User Manual for GIGI v1.06.1

Author:
Charles Y K Cheung [cykc@uw.edu]
Ellen M Wijsman [wijsman@uw.edu]
Department of Biostatistics
University of Washington

Last Modified on 1/31/2015

Contents

Introduction

GIGI is a C++ program to impute genotypes in pedigree data. It imputes missing genotypes of dense markers conditional on genotypes available on a subset of relatives in the pedigree and framework genotypes used to infer Inheritance Vectors (IVs). Our pedigree-based genotype imputation approach consists of two steps. The first step is to infer IVs at the positions of framework markers using gl_auto, which is a MCMC-based program that is part of the MORGAN package. The second step is to impute dense genotypes by GIGI using the IVs output by gl_auto and pedigree structure file from MORGAN.

GIGI is developed under the **linux** environment, but we have tested that it runs on **Mac OSX 10.10.1**.

In this documentation, we use the following terminology. Framework markers are a relatively sparse set of markers that are used to infer IVs on a chromosome of interest. Dense markers are markers with missing genotypes on some subjects that we want to impute. For example, these dense markers may be genotypes obtained from sequence data or from a dense SNP panel, and may be typed on fewer and even different subjects in the pedigree. See the publication describing GIGI,below, for more information.

Citing GIGI

Cheung, CYK., Thompson, E.A., Wijsman, E.M. GIGI: An approach to effective imputation of dense genotypes on large pedigrees. *American Journal of Human Genetics* 92(4): 504-516.

Software URL

http://faculty.washington.edu/wijsman/software.shtml

What’s New in version 1.06.1?

We have made changes so that GIGI is now compatible with the new Inheritance Vectors file format generated by MORGAN's gl_auto version 3.2! Starting in v1.06, the default behavior of GIGI is to use this new Inheritance Vectors file format. In addition, because this new file format no longer requires users to provide GIGI with the meiosis indexes, GIGI can now directly use the MORGAN pedigree file. (In the past version that uses MORGAN’s gl_auto version 3.1 or earlier, users must create the pedigree-meiosis file from manually parsing the console output of gl_auto.) Therefore, to avoid confusion, we suggest our users to use the pedigree file instead of the pedigree meiosis file from now on.

Ver. 1.06.1 corrected a bug introduced in ver. 1.06 in parsing the input information.

We also understand the importance of backward compatibility. If you intend to use gl_auto's output from the pre v3.2 and the pedigree meiosis file, you may do so by using the -IVformat31 flag.

Files in GIGI software distribution

GIGI software code and its dependency files - the Mersenne random number generator

- example folder

Inferring IVs using gl_auto

The first step to impute genotypes is to use **framework markers** to infer IVs. For this purpose, we use gl_auto, a program in the MORGAN package that is freely available at http://www.stat.washington.edu/thompson/Genepi/MORGAN/Morgan.shtml. To infer IVs in gl_auto, we need to supply the required files in MORGAN format:

(1) Pedigree file
(2) Marker file: this is a composite file that contains the map positions of framework markers (in centiMorgans assuming the Haldane map function), allele frequencies of framework markers , and genotype data of framework markers
(3) Parameter file used to run gl_auto

Refer to the documentation of MORGAN for guidance on setting up these files and on running gl_auto.

Example files used to infer IVs using gl_auto are included under the "example/gl_auto_example" directory.

Overview

From running gl_auto, we need to obtain the framework IVs file: GIGI uses this file, which contains IVs at framework positions.

Note: before version 1.06, GIGI also requires the pedigree-meiosis file. Unless you are using the -IVformat31 flag, you won't have to prepare the pedigree meiosis file from the console output of gl_auto. Refer to the GIGI_v1_04-Documentation if you intend to use the older version.

Obtaining the files

(a) Framework IVs file

In gl_auto's parameter file, we must instruct gl_auto to display the output as Meiosis Indicators instead of Founder Genome Labels. Please make sure we use the option

output meiosis indicators

instead of "output founder genome labels" in the control/parameter file of gl_auto.

