User Manual for

BLINK

Bayesian-information and Linkage-disequilibrium Iteratively Nested Keyway

(Version 1.01)

Last updated on May 14, 2016

Zhiwu Zhang Laboratory

For Statistical Genomics

ZZLab.Net

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

Support documents: Extensive support documents, including this user manual, source code, demonstration scripts, data, and results, are available at BLINK website at Zhiwu Zhang Laboratory: http://zzlab.net/blink

Questions and comments: Users and developers are recommended to post questions and comments at GAPIT forum: https://groups.google.com/forum/#!forum/blink.
Answers from other users and developers are appreciated. The BLINK team members will periodically go through these questions and comments and address them accordingly.

The BLINK project is supported by the Agricultural Research Center at Washington State University, and Washington Grain Commission

Contents

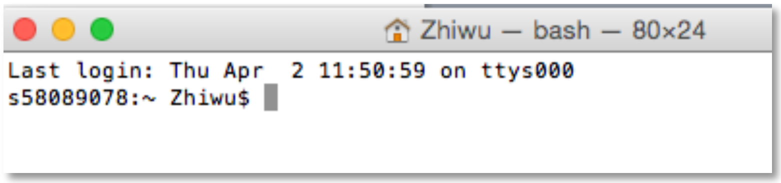
1 Introduction

The performance of computing tools for genome-wide association studies (GWAS) are measured by their computing speed, memory requirements, and statistical power[1]. These three factors are determined by the statistical methods a tool implemented and how these methods are engineered to make full use of computer hardware resources. We developed a computing tool named BLINK that implements a new statistical method. BLINK effectively controls false positives caused by population structure and unequal relatedness among individuals and improves statistical power - even when compared to mixed linear model methods. The BLINK method requires much less complex computing time, nearly matching the linear computing time complexity of the general linear model. BLINK was written in C computer language to maximize the capability of direct electronic circuit operations, including binary formatting of genotype input files and bit operations for matrix manipulations. To further increase computing speed, BLINK was developed with parallel computational capacity, so that computing times decrease linearly with the number of central processing units. Furthermore, the parallel components are dissected small enough so that graphic processing units are also able to perform parallel computations. To solve the memory footprint bottleneck, BLINK allows users to directly control memory usage when big data are analyzed on computers with limited memory. That is, users have the option to trade computing time for less memory usage. Based on these features above, BLINK makes analyses of large and complex datasets feasible without supercomputers.

2 Getting started

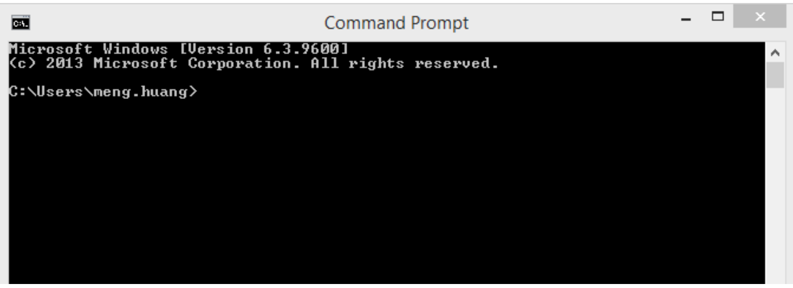
2.1 Open command line window

BLINK use Command-Line Interface (CLI). In Mac, the application is called Terminal. From Applications window, click Utilities and then Terminal.



```
Last login: Thu Apr  2 11:50:59 on ttys000
s58089078:~ Zhiwu$
```

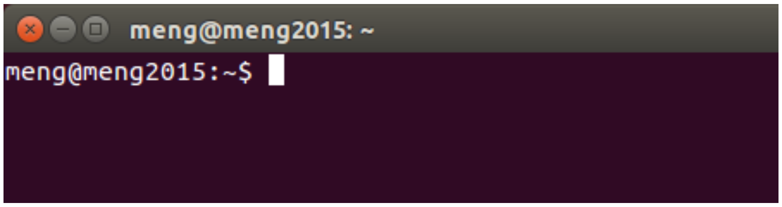
In Windows, the application is called Command Prompt. From search window, input “cmd” and then choose it from results.



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\meng.huang>
```

In Linux (Ubuntu), the application is also called Terminal. From search window, input “terminal” and then choose it from results.



