haplofreq: Software for Determining all Possible Haplotypes and their Estimated Frequencies.

Author: Nikola Rasevic

Student Number: 7748976

Department of Mathematics and Statistics

University of Ottawa

Dec. 21st, 2020

Abstract

The estimation of haplotype frequencies of a given genotype dataset has multiple uses, particularly in estimating haplotypic phase. Laboratory techniques that determine phasing do exist but are time-consuming and cost inefficient.[2] Thus, algorithms have been proposed to estimate haplotype frequencies and phase based on these frequencies. One such solution using the expectation-maximization algorithm was developed by Excoffier and Slatkin (1995).[5] The goal of the haplofreq Python package is to use this algorithm to output all possible haplotypes and their frequencies for a given genotype dataset. This can then be used for phasing purposes.

Introduction and Background

The goal of this algorithm is to identify all the possible haplotypes of a given genetic dataset containing individuals with genotyped SNPs and estimate the frequency for each of these haplotypes. This is accomplished by using the expectation-maximization (EM) algorithm described in Excoffier and Slatkin (1995).[5] SNP array data usually takes the forms of unphased genotypes, which means that it can not be directly determined which of the two inherited chromosomes an allele is located in. With the help of estimated haplotype frequencies, the haplotype phase can be imputed. This is accomplished by sampling the distribution of all possible genotypes, given their estimated frequencies. Laboratory methods have been developed to determine phasing, but these methods are more expensive and labour intensive than computational approaches.[2] Browning and Browning (2011) listed a multitude of applications for haplotype phasing.[2] These include imputing genetic variants, detecting genotype error, inferring recombination points and much more.[1,8,9,10,12]

As mentioned previously, the EM algorithm developed by Excoffier and Slatkin (1995) is to be used to determine the haplotype frequencies.[5] The EM algorithm was first introduced by Dempster et al. (1977).[4] This algorithm is an iterative approach of computing maximum-likelihood estimates of incomplete data. Then, Hudson (1990) gave mention that the EM algorithm may be implemented to infer phase relationships of haplotypes.[6] Thus, Excoffier and Slatkin (1995) developed the algorithm to estimate haplotype frequencies which enables the inference of phased haplotypes.[5]

In order to be consistent, the terminology used by Excoffier and Slatkin (1995) will be used here.[5] A phenotype is defined as a multiple locus genotype whose haplotype phase is unknown. A genotype is defined as a combination of two multiple locus haplotypes. In this algorithm, the expectation step is the probability of resolving each phenotype into its different possible genotypes, using the haplotype frequencies of the previous step. The maximization step is the calculation of each of the haplotype frequencies.

