sequoia

Jisca Huisman
jisca.huisman@gmail.com

June 29, 2016

`Sequoia` provides a method to reconstruct multi-generational pedigrees based on SNP data, as described in the manuscript "Pedigree reconstruction using SNP data: parentage assignment, sibship clustering, and beyond". The bulk of the algorithm is written in Fortran, to minimise computation times.

An example R script for pedigree reconstruction is given below, followed by detailed description of each step.

Example

An example pedigree and associated life history file are provided with the package, which can be used to try out the steps detailed in this vignette. This pedigree consists of 5 generations with interconnected half-sib clusters (Pedigree II in the manuscript).

```
# install the package (only required once)
install.packages("sequoia")

# set the working directory
setwd("E:/Sequoia/test")

# load the package
library(sequoia)

# copy the example pedigree and associated life history file to the working
# directory.
file.copy(system.file("Ped_HSg5.txt", package="sequoia"), getwd())
file.copy(system.file("LH_HSg5.txt", package="sequoia"), getwd())

# simulate genotype data for 200 SNPs, and use otherwise default values
SimGeno(PedFile = "Ped_HSg5.txt", nSnp = 200)

# run the preparation step, duplicate checking and parentage assignment,
```

```
# but not yet the slower sibship clustering. Iterate as necessary,
# weeding out duplicated and erroneous samples (using PLINK's toolkit),
# and adding estimated birth years to the life history file.
sequoia(GenoFile = "SimGeno.txt", LifeHistFile = "LH_HSg5.txt", Sibships = FALSE)

# compare the assigned parents to those in the true pedigree
PedCompare(PedIN = "Ped_HSg5.txt", PedOUT = "Parents_assigned.txt")

# run sibship clustering
sequoia(Prep = FALSE, CheckDup = FALSE, Parentage = FALSE, Sibships = TRUE)

# compare the assigned real and dummy parents to the true pedigree
PedCompare(PedIN = "Ped_HSg5.txt", PedOUT = "PedSeq.txt")
```

1 Input

Two files are required as input. The first one contains the SNP data, with one line per individual, and one column for IDs followed by one column per SNP, where each SNP is coded as 0, 1, 2 copies of the reference allele, or missing (-9). This file format can for example be obtained using PLINK [Purcell et al., 2007] in combination with `sequoia`'s 'Prep' step, as described below.

The other file contains three columns: individual ID, sex (1 = female, 2 = male, other numbers = unknown), and birth year. Ideally all genotyped individuals are included in this life history file with sex and (estimated) birth year information, but this is not necessary. The life history file may include many more individuals than the genotype file, or in a different order. In species with more than one generations per year, a finer time scale than year of birth ought to be used, ensuring that parents are never born within the same time unit as their offspring.

Selection of SNP markers Using tens of thousands of SNP markers for pedigree reconstruction is unnecessary, will slow down computation, and may even hamper inferences by their non-independence. Rather, a subset of SNPs in low linkage disequilibrium (LD) with each other, and with high minor allele frequencies (MAF > 0.3), ought to be selected first. The calculations assume independence of markers, or absence of LD in the founder population. While low (background) levels of LD are unlikely to interfere with pedigree reconstruction, high levels may give spurious results. Markers with a high MAF provide the most information, as although rare allele provide strong evidence when they are inherited, this does not balance out the rarity of such events.

Creating a subset of SNPs can be done conveniently using PLINK (`http://pngu.mgh.harvard.edu/~purcell/plink/`), using for example the command

```
plink --file mydata --maf 0.4 --indep 50 5 2
```

which will create a list of SNPs with a minor allele frequency of at least 0.4, and which