Installing GIGI

Simply unzip the files, navigate to the code directory, and type

```
make
```

If make does not work, go to the GIGI.cpp's directory and install the program by

```
g++ GIGI.cpp -o GIGI
```

On Linux, GIGI was compiled successfully under Ubuntu 10.04, Mandriva Linux release 2009.0, and RedHat release 5.8, with g++ version-4.1.2, 4.3.2, and 4.5.2.
On Mac, GIGI was compiled successfully under OS X Yosemite 10.10.1.

Running GIGI

GIGI accepts a parameter file. To run GIGI, type

```
./GIGI <parameter file> <options>
```

To run the example file, go to the main GIGI program directory, and type

```
./GIGI example/param-v1_06.txt
```

Options

[The flags are case-sensitive.]

Flag	**Purpose**
-verbose	to run GIGI with extended output to the console
-seed=NUM	to change the default seed. NUM should be a number between 1 and 999999
-prog=NUM	to print the progress of imputation at every NUM markers
-oldParam	to run GIGI using the old (GIGI v1.00's) format of the parameter file [General users should ignore] *In v1.03 and older, this flag was called "-longParam".
-outD=	to specify the absolute directory path of where the output files will be created. If this flag is missing, the output files will be saved to the user's current directory. e.g. /home/charles/GIGI_output [no character '/' at the end of the directory name]
-long	(new in v1.04!) to read dense genotype marker file as the "long" format (rows are markers and columns are genotypes of individuals)
-IVformat31	(new in v 1.06!) To use the IV generated by MORGAN gl_auto version 3.1 (or earlier). When enabling this option, GIGI also requires users to provide the pedigree meiosis file instead of the MORGAN compatible pedigree file.

An example of running GIGI with additional options.

```
./GIGI example/param-v1_06.txt -verbose -outD=/home/charles/GIGI_output
```

An example of running GIGI with IV formatted from MORGAN gl_auto version 3.1:

```
./GIGI example/param_v1_05-oldIVformat.txt -IVformat31
```

Making the Parameter File

The parameter file tells GIGI where to look for the required files and where to save the output files. GIGI needs the pedigree meiosis file (that user prepares from the output of gl_auto), dense marker file, IVs file (from gl_auto), map of the sparse marker file, map of the dense marker file, and allele frequencies of the dense marker file. It produces the called genotype file, genotype probability file, and another file that prints the number of IVs that are consistent with the observed dense genotypes [for sanity check].

An **example** of the parameter file is found in the example directory under
example/param-v1_06.txt

In GIGI, the parameter file is organized as follows:
line 1 - filename of **the pedigree** file (or the pedigree meiosis file, if the -IVformat31 is used)
line 2 - **framework IVs** file
line 3 - **number of sampled realizations** in the IV file
line 4 - filename of the **map positions** of the **framework** markers file (cM based on Haldane map function)
line 5 - filename of the **map positions** of the **dense** markers file (cM based on Haldane map function)
line 6 - filename of the **dense marker genotype** file
line 7 - filename of the **allele frequencies** of the **dense** markers file
line 8 - **Call_method** t_1 t_2 (3 numeric values separated by space: eg. 2 0.6 0.8)

On line 8, we specified up to 3 values.

1. Call_method: use the value 1 to call the most likely genotype and 2 to use threshold-based calling

2 and 3. If threshold-based calling is used, then the threshold of t_1=0.8 and t_2=0.9 are set as defaults. This pair of thresholds can be changed by the user by specifying the 2nd and 3rd numerical values. The thresholds should be between 0.5 and <1. If only t_1 is specified, t_2 will be set to t_1+ (1- t_1)/2.

The format of the parameter file used here is different than that of an older format used in GIGI_v1.00. See Appendix A for the older format and how we can continue to use the older format of the parameter file. [not recommended]

Notes:
I suggest using the absolute paths of the filenames instead of relative paths. A relative path is relative to the directory containing the executable program. (The parameter file in the example folder is created using a relative path.)

*line 1: user creates this file from the gl_auto's console output. In version 1.01, GIGI can only process 1 pedigree at a time.
*line 2: when you run gl_auto, you should instruct gl_auto to print Meiosis Indicators instead of Founder Genome Labels
*line 3: – this corresponds to the number of samples that the user actually prints to the meisois indicator file.

File Formats

Examples of these files are provided in the example directory [refer to the param.txt for the filename of these files]

a. pedigree file [line 1]. This is the pedigree file used by MORGAN gl_auto to infer IVs. For an example of this file, please refer to the ped52.MORGAN.ped in the example directory.