Statistical Description and Algorithm

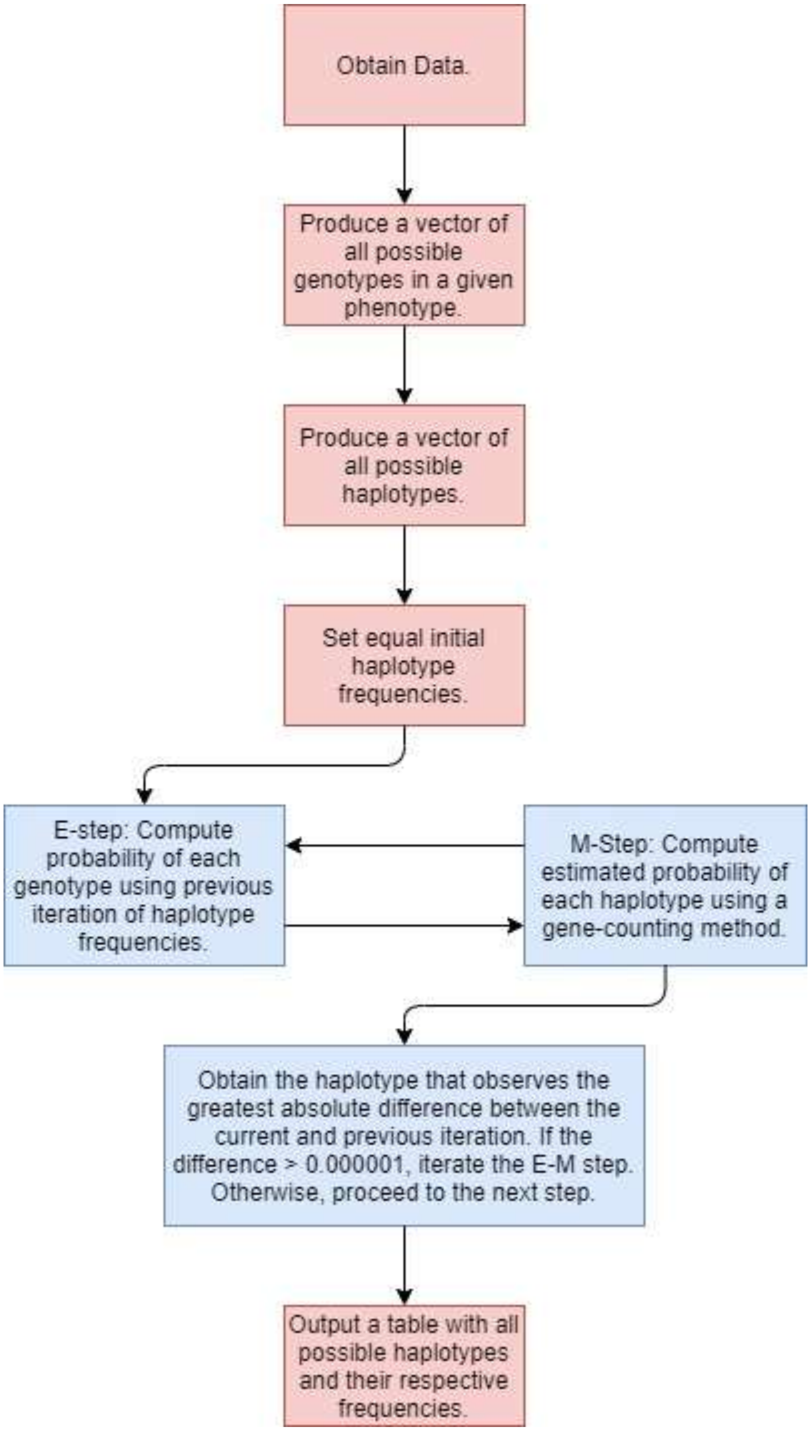
Obtain Data.
Produce a vector of all possible genotypes in a given phenotype.
Produce a vector of all possible haplotypes.
Set equal initial haplotype frequencies.
E-step: Compute probability of each genotype using previous iteration of haplotype frequencies.
M-Step: Compute estimated probability of each haplotype using a gene-counting method.
Obtain the haplotype that observes the greatest absolute difference between the current and previous iteration. If the difference > 0.000001, iterate the E-M step. Otherwise, proceed to the next step.
Output a table with all possible haplotypes and their respective frequencies.

Figure 1. Simplified Explanation of the haplofreq() Algorithm. Blue represents the iterative process.

The algorithm is described in Figure 1. When a dataset containing individuals and their genotyped SNPs is obtained, each individual has a phenotype: a multiple locus genotype whose phase is unknown. Thus, if there are m individuals, there are m phenotypes. The number of possible genotypes: c_j, for the jth phenotype is a function of the number of heterozygous loci: s_j. c_j is defined in Formula 1.

Formula 1:

$$c_j = 2^{s_j - 1}, s_j > 0$$

$$c_j = 1, s_j = 0$$

When implementing the EM algorithm, the initial haplotype frequencies were set to be equal for each genotype. This is shown in Formula 2. The estimated frequency for haplotype t is $\hat{p}_t$.

Formula 2:

$$\hat{p}_t^{(1)} = \frac{1}{\#\, of\ haplotypes}$$

The expectation step involves using the haplotype frequencies calculated in the previous iteration to determine the probability of observing a genotype in the gth iteration. This is calculated using Formula 3. Two terms of Formula 3 can be obtained from Formulas 4 and 5. $P_j(h_k h_l)$ is the probability of the ith genotype in the jth phenotype being made up of haplotypes k and l. The calculation for this probability is shown in Formula 4. P_j is the probability of the jth phenotype which is given in Formula 5.

Formula 3 (Expectation Step):

$$P(h_k h_l)^{(g)} = \frac{n_j}{n} * \frac{P_j(h_k h_l)^{(g)}}{P_j^{(g)}}$$

Formula 4:

$$P_j(h_k h_l) = p_k^2, k = l$$

$$P_j(h_k h_l) = 2 p_k p_l, k \neq l$$

Formula 5:

$$P_j = \sum_{i=1}^{c_j} genotype\ i = \sum_{i=1}^{c_j} P_j(h_k h_l)$$

The maximization step is computed using a procedure equivalent to the gene counting method.[3,13] This is shown in Formula 6. $\delta_{it}$is an indicator variable that is equal to the number of times haplotype t is present in genotype i. The number of phenotypes is represented by m.

