GLASCOW

Frédéric Farnir, Tom Druet, François Guillaume

July 23, 2013

Contents

1 Requirements **1**

2 Keys for the parameter file **4**

3 Options **7**

4 Misc **9**

5 Citation **10**

Introduction

GLASCOW performs genome-wide association studies using Generalized Linear Mixed Models (GLMM) and a score test as described in Zhang et al. Association is performed between factors (SNPs or haplotypes) and a phenotype (with different type of distributions: normal, binomial, counts). The method relies on two steps. In the first one, a GLMM without the factor of interest is solved (including estimation of variance components and of fixed and random effects solutions) and residuals are computed. In the second step, the residuals are used to test significance of association between the factor and the phenotype at each position along the genome (or the selected data).

1 Requirements

1.1 Input files

GLASCOW needs a phenotype file, a genotype file and a parameter file to specify the parameters of the run. By default, in all these files all fields are separated by blanks (space or tabulation).

The phenotype file This file provide the phenotypic observations along with eventual fixed effects values. Each observation corresponds to one line in the file, in the following format:

```
<Identifier> [<Fixed effect 1> [<Fixed effect2>...]] <Trait>
```

<**Identifier**> is an integer number, from 1 to N. Note however that monozygotic twins (i.e. individuals with the same identical genotypes) should receive the same identifier. This means that N is equal to the number of records minus the number of sets of monozygotic animals.

<**Fixed effect** i> contains the level of the corresponding fixed effect i, a number between 1 and f_i where f_i is the total number of levels for fixed effect i. When <Fixed effect i> is a covariate, the value can be any real.

<**trait**> is the value of the trait: it can be 0 or 1 for a binary trait, an integer number between 0 and $+\infty$ for count traits, and any real value for a normally distributed trait.

Example file, with two fixed effects and one binary traits

```
1 1 0.25 0
2 2 0.50 0
3 1 0.75 1
4 1 0.25 0
```

The genotype file Two kind of genotype files are accepted either a genotype file (e.g. derived from plink ped file), or a phase file (as the one obtain with PHASEBOOK).

genotype file The genotype file provide the genomic information. This data is provided on one line for each distinct (from a genetical point of view) individual. The format is:

```
<Identifier> [<Geno effect 1a> <Geno effect 1b> [<Geno effect 2a> <Geno effect 2b>...
```

The <identifier> field is an integer number, from 1 to N, corresponding to the N value explained for the phenotypes file. The genomic effects are provided through sets of 2 values, the number of sets corresponding to the number of positions to be examined (for example to cover a complete chromosome or even to cover the whole genome).

<Geno effect i> should take a value between 1 and n, where n is the possible number of values for that position (2 for SNP, > 2 for microsatellites or for haplotype groups). No missing genotypes are allowed.

Example with three positions

```
1 1 2 1 2 8 2
2 4 2 3 5 2 4
3 3 5 3 5 3 5
4 3 6 3 6 6 1
```

Phase file The phase file provide the genomic information. This data provide two line per individual, one line for each distinct phase. . The format is:

```
<Identifier> phase [<Geno effect 1a>] [<Geno effect 2a>...]
<Identifier> phase [<Geno effect 1b>] [<Geno effect 2b>...]
```

The <identifier> field is an integer number, from 1 to N, corresponding to the N value explained for the phenotypes file. The genomic effects are provided through sets of 2 values (one per line), the number of sets corresponding to the number of positions to be examined (for example to cover a complete chromosome or even to cover the whole genome).

<Geno effect i> should take a value between 1 and n, where n is the possible number of values for that position (2 for SNP, > 2 for microsatellites or for haplotype groups). No missing genotypes are allowed.

Note : if you previously used PHASEBOOK to obtain hidden state data, then you just have to give PHASEBOOK output files as input. Note also, that as PHASEBOOK work one chromosome at a time, GLASCOW allow to read several different genotypes files (cf above).

Example with three positions

```
1 1 1 1 8
1 2 2 2 2
2 1 4 3 2
2 2 2 5 4
3 1 3 3 3
3 2 5 5 5
```

The parameters file This file provides details on the datasets, the tested model and the estimation procedure. The file can be created using a text editor and specifies the necessary instructions in the following format:

```
<KEYWORD> [<arguments-list>]
```

where :

<**KEYWORD**> is one of the keywords listed below (see 2 "Keys index")

<**arguments-list**> is a list of arguments depending on the used option.

Keys are expected to be 5 letters long and written with capital letters. Any misspelled or badly formatted key will force GLASCOW to stop.

This file can contain commentaries lines, the latter should start by a "#" symbol. Empty lines are also allowed.