Specifically, GIGI only cares that

1) the first word in the first line is "input" (to verify that this is a pedigree file)
2) the first subject in the pedigree starts on the fifth line, with the first 3 columns being
   subjectID fatherID momID

e.g.

```
input pedigree size 52
input pedigree record names 3 integers 2
input pedigree record trait 1 integer 2
*****
101 0 0 1 0
102 0 0 2 0
201 101 102 1 0
202 101 102 2 0
2010 0 0 2 0
```

For both gl_auto and GIGI, the pedigree should be ordered in a way such that the ancestors are specified before the descendants.

For more information, please refer to
http://www.stat.washington.edu/thompson/Genepi/MORGAN/morgan311-tut-html/morgan-tut_2.html - SEC13

b. framework IVs file [line 2]
The Inheritance Vectors file describes the descent pattern of chromosomes at the positions of the framework markers. It is the output file that **gl_auto** generates.

c. map positions of framework markers [line 4]
The marker map positions of the framework markers file is a text-file which contains the map distance in centi-Morgans (cM) based on the Haldane map function. Markers must be ordered in ascending order and consistent with the order used in gl_auto. Each line contains the position of a marker.

```
position of Marker1
position of Marker2
position of Marker3
```

...
position of MarkerN

eg
1.0
2.0
3.0
4.0
...

e. map positions of dense markers [line 5]

Similar to the marker map for framework panel of markers, the marker map of dense markers is a text-file which contains the map distance in centi-Morgans (cM) based on the Haldane map function. Markers must be ordered in ascending order. Each line contains the position of a marker.

eg
0.5
0.7
0.9
1.1
1.15
...

b. dense marker genotype file [line 6]

The dense markers are the markers that we want to impute on some individuals in the pedigree. The dense marker file contains the genotypes of observed individuals.

(a) This file is space-delimited
(b) The markers should be sorted by ascending map positions.
(c) Alleles are labeled numerically starting from 1,2,... in ascending order. We use 0 to indicate a missing allele. *If the original genotypes are in alpha-numeric, users first need to convert the genotypes to an indexed numerical format.* (GIGI supports imputation of multi-allelic markers.)

(new in v1.04) User have two options in the format of the genotype file:
(1) the newer *long* format (rows are markers) or
(2) the more traditional *wide* format (rows are individuals) . The long format should be used for file with many markers (e.g. > 10,000).

Rationale: The wide format is not suitable for analysis with a large number of markers. If we use the wide format, GIGI must read the entire genotype file into the computer memory. If the genotype file contains a large number of markers, reading the file requires allocation of large memory and may cause problem in memory allocation. On the other hand, if we use the long format, GIGI can process one marker at a time, so GIGI can handle arbitrarily large number of markers. Therefore, we recommend using the long format if the genotype file contains many markers.

Note: The default (assumed) genotype file format in GIGI is the traditional *wide* format. To let GIGI

know that your file is in the *long* format, you use “-long” flag.
e.g. ./GIGI example/param_longFormat.txt **-long**

Option 1. The *long* format
Each row of the file contains the genotypes for a *marker*. Consistent with the BEAGLE’s genotype data file format, this file requires a header line.

The header contains the following information:
id person1 person1 person2 person2 person3 person3 …

the id is the name of the marker. The pair of columns for each individual represents the first allele and second allele of the specified individual. Unlike BEAGLE’s genotype file format, GIGI’s long file format does not have the “I” column because GIGI assumes that every row contains a marker.

An example of this file looks like:

```
id 101 101 102 102 103 103
rs0001 1 1 1 2 1 1
rs0002 1 2 0 0 1 2
rs0003 2 2 2 2 1 2
```

For another example, see the “dense.genotypes.long” file under the Example folder.
Note: if the *long* format is used, the outputs will also be generated in the *long* format.

Option 2. The default *“wide”* format
Each row of the file contains the genotypes for an *individual*. This file does not contain a header line. The first column specifies the name of an individual. The subject_ID can be an alphanumeric string. In this “wide” format, the name of the marker is not retained.