```
meng@meng2015:~$
```

2.2 Download BLINK

The BLINK executable program (BLINK) can be download at http://ZZLab.net/blink. Create a folder on your hard disk, for example, myBLINK and save the BLINK executable program in the folder.

2.3 Download input files

Go to http://ZZLab.net/blink and download the demo data, then copy all the files including data and BLINK executable program to the same folder (e.g. myBLINK).

NOTE: Although most of the file have the same format as GAPIT[2] and TASSEL[3], differences do exist.

2.4 ***Run BLINK***

Users need to specify the pathway and names of BLINK executable file and input files. A convenient way is to change current pathway to the one containing these files. This can be done with this command in the Terminal:

```
cd /users/Zhiwu/myBLINK
```

To perform GWAS between phenotype and one of the genotype formats, for example the compress format, type the following command:

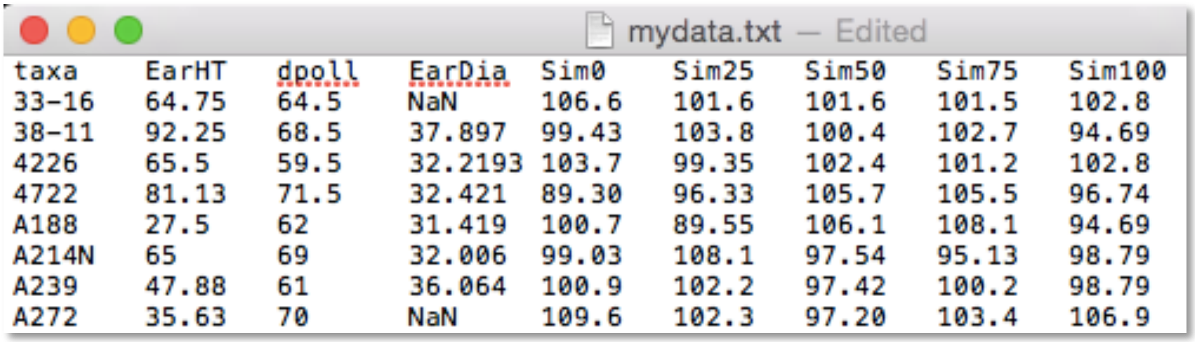
```
blink --gwas --file myData --binary
```

There are five input files involved in this analyses: myData.pre, myData.pos, myData.val, myData.map, and myData.txt. UNIX operating system (e.g. Mac and Ubuntu) may require adding “./” in front of these commmand lines to specify the current directory.

3 Phenotype file

3.1 Format

Phenotype file is coded as text file with extension of "txt". The file name must be the same as genotype file(s) so they can be analyzed together. BLINK supports multiple traits. The first column is reserved for individual name. Each trait occupies one column. The first row is reserved as the header of each column. The following figure demonstrates the first eight rows of the phenotype file from the demonstration dataset.



taxa	EarHT	dpoll	EarDia	Sim0	Sim25	Sim50	Sim75	Sim100
33-16	64.75	64.5	NaN	106.6	101.6	101.6	101.5	102.8
38-11	92.25	68.5	37.897	99.43	103.8	100.4	102.7	94.69
4226	65.5	59.5	32.2193	103.7	99.35	102.4	101.2	102.8
4722	81.13	71.5	32.421	89.30	96.33	105.7	105.5	96.74
A188	27.5	62	31.419	100.7	89.55	106.1	108.1	94.69
A214N	65	69	32.006	99.03	108.1	97.54	95.13	98.79
A239	47.88	61	36.064	100.9	102.2	97.42	100.2	98.79
A272	35.63	70	NaN	109.6	102.3	97.20	103.4	106.9

The individuals have to be in the same order as the genotype data.

3.2 Missing values

Missing data are allowed in phenotype data. Missing data in genotype can be any character (such as N, NA, or NaN) except numerical number and decimal. But missing data in phenotype should only be "NaN". Traits are analyzed independently. None missing values of each trait are matched with genotype for each trait.

NOTE: When the trait has missing value and do GWAS

4 Genotype file format and conversion

4.1 Genotype formats

BLINK currently supports five types of genotype formats: compress, numeric, hapmap, vcf and plink. The files must have the same name for each format except the extension names. The compress format is the working format to perform analyses. All the other formats are converted to compress format before the analyses. The compress format has four files. Their extensions are pre, pos, val and map. "pre" and "pos" are coded as binary files containing genotype data. The other two are coded as text. "val" store the inter product of each marker. "map" is the map information of markers, including marker name, chromosome and base pair position. These four files must be used together for the compress format.

The numeric format has two file extensions: dat and map. The file with extension "dat" contains genotype data. The other file has extension of "map", which is the same as the map file of the compress format. These two files must be used together for the compress format.

The plink binary format also has three file extensions: .bed, .bim, and .fam. Please find detail of plink format at http://pngu.mgh.harvard.edu/~purcell/plink/data.shtml#bed

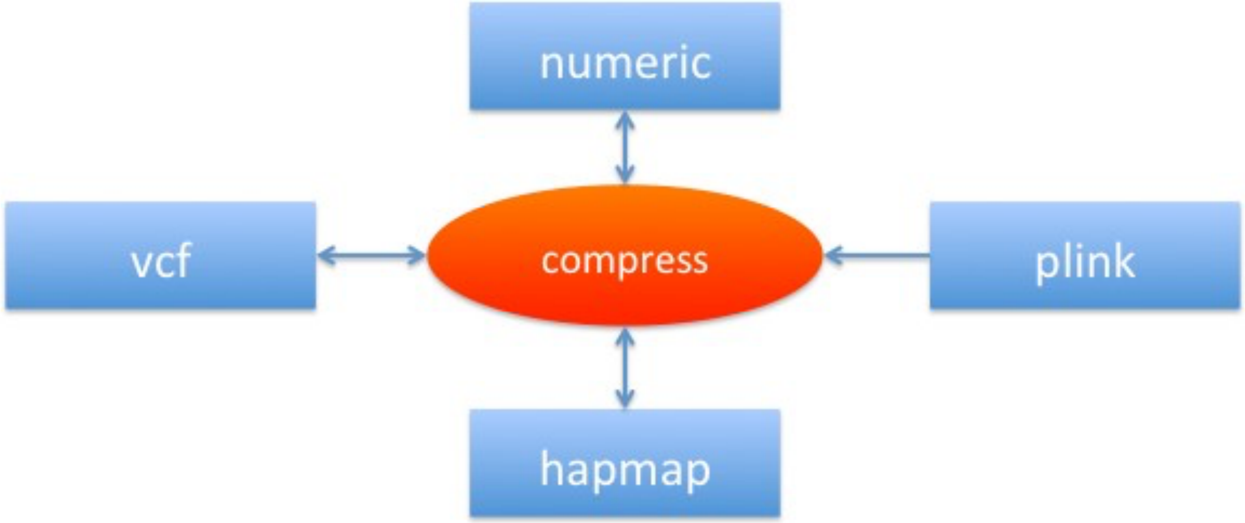
The hapmap and vcf file only have one file each. They are both coded as text. The genotype in hapmap format has extension of "hmp". The genotype in vcf format has extension of "vcf".

The relationship between genotype formats and file extensions are summarized in following table:

Extension	Code	compress	numeric	hapmap	vcf	plink
pre	binary	x				
pos	binary	x				
val	text	x				
dat	text		x			
map	text	x	x			
hmp	text			x		
vcf/vcf.gz	text				x	
bed	binary					x
bim	binary					x
fam	binary					x

4.2 ***Format conversion***

BLINK uses compress genotype format by default as working data format to perform analyses efficiently. BLINK supports other genotype formats by converting them to compress format first. BLINK also supports converting compress format to other genotype formats, including numeric, hapmap, and VCF. See diagram below:



BLINK uses --recode option to convert compress format to other formats except PLINK format:

```
blink --file myData --recode --out newdata --numeric
blink --file myData --recode --out newdata --hapmap
blink --file myData --recode --out newdata --vcf
```

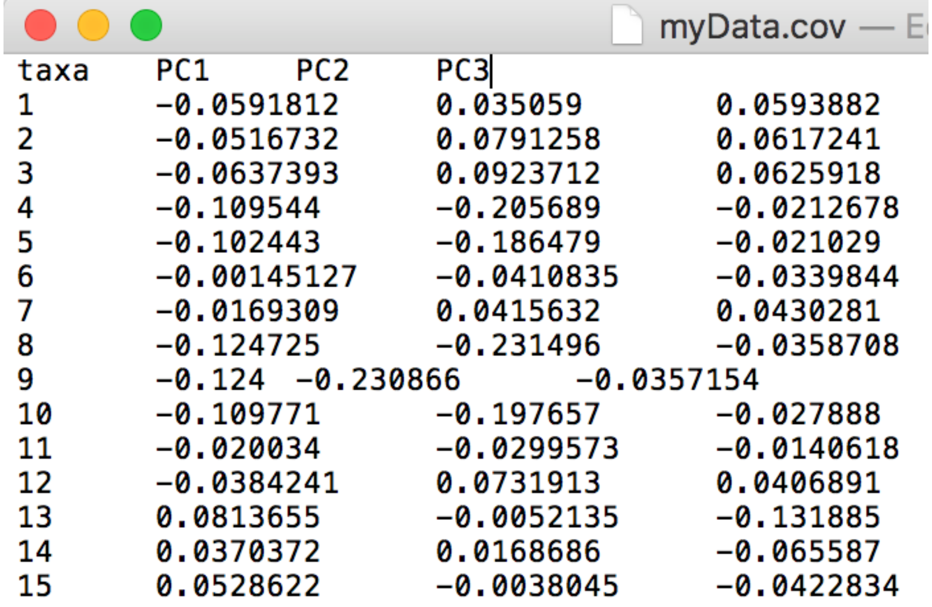
The option of "--out" is used to specify the name of output of the converted genotype format. When it is omitted, the default uses the name of input.
NOTE: When we use recode function, all the four compressed format files and phenotype file should be existed in the same folder and with same name (e.g. myData.pre, myData.pos, myData.val, myData.map, myData.txt). All the missing information will be filled with "NA".

BLINK also can convert other formats (numeric, HapMap, VCF and PLINK) to compress format without performing additional analyses by using "--compress" option:

```
blink --file myData --compress --numeric
blink --file myData --compress --hapmap
blink --file myData --compress --vcf
blink --file myData --compress --plink
```

4.3 Covariance format

The covariance will be saved column by column with title and ID into the text file. Different column means different covariance and different row means different individuals. When you want to add covariance into model, just keep its file name same as genotype files and with the extension “.cov” (e.g. myData.cov), then put them into same folder and do GWAS analysis.



```
taxa    PC1       PC2      PC3
1       -0.0591812         0.035059          0.0593882
2       -0.0516732         0.0791258         0.0617241
3       -0.0637393         0.0923712         0.0625918
4       -0.109544          -0.205689         -0.0212678
5       -0.102443          -0.186479         -0.021029
6       -0.00145127        -0.0410835        -0.0339844
7       -0.0169309         0.0415632         0.0430281
8       -0.124725          -0.231496         -0.0358708
9       -0.124  -0.230866       -0.0357154
10      -0.109771          -0.197657         -0.027888
11      -0.020034          -0.0299573        -0.0140618
12      -0.0384241         0.0731913         0.0406891
13      0.0813655          -0.0052135        -0.131885
14      0.0370372          0.0168686         -0.065587
15      0.0528622          -0.0038045        -0.0422834
```

The PCA could also be calculated by using “--pca”. BLINK will output top 3 PCs under default setting, but user could change the number of output PCs, e.g. “--pca 6” could output top 6 PCs. The output PCA files have the extension “.eigenval” and “.eigenvec”, which include eigen value and eigen vector respectively. For PCA, BLINK only accept PLINK binary format as input file.

5 GWAS

Both phenotype and genotype files are required to perform GWAS. These files must share a common name with different extensions specified by phenotype and different genotype formats. Analyses of GWAS is specified with “--gwas” option.

5.1 Working with different genotype formats

To perform GWAS with BLINK on one of the genotype formats, type the one of the corresponding file formats:

```
blink --gwas --file myData --binary
blink --gwas --file myData --numeric
blink --gwas --file myData --hapmap
blink --gwas --file myData --vcf
blink --gwas --file myData --plink
```

The GWAS result contains map information of the marker and corresponding p values. The output file is named by the trait name followed by “_GWAS_result.txt” in format of ‘TraitName_GWAS_result.txt’. The file can be directly used by third-party software (e.g. GAPIT in R) for visualizations, such as Manhattan and QQ plots.

5.2 Changing output file name

Users have the need to change output file name in some cases. BLINK provides an option to fit the need. The default output file name can be changed by using “--out” option as following:

```
blink --file myData --compress --out newData
```

6 Genomic prediction

Under development

7 Advanced operations

BLINK provide more options for analyses with special needs, such as analyses on particular trait, memory saving and customized optimization.

7.1 Memory saving

Define the memory usage by control the number of markers in one cycle (default value is 1000). --cycle_size 2000

7.2 Parallel computation

Choose parallel or not. blink --file myData --gwas --parallel 1
This option will let BLINK switch to parallel computing in CPU device and the number of threads is specified by --cycle_size.

7.3 Specifying a trait

BLINK only analyze the first trait by default. A specific trait, for example the third trait, can be analyzed by option "--trait 3" as following:

```
blink --file myData --binary --gwas --trait 3
```

When "--trait 0" is specified, BLINK will automatically analyses on all the available traits.

7.4 Optimization

1. Define the size of bin divided in whole genome, and the unit is 1+e6 bp. The first number is the length of bin_size array, the numbers start from second one is the length of bin size.
--bin_size 3 50 5 0.5
2. Define the chosen number of top SNPs coming from each bin. The first number is the length of bin_selection array, the numbers start from second one is the value of bin selection. --bin_selection 3 10 20 30
3. Define the max number of iterations. --max_loop 5
4. Add prior QTN. The first number is the total number of prior QTN, the numbers start from second one are the order of prior QTN in all the SNPs in .map file.
--prior 3 12345 54321 43215

7.5 Run BLINK from R

As a command, BLINK can be run from R by using system function. The following R code demonstrates the usage of GAPIT demonstration data, simulation of phenotype, analyses with BLINK and visualization.

```
#Import data
setwd("/Users/Zhiwu/myGAPIT")
myGD <- read.table("mdp_numeric.txt", head = TRUE)
myGM <- read.table("mdp_SNP_information.txt", head = TRUE)

#Import library
#source("http://www.bioconductor.org/biocLite.R")
#biocLite("multtest")
#install.packages("gplots")
#install.packages("scatterplot3d")

library('MASS') # required for ginv
library(multtest)
library(gplots)
library(compiler) #required for cmpfun
library("scatterplot3d")
source("http://www.zzlab.net/GAPIT/emma.txt")
source("http://www.zzlab.net/GAPIT/gapit_functions.txt")

#Simulating phenotype
set.seed(99163)
myPheno=GAPIT.Phenotype.Simulation(GD=myGD,h2=.75,NQTN=20,QTNDist="geometry",
 effectunit=.92)
myY=myPheno$Y
QTN.position=myPheno$QTN.position

#Create BLINK input data
setwd("/Users/Zhiwu/myBLINK/Simulation")
GD=t(myGD[,-1])
write.table(GD,file="myData.dat",quote=F,sep="\t",col.name=F,row.name=F)
write.table(myGM,file="myData.map",quote=F,sep="\t",col.name=T,row.name=F)
write.table(myY,file="myData.txt",quote=F,sep="\t",col.name=T,row.name=F)

#Run BLINK
system("/Users/Zhiwu/myBLINK/blink    --file    /Users/Zhiwu/myBLINK/Simulation
 /myData --out /Users/Zhiwu/temp/myData")

#Manhattan and QQ plots
GMP <- read.delim("myData_GWAS_result.txt", head = T)
```

```
GMP=GMP[,c(2,3,5)]
GAPIT.Manhattan(GI.MP = GMP, name.of.trait = "Trait",plot.type =
 "Genomewise",DPP=50000,cutOff=0.01,band=2,seqQTN=QTN.position)
GAPIT.QQ(P.values=GMP[,3], plot.type = "log_P_values", name.of.trait =
 "Trait",DPP=50000)
```

8 Appendix

8.1 Tutorial Data sets

The data set contains 26 files and can be downloaded at: https://zzlab.net/GAPIT/GAPIT_Tutorial_Data.zip

8.2 Frequently Asked Questions

1. Why do I use BLINK?

A: BLINK is designed to make you more successful for finding genes of your interest, such as the ones lead to cure of cancers, or reduction of using pesticides. It also aims to reduce computing time and memory usage so that big can be analyzed.

2. How do I cite BLINK?

A: We are in the process for your convenience of citation. Please cite: "https://academic.oup.com/gigascience/article/8/2/giy154/5238723".

8.3 BLINK Biography

Date	Version	Event
February 11, 2015	0.01	First public release and start beta testing
April 3, 2015	0.02	Multiple traits, missing phenotypes, and genotype conversion
March 18, 2024	0.03	add PCA function

9 References

1. Zhang, Z., Buckler, E. S., Casstevens, T. M. & Bradbury, P. J. Software engineering the mixed model for genome-wide association studies on large samples. *Br. Bioinform* **10,** 664–675 (2009).

2. Lipka, A. E. *et al.* GAPIT: genome association and prediction integrated tool. *Bioinformatics* **28,** 2397–2399 (2012).

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