1.2 Running the program

The program is run from a terminal using GLASCOW folowed by the name of the parameter file.

```
./GLASCOW paramfile
```

1.3 Results

At the end of computation, GLASCOW will output several results files whose name will be constructed according to the analysis name (<aname>).

<**aname**>**.out** : This file contains for each tested position, parameters (a and b) of the gamma distribution estimated using permutations, the score and it's associated p-value (estimated from the gamma distribution), the local rank p-value (obtained through local permutations) and the p-value estimated from the gamma distribution corrected for multiple testing (corrected for the number of test performed in the analysis and accounting for possible correlation between successive tests).

Note: In some particular cases, the subroutine estimating p-values from the gamma distribution gives incorrect results. Correct p-values can be obtained for instance with R with the following command: pgamma(score,shape=a,scale=b,lower.tail=F)

Example

```
47005    3.313473    1.569912    36.483599    0.4750E-07    0.1000E-03    0.8400E-02
47006    3.400174    1.508535    31.257483    0.5458E-06    0.1000E-03    0.4260E-01
47007    3.499887    1.493272    31.947637    0.3663E-06    0.1000E-03    0.3220E-01
47008    3.182201    1.671054    21.548003    0.3310E-03    0.9000E-03    0.9454E+00
```

2 Keys for the parameter file

In the following list of keys, default arguments (when any) will be underlined. Bold fonts will be use for mandatory keys.

ANAME <NAME>

This option assign a name to the analysis. The generated files will use that name (referenced as <aname> hereafter) as file name. This option is mandatory to avoid unwillingly overwriting previous analysis results.

FIXED <NAME> <TYPE> <POSITION>

This option specifies fixed effects to be accounted for in the analysis.

The <name> argument provide a name for the fixed effect, and can be any alphanumerical string of length ≤ 20 characters. The argument <type> provides the number of levels associated to the fixed effect (an integer number ≥ 1). When set to 0, it means that the corresponding fixed effect is a covariate rather than a factor. The <position> argument specifies the field corresponding to the fixed effect value in the record (and must therefore be a positive integer).

If several fixed effects are used in the model, each one must have it's own specification line.

GFILE <FILENAME>

This key provide the name of the genomic data file.

HAPLO <NB HAPLOS> <NB POS> <POS1>

This option describe the haplotypes effect. <nb haplo> arguments provide the (maximum) number of haplotype effect at each position. <nb pos> is the number of positions along the tested genomic region. <pos1> is the position in the file of the first marker.

PERMS < NB_PERMUTATIONS | <u>1000</u> >

This option specifies the number of permutations to be performed to assess the p-value of the obtained scores.

PFILE <FILENAME>

This keyword provide the name of the phenotypes file.

PHENO <NAME> <TYPE> <POSITION>

This option describe the studied phenotype. The <name> argument provide a name for the phenotype, and can be any alphanumerical string of length ≤ 20 characters. The <type> argument is a strictly positive integer value indicating the type of phenotypes to be dealt with: a value of 1 indicates a 'NORMAL' phenotype (such as weight, or height), a value of 2 indicates a 'BINARY' phenotype (such as disease status (healthy or ill)), a value > 2 indicates a 'COUNT' phenotype (such as the number of seizures of a disease in a given period of time). The <position> argument specifies the number of the field corresponding to the phenotype value in the record,(and must therefore be a positive integer).

POLYG <NUMBER INDIVIDUALS> <POSITION>

This option describe the polygenic effect. <number individuals> provide the number of individual in the pedigree and <position> indicate the field number where the individual id should be found in the data file.

REMLI <ITERATIONS| 1000 >

The REML estimation iterates until either successive solution and parameter vectors are sufficiently close, indicating convergence, or when the maximum number of iterations is exceeded. This option allows specifying the maximum number of iterations to be performed: if this number is reached, it is considered that, although REML estimation stops, no convergence has been achieved and the current (possibly invalid) solutions are reported.

REMLS < POLYG|RESIDUAL > <VALUE|1.0

This instruction can provide starting values for variance components (σ^2_P, σ^2_E). By default, variances are set to 1.0. Using this option it is possible to change the initial value of some (or all) variance components.

REMLT <THRESHOLD|$\underline{110^{-10}}$>

The REML estimation iterates until either successive parameter (variances) vectors are sufficiently close, indicating convergence, or when the maximum number of iterations is exceeded. This option specifies the threshold below which convergence is reached. The convergence criterion is equal to the sum of the squared differences between successive variances divided by the sum of squared variances.

RFILE <MODE> <FILE>

This option specifies the correlation structures linked to the random effects.

The <mode> field is used to mention whether the provided data is the correlation matrix (<mode> = 1) or the inverse of the correlation matrix (<mode> = -1).

Next, the <file> field is the name of the file containing the upper or lower triangle of the (inverse) correlation matrix in the format: <line> <column> <value>.

SCORE <POSITION_1> <POSITION_N>

The score statistics will be computed for the specified positions (from position_1 to position_ n included).

SHOWP < NB_POSITIONS | $\underline{1000}$ >

This option avoid too verbose reporting of the run: a message will be issued every <nb_position> positions.

SLICE <SIZE|$\underline{10000}$>

Since GWAS analyses might possibly involve overwhelming number of positions, it could be easier to work using successive "slices" rather than the whole set of positions. To that end, this option can be used to specify the size (number of positions) of the successive slices. For example, if 1.000.000 positions have to be scanned, it can be easier to load 10 successive slices of 100.000 positions each, which could be done using the instruction 'SLICE 100000'.

3 Options

Additional features can be enable through command line arguments. The latter should follow the parameter file name.

3.1 -dbg

This option will write additionnal informations on screen for debugging purpose.

3.2 -exportcorr

This option will write files containing the lower triangular terms of the correlation structure linked to the random effects and its inverse, these can in turn be used in conjunction with the RFILE keyword 2.

3.3 -exportresidual

This option will write a file containing the residuals of the model. The latter may be used in conjunction the "-skipreml" option3.9.

3.4 -exportsol

This option will write two files containing solutions for fixed (<aname>.fix.res>) and random (<aname>.rnd.res>) effects of the model.

3.5 -init

This option will run a serie of questions in order to create GLASCOW's parameter file. The parameter file described as a second argument will then be written and a complete new analyse will start.

3.6 -n T

This option will set the number of threads to be used by GLASCOW to T . By default one thread is used. When setting the number of threads, beware that memory requirement will increase proportionnaly and that performance scale-up are not linear ! To obtain the best efficiency, you may divide your genotype data in several small (< 500 Mb) files.

3.7 -map

This option will read a map file in order to output a ".assoc" file (that can be viewed with IGV as instance). The map file should contain as many line as the number of position tested. SNP should be sorted in the same order as in the analysis. The map format should be Chromosome, name of the SNP, Base pair

```
1 BovineHD0100000015 36337
1 BovineHD0100000024 67130
1 BovineHD0100000026 78655
```

3.8 -seed

This option will ask for a specific random seed in order to obtain some reproducible results.

3.9 -skipreml

This option will rely on variances defined in the parameter indicated in parameter file and read a residual file named after the project name (the declared in parameter file) and suffixed with ".residuals".

3.10 -version

This option will print the version of GLASCOW, and compilation date.

4 Misc

Last version of GLASCOW is available at : `http://www.giga.ulg.ac.be/jcms/prod_381171/software`

The archive is provided with 5 directories.

MAN contains GLASCOW manual (along with its LaTeXsource)

BIN contains GLASCOW executable

SRC Contains GASCOW source code with a Makefile

EX contains directories with samples data

.CHK contains tests scripts

- GLASCOW has been successfully compiled on linux with gfortran 4.6 and ifort 12.1. The provided Makefile will use gfortran without LAPACK by default, we strongly encourage to use LAPACK and ifort if the latter are available and correctly set-up.
- In order to use ifort instead of gfortran just replace the COMP variable in the Makefile.
- The Makefile and source code have been designed in order to be easily compiled with or without LAPACK. One just have to uncomment the LAPACK variable declaration in the Makefile.
- GLASCOW use OPENMP to speed-up computations
  - While constructing the genomic relationship matrix, if you indicate several phases file, each of them can be read by a separate thread. In order to obtain the best performances one can split phases files by chunk of chromosome. On a computer with 24 cores, a set of 5000 individuals genotyped on 600000 SNP (divided in 55 files) could be read in less then 20 minutes using less then 6 Gb of RAM.
  - The REML part by using mkl can also take advantage of shared memory, manycores architecture.
  - Last, permutations can also take advantage of a shared memory manycore archictecture. Anyway due to the implementation of this part, multithreading may not guarantee speed improvment.
- Some compilation flag may create conflict with openmp, thus avoid using -ipo with ifort or -static with gfortran.

5 Citation

If you use GLASCOW in a published analysis, please report the GLASCOW version used and cite the folowing article :

Z. Zhang, F. Guillaume, A. Sartelet, C. Charlier, M. Georges, F. Farnir and T. Druet. (2012). Ancestral haplotype-based association mapping with generalized linear mixed models accounting for stratification. submitted