Formula 6 (Maximization Step): $\hat{p}_t^{(g+1)} = \frac{1}{2}\sum_{j=1}^{m}\sum_{i=1}^{c_j}\delta_{it}P_j(h_k h_l)^{(g)}$

This algorithm continues until convergence. In the haplofreq() function, convergence is when the maximum absolute difference between the updated haplotype frequencies, and the previous iteration of haplotype frequencies is below 10^{-6}.

Example of one Iteration

Often, it may be difficult to grasp the concept of an algorithm when only formulas are displayed. As such, an example of one iteration is provided. A simple dataset containing 3 individuals and 2 SNPs is to be used for the example. This dataset is displayed in Table 1. Note, this is not the format of the inputted dataset for the function. The number of genotypes: c_j is also displayed.

Table 1. Individuals/Phenotypes with Two Genotyped SNPs.

	Individual/Phenotype 1		**Individual/Phenotype 2**		**Individual/Phenotype 3**	
SNP 1	G	T	G	G	T	T
SNP 2	C	T	C	C	C	T
c_j	2		1		1	

There are four possible genotypes, two for phenotype 1 and one each for phenotypes 2 and 3. These genotypes are shown in Table 2.

Table 2. All Possible Genotypes Given the Phenotype Data.

	Phenotype 1				**Phenotype 2**		**Phenotype 3**	
	Genotype 1		**Genotype 2**		**Genotype 3**		**Genotype 4**	
SNP 1	G	T	G	T	G	G	T	T
SNP 2	C	T	T	C	C	C	C	T

There are four possible haplotypes given the genotype data. An initial haplotype frequency is assigned to each haplotype. The haplotype frequency is determined in Formula 2. Since there are four haplotypes, each haplotype has an initial frequency of ¼. This is seen in Table 3.

Table 3. All Possible Haplotypes Given the Genotype Data. Initial frequencies are set to be equal across all haplotypes.

	Haplotype 1	**Haplotype 2**	**Haplotype 3**	**Haplotype 4**
SNP 1	G	G	T	T
SNP 2	C	T	C	T
Initial Frequency	1/4	1/4	1/4	1/4

In the expectation step, the probability of resolving each phenotype into the different possible genotypes are calculated. Genotypes one and three will be used as examples. Formula 3 is used for this step. Note, Formulas 4 and 5 are also used in conjunction with Formula 3. The results for all genotype frequencies are in Table 4.

Genotype 1:

$$P(h_1h_4)^{(1)} = \frac{1}{3} * \frac{P_j(h_1h_4)^{(1)}}{P_j^{(1)}} = \frac{1}{3} * \frac{2*\left(\frac{1}{4}\right)*\left(\frac{1}{4}\right)}{2*\left(\frac{1}{4}\right)*\left(\frac{1}{4}\right)+2*\left(\frac{1}{4}\right)*\left(\frac{1}{4}\right)} = \frac{1}{6}$$

Genotype 3:

$$P(h_1h_1)^{(1)} = \frac{1}{3} * \frac{P_j(h_1h_1)^{(1)}}{P_j^{(1)}} = \frac{1}{3} * \frac{\left(\frac{1}{4}\right)^2}{\left(\frac{1}{4}\right)^2} = \frac{1}{3}$$

Table 4. Expectation Step: Computing Genotype Frequencies.

	Phenotype 1				**Phenotype 2**		**Phenotype 3**	
	Genotype 1		**Genotype 2**		**Genotype 3**		**Genotype 4**	
SNP 1	G	T	G	T	G	G	T	T
SNP 2	C	T	T	C	C	C	C	T
$P(h_kh_l)^{(1)}$	1/6		1/6		1/3		1/3	

After calculating the probability of each genotype, the haplotype frequencies can then be estimated. The frequency of haplotype 1 is given as an example. Formula 6 is used for this calculation. The updated haplotype frequencies can be seen in Table 5.

$$\hat{p}_1^{(2)} = \sum_{j=1}^{m}\sum_{i=1}^{c_j} \delta_{i1} P_j(h_k h_l)^{(1)} = \left(\frac{1}{2}\right)\left(1\left(\frac{1}{6}\right) + 0\left(\frac{1}{6}\right) + 2\left(\frac{1}{3}\right) + 0\left(\frac{1}{3}\right)\right) = 5/12$$

Table 5: Maximization Step: Computing Estimated Haplotype Frequencies.