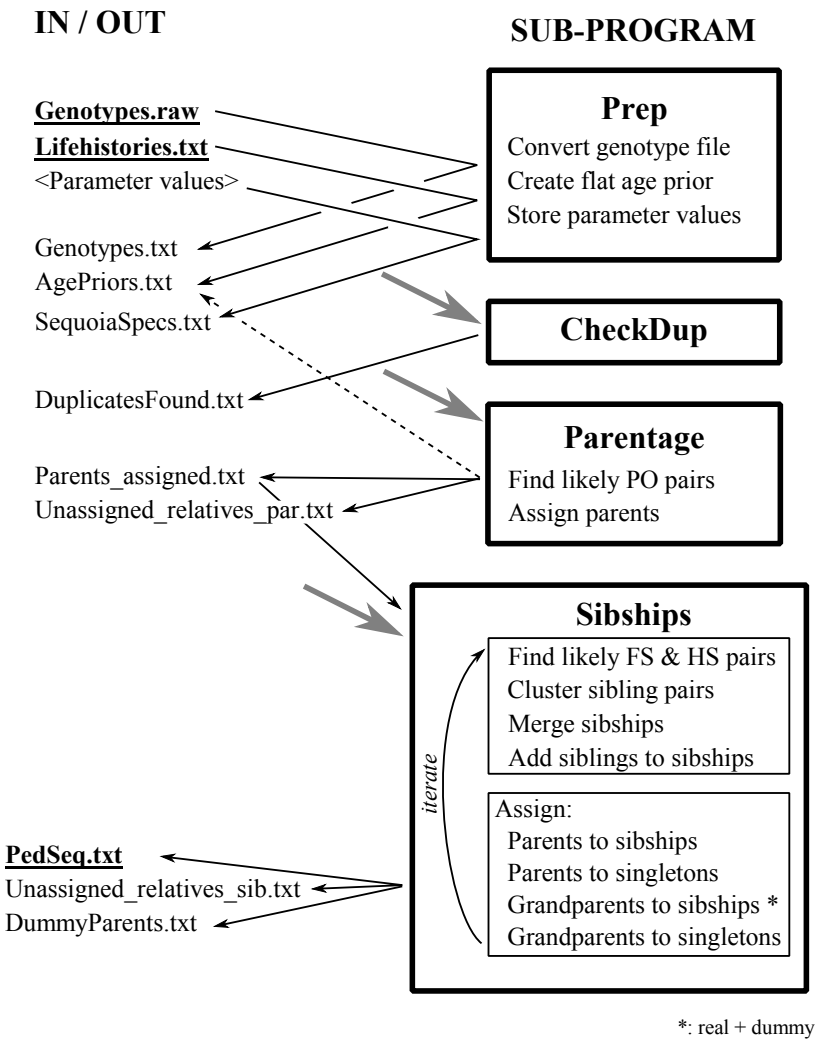

Figure 1: Overview of the input files required and output files generated by the various sub-programs of Sequoia. 'CheckDup', 'Parentage' and 'Sibship' each require a genotype file (Genotypes.txt), a lifehistory file (Lifehistories.txt), the age-difference based prior probabilities (AgePriors.txt) and the parameter values for the run (SequoiaSpecs.txt), symbolised by the grey incoming arrows.

in a window of 50 SNPs, sliding by 5 SNPs per step, have a VIF of maximum 2. VIF, or variance inflation factor, is $1/(1-r^2)$. It is advised to 'tweak' the parameter values until a set with a few hundred SNPs (300-700) is created. For further details, see `http://pngu.mgh.harvard.edu/~purcell/plink/summary.shtml#prune`.

The resulting list ('plink.prune.in') can be used to create the genotype file used as input for Sequoia, with SNPs codes as 0, 1, 2, or NA, with the command

```
plink --file mydata --extract plink.prune.in --recodeA --out inputfile_for_sequoia
```

which will create a file with the extension .RAW.

Simulating SNP data When SNP data is not (yet) available, but an approximate pedigree is, it is possible to test `sequoia` on a simulated dataset. This may be useful to

for example explore the number of markers required to reliably infer a pedigree of that particular structure.

The function `SimGeno()` can be used for this, which lets the user specify the average proportion of missing genotypes per individual, the genotyping error rate, and the fraction of known parents (in the supposed 'true' pedigree) which have not been genotyped. Moreover, the data can be made to contain a fraction of low-quality samples, to assess whether inclusion of samples which did not pass stringent quality control would improve or hamper pedigree reconstruction.

