```
#Step 1: Set data directory and import files
myY  <- read.table("mdp_traits_validation.txt", head = TRUE)
myGM <- read.table("mdp_SNP_information.txt", head = TRUE)
myGD <- read.big.matrix("mdp_numeric.txt", type="char", sep="\t", head = TRUE)

#Step 2: Run FarmCPU
myFarmCPU <- FarmCPU(
                    Y=myY[,c(1,2)],
                    GD=myGD,
                    GM=myGM,
                    method.bin="optimum",
                    bin.size=c(5e5,5e6,5e7), #default set of bin.size
                    bin.selection=seq(10,100,10) #default set of bin.selection
                    )
```

## 6.3 Perform multi Iterations

According to our study, more iterations boost the Power/FDR (False Discovery Rate). “maxLoop” can used to set the number of iterations in FarmCPU. “maxLoop=1” switches FarmCPU to t test, “maxLoop=2” gives the fastest speed of FarmCPU and “maxLoop=10” or a larger number that makes FarmCPU stopped automatically that gives the best Power/FDR. We use “maxLoop=10” as default. The analysis can be performed by typing following command lines:

```
#Step 1: Set data directory and import files
myY  <- read.table("mdp_traits_validation.txt", head = TRUE)
myGM <- read.table("mdp_SNP_information.txt", head = TRUE)
myGD <- read.big.matrix("mdp_numeric.txt", type="char", sep="\t", head = TRUE)

#Step 2: Run FarmCPU
myFarmCPU <- FarmCPU(
                    Y=myY[,c(1,2)],
                    GD=myGD,
                    GM=myGM,
                    maxLoop=10
                    )
```

## 6.4 Add prior knowledge

Known QTNs can be added to the FarmCPU model by using “Prior=prior”. The details of prior data format see 3.5. The analysis can be performed by typing following command lines:

```
#Step 1: Set data directory and import files
myY  <- read.table("mdp_traits_validation.txt", head = TRUE)
myGM <- read.table("mdp_SNP_information.txt", head = TRUE)
myGD <- read.big.matrix("mdp_numeric.txt", type="char", sep="\t", head = TRUE)
myPrior  <- read.table("prior.txt", head = TRUE)

#Step 2: Run FarmCPU
myFarmCPU <- FarmCPU(
                    Y=myY[,c(1,2)],
                    GD=myGD,
                    GM=myGM,
                    Prior=myPrior
                    )
```

## 6.5 The p-value threshold used to judge whether FarmCPU is suitable for the phenotype

If there is no significant SNPs pass the p-value threshold, FarmCPU will stop. The default p-value threshold is bonferroni-corrected threshold with 0.01. As bonferroni-corrected threshold is overly strict when the LD among genotypic markers is large, users can change the threshold using the parameter “p.threshold”. As use a looser threshold may make a wrong judgment, users should change the parameter very carefully! The analysis can be performed by typing following command lines:

```
#Step 1: Set data directory and import files
myY  <- read.table("mdp_traits_validation.txt", head = TRUE)
myGM <- read.table("mdp_SNP_information.txt", head = TRUE)
myGD <- read.big.matrix("mdp_numeric.txt", type="char", sep="\t", head = TRUE)

#Step 2: Run FarmCPU
myFarmCPU <- FarmCPU(
                    Y=myY[,c(1,2)],
                    GD=myGD,
                    GM=myGM,
                    p.threshold=0.05/nrow(myGM)
                    )
```

FarmCPU also provides a function to give a recommended **p.threshold** value. In this function, the phenotypes are permuted to break the relationship with the genotypes. The experiment is replicated for a number of times. A vector of minimum p value of each experiment is outputted and the 95% quantile value of the vector is recommended for **p.threshold** in FarmCPU model. The analysis can be performed by typing following command lines:

```
#Step 1: Set data directory and import files
myY  <- read.table("mdp_traits_validation.txt", head = TRUE)
myGM <- read.table("mdp_SNP_information.txt", head = TRUE)
myGD <- read.big.matrix("mdp_numeric.txt", type="char", sep="\t", head = TRUE)

#Step 2: Run FarmCPU.P.Threshold function
          FarmCPU.P.Threshold(
           Y=myY[,c(1,2)], #only two columns allowed, the first column is taxa name and the second is phenotype value
                    GD=myGD,
                    GM=myGM,
                    trait="trait_name", #name of the trait, only used for the output file name
                    theRep=30 #number of permutation times
                    )
```