Subject_ID allele1_marker1 allele2_marker1 allele1_marker2 allele2_marker2 allele1_marker3 allele2_marker3 ...
An example of this file looks like:

```
101 1 1 1 2 2 2
102 1 2 0 0 2 2
103 1 1 1 2 1 2
```

6. allele frequencies of the dense marker file [line 7]
Each row contains the allele frequencies of the dense markers. The first column is the allele frequency of allele 1, the second column is the allele frequency of allele 2, etc... Each row must sum to 1. The allele frequencies file is space-delimited.

allele frequencies of marker 1
allele frequencies of marker 2
allele frequencies of marker 3
...

eg.

```
0.4 0.6
0.2 0.3 0.5
0.8 0.2
...
```

Output files

Three (or Four) output files are created in the directory where GIGI is run:

1. impute.geno

The imputed genotype file has the same format as the input dense genotype file.

2. impute.prob

The imputed genotype probabilities file displays the genotype probabilities for each marker. For each bi-allelic marker, three columns are used to store the probability of genotype configuration 1/1, 2/1, and 2/2. For each tri-allelic marker, the probabilities are displayed and space-delimited with 1/1, 2/1, 2/2, 3/1, 3/2, 3/3. For multi-allelic markers in general, the appropriate number of columns are used to store all the combinations.
Markers are separated by tabs.

For the *long* format*:*
e.g.
id person1 person1person1   person2 person2 person2 … (as is specified in the header of the file).

- id: is the name of the marker
- genotype probabilities of different individuals are separated by a tab
- forthe number of columns correspond to the

Note: In the header, 3 columns are specified for each individual, which has the implicit assumption that each marker is di-allelic. However, GIGI does not want to limit imputation to diallelic marker. Instead, GIGI will continue to output the estimated probabilities of all genotype configurations in multi-allelic markers. In this case, the number of columns (>3) for that particular multi-allelic marker would not match the header. A warning is generated in GIGI to remind users to apply caution when interpreting the results from this file.

For the *wide* format*:*
e.g.
subject_ID genotype_11_marker1 genotype_21_marker1 genotype_22_marker1 genotype_11_marker2 genotype_21_marker2 genotype_22_marker2

```
101 0.9 0.1 0           0.8 0 0.2       ...
102 0.9 0.1 0           1 0 0   ...
...
```

3. impute.consistentIV

The consistency file counts the number of IVs consistent with the observed genotypes of each dense marker. This file is used for a sanity check. A number that is low compared to the number of IVs used may indicate incompatibility between observed genotypes and the sampled IVs.

4. impute.dosage (new in v1.03 !)

If all dense markers to be imputed are di-allelic, the dosage file will also be created. The dosage summarizes the expected percent of the **allele-1** in each genotype:

dosage of a genotype = 1*P(genotype is 1/1) + 0.5*P(genotype is 1/2)

Markers are separated by space:

For the *long* format, the file is formatted as:

id dosage_person1 dosage_person2 dosage_person3…

- id is the name of the marker

For the *wide* format, the file is formatted as:

```
Subject_ID   dosage_marker_1   dosage_marker_2   dosage_marker_3   dosage_marker_4   …
101 0.5 1 0.875 0
102 1 0.5 0.5 0
…
```

Note: **allele-1** is not necessarily the rare allele. It is the user’s responsibility to define what allele-1 is.

Frequently Asked Questions (FAQs)

1. **How do I choose framework markers?** Because the framework markers are assumed to be sparse, we want to choose framework markers that are informative about which chromosomes are being transmitted at the framework loci. First, framework markers typed on a large number of subjects tend to be most informative. Second, framework markers that are multi-allelic tend to be more informative than di-allelic markers if they are available. Third, if framework markers are SNPs, markers with high minor allele frequencies in the sample tend to be more informative. Fourth, framework markers should be moderately but not too dense (eg. not denser than 1 marker per 0.3 cM because of concern about MCMC mixing and violation of the assumption of Linkage Equilibrium) If the framework markers are multiallelic, this spacing should be greater, e.g., not denser than 1 marker per 2 cM for good MCMC mixing.

2. **What do I have to be careful about the genetic map that I am using?** The map positions are in map distances based on the Haldane map function, instead of Kosambi map function or sequence positions. If this map is based on the Kosambi map function, the user will need to convert the map positions to Haldane map function using an appropriate conversion method.