2 Preparation

The program `sequoia` consists of various sub-programs, which are all called via

```
sequoia(RawFile = NULL, GenoFile = NULL, LifeHistFile = NULL,
Prep = TRUE, CheckDup = TRUE, Parentage = TRUE, Sibships = TRUE)
```

details of the various optional parameters can be found using `?sequoia`, and the sub-programs 'Prep' (prepare), 'CheckDup' (check for duplicates), 'Parentage' and 'Sibships' (sibship clustering and grandparent assignment) are described below. `RawFile` denotes the genotype file created by PLINK, and `LifeHistFile` the lifehistory file with invidual ID, sex and birth year.

The preparation step amongst others slightly reformats the genotype file, by removing the header row, removing columns 2–6, which in Plink are intended for family ID, sex and phenotypic data, and replacing 'NA' by '-9' for missing values. Genotype files which are already in this format, such as simulated genotype files, can be specified as `GenoFile`.

Setting parameter values When the program is called, with `Prep = TRUE`, it takes the parameter values supplied, or the default values, and writes those to the file 'SequoiaSpecs.txt' in the current working directory, or the specified directory (option `Dir`). This file is used by the Fortran part of the program, which does all the heavy lifting. Please do not alter the name of this file, or the number of rows in it. Do feel free to change parameter values within this file, provided the number of SNPs or individuals does not exceed the total of SNPs or individuals in the genotype file.

Check for duplicates The data may contain positive controls, as well as other intentional and unintentional duplicated samples, which ought to be removed prior to parentage assignment. Sequoia includes a function to quickly search the data for identical genotypes, called with the option `CheckDup = TRUE`. It allows a few mismatches between the genotypes (depending on the assumed genotyping error rate), with or without the same individual ID. Note that when the number of SNPs is limited, very inbred individuals may be nearly indistinguishable from their parent(s); such individuals should not be excluded.

This function additionally searches the life history file for duplicated entries, and will also print a list of individuals included in the genotype file, but not in the life history file. This latter list is merely a service to the user; individuals without life history information can often be successfully included in the pedigree.

Age based prior During preparation the file 'AgePriors.txt' is created, which contains 8 columns, and as many rows as the birth year range detected in the life history data input file. It initially only indicates whether a given relationship is biologically possible (1) or not (0) for a given age difference between individuals, where the first row is for individuals born in the same year, the second row for individuals born one year apart, etc. The columns are labelled for various relationship categories, with M = mother, P = father, MS = maternal sibling, PS = paternal sibling, MGM = maternal grandmother, PGF = paternal grandfather, MGF = maternal grandfather and paternal grandmother, and AU = avuncular.

For example, the first value in the column 'MS' can be interpreted as 'if I were to pick two individuals born in the same year, and two individuals from my sample at random, how much more likely are the first pair to be maternal siblings, compared to the second pair?' Values below 1 indicate less likely, and values above 1 more likely. For MS, PS and AU absolute age differences are used (with overlapping generations, nephews may be older than their aunts), while parents and grandparents are necessarily older than their (grand-)offspring (categories M, P, MGM, PGF and MGF).

These age-difference based priors are by default automatically updated after parentage assignment and prior to sibship clustering, based on the empirical distribution of age differences between individuals and their assigned fathers and mothers. This update can be prevented with the option `Agepriors="old"`, or enforced later with the option `AgePriors="par"`.

As with the parameter specification file, feel free to alter the values in the AgePriors file to match the biological characteristics of the species, but please do not alter the number of columns. The number of rows may be increased (but not decreased below the age range amongst the genotyped individuals), in which case the entry 'nAgeClasses' in the file 'SequoiaSpecs.txt' should be updated to match the new number of rows.