## 6.6 *Filter SNPs by Minor Allele Frequency*

“MAF.calculate=TRUE” can be used to start this function and “maf.threshold=0.05” can be used to remove the SNPs which have MAF under 0.05, “prior” can not be used with “maf.threshold” together. The analysis can be performed by typing following command lines:

```
#Step 1: Set data directory and import files
myY  <- read.table("mdp_traits_validation.txt", head = TRUE)
myGM <- read.table("mdp_SNP_information.txt", head = TRUE)
myGD <- read.big.matrix("mdp_numeric.txt", type="char", sep="\t", head = TRUE)

#Step 2: Run FarmCPU
myFarmCPU <- FarmCPU(
                    Y=myY[,c(1,2)],
                    GD=myGD,
                    GM=myGM,
                    MAF.calculate=TRUE,
                    maf.threshold=0.05
                    )
```

# 7 Appendix

## 7.1 Frequently Asked Questions

### 1. What do I do if I get frustrated?

A: The FarmCPU team makes the effort to provide suggestions to any errors that users might run into. If you experience some problems, send us an email. The email should includes following information: (1) Copy and paste the error message from R environment; (2) Your source code; (3) The datasets that we can repeat the error. The email can be sent to Xiaolei Liu (xll19870827@hotmail.com) or Zhiwu Zhang (Zhiwu.Zhang@WSU.edu).

### 2. Warning: unable to move temporary installation 'XYZ' to 'ZYX'

A: If you get this warning while installing or updating packages, most likely some other Windows program is accessing the newly installed files and prevents R to move them to the correct place. Often simply trying to install the package a second time works. Otherwise try to pause your antivirus programs or search indexing (like Google Desktop Search) while installing the packages. (This answer is from “http://cran.r-project.org/web/packages/gMCP/INSTALL”)

### 3. How to make further optimization of FarmCPU parameters in your specific study?

A: Some parameters may be useful to improve your results. The parameters include: MAF.calculate, maf.threshold, bin.size, bin.selection, p.threshold and bound.

(1) Filter SNPs by minor allele frequency: MAF.calculate and maf.threshold (see 6.6);

(2) Split whole genome into bins: bin.size and bin.selection;

(3) The threshold of possible QTNs selected into FarmCPU model: p.threshold (default is 1% bonferroni test threshold, users can change the threshold due to your specific data, lower the threshold especially when you find the results are still a little bit inflated, see 6.5);

(4) Number of possible QTNs selected for optimization: bound (see Gallery of FarmCPU input parameters).

### 4. How to cite FarmCPU?

A: Liu X, Huang M, Fan B, Buckler ES, Zhang Z (2016) Iterative Usage of Fixed and Random Effect Models for Powerful and Efficient Genome-Wide Association Studies. PLoS Genet 12(2): e1005767. doi:10.1371/journal.pgen.1005767

## 7.2 FarmCPU Biography

| Date | Version | Event |
|---:|:---:|---|
| Jan 1, 2016 | 1.0 | First public release with following method implemented:<br>Fixed and random model Circulating Probability Unification (FarmCPU) |
| May 4, 2016 | 1.01 | Add SNP effect in Association Table, add a new effect table for user-provide covariates, and debugged an error when using user-provide covariates |
| Dec 21, 2016 | 1.02 | Debug FarmCPU.Burger function; Add a new parameter named "QTN.threshold" which can be used as a threshold for selecting pseudo QTNs |

# REFERENCES

1. Lipka, A. E. *et al.* GAPIT: genome association and prediction integrated tool. *Bioinformatics* **28,** 2397–9 (2012).

2. Bradbury, P. J. *et al.* TASSEL: software for association mapping of complex traits in diverse samples. *Bioinformatics* **23,** 2633–2635 (2007).

3. Price, A. L. *et al.* Principal components analysis corrects for stratification in genome-wide association studies. *Nat. Genet.* **38,** 904–9 (2006).