	Haplotype 1	Haplotype 2	Haplotype 3	Haplotype 4
SNP 1	G	G	T	T
SNP 2	C	T	C	T
Initial Frequency	1/4	1/4	1/4	1/4
Updated Frequency	5/12	1/12	1/4	1/4

As mentioned, this process is iterated until convergence. To determine convergence, the haplotype that observes the greatest absolute difference of frequency between the current and previous iteration is obtained. If the difference is greater than 10^{-6}, the algorithm is re-iterated. Otherwise, all the possible haplotypes and their estimated frequencies are outputted.

Using haplofreq

Python 3 needs to be installed on your computer to run the package.[14] Since the package is local, make sure to run your Python scripts in the "Final Assignment" folder. To begin estimating haplotype frequencies with haplofreq(), genotype data must be loaded in as a Pandas data frame. Genotype data must contain SNPs and individuals in a .ped file. A .ped file is a genotype dataset that contains individuals on the rows and SNPs and further information on the columns and are separated by a white space. There is no header in .ped files. The first 6 columns of a .ped file represent the Family ID, Individual ID, Paternal ID, Maternal ID, Sex (1 for male and 2 for female) and Phenotype. Note, the phenotype in the .ped file is not the same as the phenotype in the algorithm. Columns 7 and onwards represent the SNPs. For example, columns 7 and 8 represent the 1st SNP. Columns 9 and 10 represent the 2nd SNP and so on. Further information of .ped files can be found in the following url: https://zzz.bwh.harvard.edu/plink/.[11] The code for the haplofreq() function is located in EM.py.

There cannot be any missing data and all SNPs must be bi-allelic. Uploading the .ped data frame can be accomplished with the following syntax. The dataset name is FILE.ped.

```
import pandas as pd
peddf = pd.read_csv("directory\\FILE.ped", delim_whitespace=True, header = None)
```

Once the genotype data is read as a Pandas data frame, all the possible haplotype frequencies can be determined using the haplofreq() function. This is outputted as a table. The first column represents all possible haplotypes within square brackets. The second column is the haplotype's respective frequency. In the case of the above uploaded data, the syntax for generating this output is as follows:

```
haplofreq(peddf)
```

Furthermore, Python style help documentation is provided with the following syntax:

```
help(haplofreq)
```

Examples of haplofreq

A total of seven datasets are included in the package. Each dataset contains six individuals and a varying number of SNPs. You can access the Guide.ipynb Jupyter file for a step by step guide with the example data frames. Jupyter notebook can be installed from the following url: https://jupyter.org/.[7] The 2snps.ped file will be used as an example. In order to upload the file as a Pandas dataset, the following syntax is required:

```
Import pandas as pd
data = pd.read_csv("haplofreq\\2snps.ped", delim_whitespace=True, header = None)
```

Next, the haplotype frequencies for all possible haplotypes are computed using the following syntax:

```
haplofreq(data)
```

The output is displayed in Table 6.

Table 6. haplofreq() Output for the 2snps.ped Dataset.

Haplotypes	Frequency
[G, C]	0.5
[G, T]	0.166667
[T, C]	1.38533e-07
[T, T]	0.333333

Notes

As mentioned before, the .ped file must contain only bi-allelic SNPs and cannot contain any missing data. If the dataset contains such data, the whole SNP or individual must be removed. Secondly, the algorithm is computationally intensive. There are 2^n possible haplotypes where n is the number of SNPs. With an increase in the number of SNPs, there is an increase in the number of haplotype frequencies being calculated which increases computational time. It is not recommended to use haplofreq() on datasets that contain more than roughly 10 SNPs. Figure 2 displays the time it takes for the algorithm to run as a function of the number of SNPs. As shown, the time it takes to calculate haplotype frequencies increases exponentially with increasing number of SNPs.

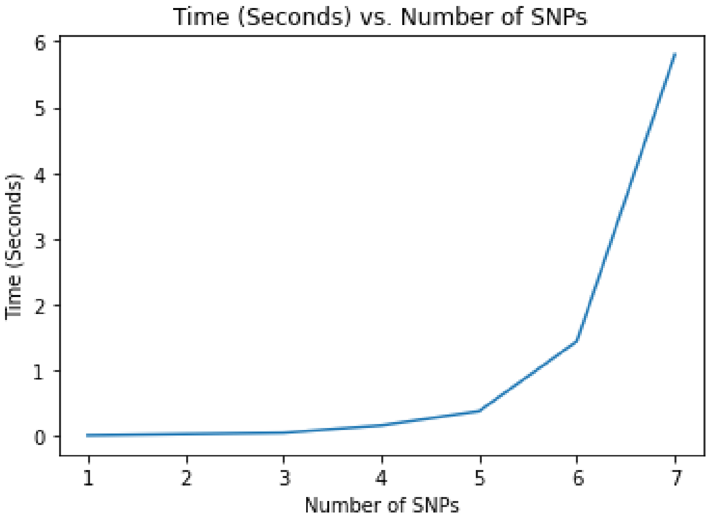

Figure 2. Time Elapsed vs. the Number of SNPs in the Dataset.