3 Parentage assignment

Parentage assignment is performed with the option `Parentage = TRUE`, which will use the parameter settings in the file 'SequoiaSpecs.txt', created by `Prep = TRUE`. Parentage assignment is quick, and takes about 10–20 seconds for a dataset with 2,500 genotyped individuals on a laptop with an intel i7 2.3 GHz CPU and 8GB RAM.

Output The assigned parents are written to a text file (by default 'Parents_assigned.txt'), rather than returned within R. This file contains columns with

- ID of the individual, its assigned mother and assigned father;
- The log10 likelihood ratio (LLR) of the mother, father and the parent pair; this is the ratio between the likelihood of the assigned parent being the parent, versus the most likely alternative type of relative (e.g. full sibling or grandparent) or unrelated, to the focal individual (999.0 = missing value);
- The number of loci at which the offspring and the mother or father are opposite homozygotes (-9 = missing value);
- The row number in the genotype file of the offspring, mother and father; used when reading in this file for subsequent sibship clustering.

Some parents may have a very small or even negative single-parent LLR, but the LLR of the parent pair will always be positive, and is relative to the most likely assignment of a single parent. Note that the reported LLR differs from for example Cervus [Marshall et al., 1998], which returns the natural log of the ratio between the probability that the assigned parent is the parent, and that the next most likely candidate is the parent.

Non-assigned parent-offspring pairs In addition, the file 'Unassigned_relatives_par.txt' may be created, with identified but non-assigned parent-offspring pairs. These are non-assigned either because it was not possible to tell which of the two was the parent and which was the offspring, due to either or both individuals having an unknown birth year, or because it was not possible to tell whether the candidate parent was the mother or the father. These situations can be remedied by providing estimated birth years, or guessed genders, for the individuals involved in the life history input file, and re-running `sequoia`. The short computational time of parentage assignment means that this step can be repeated a number of times as a part of data quality control.

4 Sibship clustering

Sibship clustering amongst those individuals which have not been assigned a genotyped parent of each sex is performed with the option `Sibships = TRUE`, which will again use the parameter settings in the file 'SequoiaSpecs.txt'. Sibship clustering is considerably slower than parentage assignment, and may take from a few minutes to a few hours, depending on the number of individuals without a parent, the number of sibships that is being clustered, and their degree of interconnection.

Convergence is typically reached within five iterations, even for complex pedigrees. When convergence is reached before the user-set maximum number of iterations, a final iteration with stronger dependence on the age prior is ran, (dummy)parental likelihoods are calculated, and the algorithm is terminated.

Dummy names By default, dummy parents are denoted by increasing numbers, with prefix 'F' for females and 'M' for males. The prefixes can be altered in the 'SequoiaSpecs.txt' file, for example to avoid confusion with IDs of real individuals.

Output The output of the sibship clustering is by default written to a text file called 'PedSeq.txt', and which is highly similar to the output of the parentage assignment. This file does contain all assigned real parents, as well as the dummy parents assigned during sibship clustering. Dummy individuals are appended at the bottom of this pedigree file, with their assigned parents, i.e. the sibship's assigned grandparents.

In addition, a file with non-assigned pairs of relatives may be created, containing for example half-siblings were it could not be determined whether they are maternal or paternal half-siblings, and grandparents of singletons.

5 Comparison with previous pedigree

Often times, a (part) pedigree is already available to which one wants to compare the results. This pedigree may consist only of maternal links, deduced from observations in the field. The R function `PedCompare()` performs such comparisons, and takes as only arguments the file names of the 'true' pedigree and of the inferred pedigree.

References

TC Marshall, JBKE Slate, LEB Kruuk, and JM Pemberton. Statistical confidence for likelihood-based paternity inference in natural populations. *Molecular ecology*, 7(5):639–655, 1998.

S Purcell, B Neale, K Todd-Brown, L Thomas, MAR Ferreira, D Bender, J Maller, P Sklar, PIW De Bakker, MJ Daly, et al. Plink: a tool set for whole-genome association and population-based linkage analyses. *The American Journal of Human Genetics*, 81(3):559–575, 2007.