   Also, since recombination fractions are relative to each other, we strongly encourage the user to generate both the framework map positions and the dense marker positions at the same time.

3. **Can GIGI process multiple pedigrees?** No. In version 1.01, GIGI can only process 1 pedigree at a time. For this reason, gl_auto should also be run individually for each pedigree.

4. **Can I use GIGI to impute genotypes on unrelated individuals?** No.

5. **What should I know when I use GIGI to impute genotypes on trios?** Since there is no way to infer recombination on trios using unphased genotypes, gl_auto produces an empty inheritance vectors file. From the pedigree-meiosis file, GIGI recognizes that this inheritance vectors file is empty but will proceed to impute genotypes. For trios, imputing genotypes using GIGI is equivalent to imputing genotypes conditional on the pedigree structure and minor allele frequencies.

   Information about linkage-disequilibrium is potentially useful for genotype imputation on trios. Since the current version of GIGI does not leverage information from linkage-disequilibrium, you may want to use existing linkage-disequilibrium-based software that supports imputation on trios (e.g. BEAGLE).

Appendix A

GIGI v1.00 used an old format for the parameter file. It is not recommended for new users.
For old users, if you want to continue to use this old format, use the flag "-longParam" when you run GIGI. This file format allows you to specify the exact filename and location of the output files.
e.g. ./GIGI example/oldparamFormat.txt -oldParam

The old file format:

line 1 - filename of the pedigree meiosis file
line 2 - number of pedigrees, current max is 1. Just use the value **1**
line 3 - filename of the dense marker file
line 4 - IVs file
line 5 - number of saved realizations in the IVs file
line 6 - filename of the map positions of the framework markers file (cM – Haldane map distance)
line 7 - filename of the map positions of the dense markers file (cM – Haldane map distance)
line 8 - filename of the allele frequencies of dense markers file
line 9 - random seed number (e.g. 1234)
line 10- *filename of the output 1: called genotypes*
line 11- Use the value **2**. This is the calling method to use. 1: most likely genotype calling, 2: threshold-based calling
line 12- the first threshold of threshold-based calling: t1- to call both alleles (this number should be between 0.5 and <1). e.g. 0.9
line 13- the second threshold of threshold-based calling: t2- to call 1 of 2 alleles, if GIGI can't call both alleles under the threshold specified in line 12. (a number that is between the first threshold <1). e.g. 0.9
line 14- *filename of output 2: estimated genotype probabilities*
line 15- *filename of output 3: number of IVs consistent with each observed dense markers*

Appendix B

If you intend to use the IV generated by MORGAN v3.1 (or earlier), you should use the Pedigree-Meiosis file instead of the pedigree file.

You have to create the Pedigree-Meiosis file manually from the console output of gl_auto.

Using a text editor, we open the console_output.txt and fetch the line that begins with
“name   name.pa   name.ma Compnt pat.meio mat.meio”
We copy this line and table below. We paste this table to another file and save it.
The file includes the header line and looks like this:

```
 name      name.pa      name.ma Compnt pat.meio mat.meio
        2100_6           0             0      1      0      0
       2100_21           0             0      1      0      0
       2100_25           0             0      1      0      0
       2100_29           0             0      1      0      0
       2100_31           0             0      1      0      0
       2100_39           0             0      1      0      0
…
      2100_907           0             0      1      0      0
      2100_908      2100_901      2100_902      1      2      1
      2100_909           0             0      1      0      0
      2100_910      2100_901      2100_902      1      4      3
      2100_911      2100_901      2100_902      1      6      5
      2100_915      2100_907      2100_908      1      8      7
```

...(until the end of table)

ChangeLog

The ChangeLog describes the recent changes to the program. It is located in the root directory of the uncompressed file.

License

There is NO WARRANTY for the program, to the extent permitted by applicable law. In no event unless required by applicable law will any GIGI copyright holder be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use the program (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or a failure of the program to operate with any other programs).

Acknowledgement

Supported by funding from the National Institutes of Health, grants R37GM046255, P01HL030086, and P50AG05136.

I would also like to thank Elizabeth Marchani, Mohamad Saad, Alejandro Nato, and the Staistical Genetics community for providing valuable feedback as testers of GIGI.