Summary

An algorithm for estimating haplotype frequencies was developed by Excoffier and Slatkin (1995).[5] A Python package called haplofreq containing the haplofreq() function was developed using this algorithm. The function takes .ped files as input and outputs all possible haplotypes and frequencies. This output can be used for phasing. Several limitations exist in the package, including exhaustive computation, and no methods for handling non bi-allelic or missing alleles. Future updates of the package will be conducted in order to address these limitations. As well, a new phasing function that takes advantage of the outputted haplotype frequencies is to be implemented in future updates.

References

1. Browning, B. L., & Browning, S. R. (2009). A unified approach to genotype imputation and haplotype-phase inference for large data sets of trios and unrelated individuals. American journal of human genetics, 84(2), 210–223. https://doi.org/10.1016/j.ajhg.2009.01.005
2. Browning, S. R., & Browning, B. L. (2011). Haplotype phasing: existing methods and new developments. Nature reviews. Genetics, 12(10), 703–714. https://doi.org/10.1038/nrg3054
3. CEPPELLINI, R., SINISCALCO, M., & SMITH, C. A. (1955). The estimation of gene frequencies in a random-mating population. Annals of human genetics, 20(2), 97–115. https://doi.org/10.1111/j.1469-1809.1955.tb01360.x
4. Dempster, A., Laird, N., & Rubin, D. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society. Series B (Methodological), 39(1), 1-38. Retrieved December 13, 2020, from http://www.jstor.org/stable/2984875
5. Excoffier, L., & Slatkin, M. (1995). Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. Molecular biology and evolution, 12(5), 921–927. https://doi.org/10.1093/oxfordjournals.molbev.a040269
6. Hudson R. R. (1990). Genetic Data Analysis. Methods for Discrete Population Genetic Data. Bruce S. Weir. Sinauer, Sunderland, MA, 1990. xiv, 377 pp., illus. $48; paper, $27. Science (New York, N.Y.), 250(4980), 575. https://doi.org/10.1126/science.250.4980.575
7. Kluyver, T., Ragan-Kelley, B., Fernando P&#x27;erez, Granger, B., Bussonnier, M., Frederic, J., … Willing, C. (2016). Jupyter Notebooks – a publishing format for reproducible computational workflows. In F. Loizides & B. Schmidt (Eds.), Positioning and Power in Academic Publishing: Players, Agents and Agendas (pp. 87–90).
8. Kong, A., Masson, G., Frigge, M. L., Gylfason, A., Zusmanovich, P., Thorleifsson, G., Olason, P. I., Ingason, A., Steinberg, S., Rafnar, T., Sulem, P., Mouy, M., Jonsson, F., Thorsteinsdottir, U., Gudbjartsson, D. F., Stefansson, H., & Stefansson, K. (2008). Detection of sharing by descent, long-range phasing and haplotype imputation. Nature genetics, 40(9), 1068–1075. https://doi.org/10.1038/ng.216
9. Li, Y., Willer, C. J., Ding, J., Scheet, P., & Abecasis, G. R. (2010). MaCH: using sequence and genotype data to estimate haplotypes and unobserved genotypes. Genetic epidemiology, 34(8), 816–834. https://doi.org/10.1002/gepi.20533
10. Marchini, J., Howie, B., Myers, S., McVean, G., & Donnelly, P. (2007). A new multipoint method for genome-wide association studies by imputation of genotypes. Nature genetics, 39(7), 906–913. https://doi.org/10.1038/ng2088
11. Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira MAR, Bender D, Maller J, Sklar P, de Bakker PIW, Daly MJ & Sham PC. (2007). PLINK: a toolset for whole-genome association and population-based linkage analysis. American Journal of Human Genetics, 81.
12. Scheet, P., & Stephens, M. (2008). Linkage disequilibrium-based quality control for large-scale genetic studies. PLoS genetics, 4(8), e1000147. https://doi.org/10.1371/journal.pgen.1000147
13. SMITH C. A. (1957). Counting methods in genetical statistics. Annals of human genetics, 21(3), 254–276. https://doi.org/10.1111/j.1469-1809.1972.tb00287.x
14. Van Rossum, G., & Drake Jr, F. L. (1995). Python tutorial. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands.