Pedigree checker manual

Jisca Huisman

2023-05-08

Description

This Fortran program takes a pedigree file and genotype data, and calculates for each offspring - dam - sire trio the probability that either or both parents are

- PO = parent - offspring,
- FS = full siblings (optional)
- GP = grandparent or full avuncular,
- HA = half avuncular or other 3rd degree relationship,
- UU = unrelated;

returning all 5x5=25 probabilities for each trio.

In fact, the 'pedigree' file may contain any arbitrary list of trios, such as offspring - sire - maternal grandsire. Each individual may occur multiple times in the offspring column, e.g. with different candidate parents.

Pairs can be included too, as simply trios where one 'parent' is 'NA'; probabilities are only calculated for this parent being unrelated (UU). Non-genotyped parents are treated the same as missing parents.

Assumptions

- Founders (individuals with no parents in the pedigree) are all unrelated;
- One 'parent' is related via the maternal side, and the other via the paternal side; the two 'parents' are unrelated (except when FS);
- Individuals can only be related as PO, FS, GP/FA/HS, HA, or UU. Inbred and double relationships are ignored;
- All SNPs are independent, have the same (average) genotyping error rate, and genotype frequencies are in Hardy-Weinberg equilibrium;
- None of the relationships are impossible based on age difference.

Warnings

- Under these simplifying assumptions, the likelihoods for the three types of second degree relatives (GP/FA/HS) are identical, as are the likelihoods for all types of third degree relatives (HA).
- Inbred parent-parent-offspring trios, where the two parents are themselves parent and offspring, violate the second assumption and `prob_PO_PO` in the output will be (much) less than one for correct pairs. Use sequoia to identify and/or check such pairs.
- There is no check whether two samples in a trio may come from the same individual; please use the sequoia R package with `Module=dup` or non-R sequoia with `--dup` first if duplicates may be included.

Input

SNP data

To convert genotype data from standard PLINK format to the format for `sequoia` and `pedigree_checker`, use

```
plink --bfile FileNameIN --recode A --out FileNameOUT
mv FileNameOUT.raw tmp.raw
cat tmp.raw | tr -s ' ' | cut -d ' ' -f2,7- > FileNameOUT.raw
sed -i '1d' FileNameOUT.raw
sed -i 's/NA/-9/g' FileNameOUT.raw
rm tmp.raw
```

which

- recodes to 1 column per SNP, coded as 0/1/2 copies of minor allele, missing = NA
- drops columns 2–6 (family ID, sex, phenotype, etc)
- drops header row
- replaces missing value code NA by -9

This is implemented in `format_SNP_data_for_sequoia.sh`:

```
./format_SNP_data_for_sequoia.sh FileNameIN
```

Pedigree / trios

The only other input file contains the trios of individuals. This can be a pedigree with columns id - dam - sire, but may for example also be id - sire - maternal grandsire. Column names should be maximum 7 characters long, longer names are shortened in the output. IDs may be maximum 40 characters long.

Execute

The code is compiled with

```
gfortran -std=f95 -fall-intrinsics -O4 pedigree_checker.f90 -o PedChecker
```

and run with

```
./PedChecker --trios PedigreeFile --geno FileNameOUT.raw --err 0.005 --out PedOUT_test.txt
```

an overview of the commands can be found with `./PedChecker --help`.

Calculations

For each trio, the probability of observing the offspring genotype is calculated, given the parents' genotypes and Mendelian inheritance laws, for each of the 16 possible relationships.

Parent genotype

The parents actual genotype, correcting for any genotyping errors and filling in any non-called SNPs, is estimated as the product of:

- the probability to draw 0,1, or 2 copies of the minor allele from the population at random, assuming HWE,
- the probability of observing the observed genotype, if the actual genotype were 0,1, or 2 copies.

The actual genotype probabilities are scaled to sum to one at each locus.

Note that this differs from the approach in sequoia, where a parent's parents' genotypes contribute to the probability. Here this is omitted, as otherwise an incorrect grandparent will confuse the interpretation of results.

Scaling

Following Bayes' theorem, the probability of a relationship R given the genetic data G can be calculated from the probability of G given R as

$$P(R|G) = \frac{P(G|R)}{P(G)} P(R)$$

Where $P(G)$ is the probability of observing the data ignoring the relationship, which for simplicity can be approximated by the probability if both parents where unrelated, $P(G) \approx P(G|UU)$.

The non-genetic probability of the relationship $P(R)$ includes in `sequoia` the age-difference based probability. Here, it is merely a scaling factor to ensure that $\sum_R P(R|G) = 1$.

Output from Fortran part

The output consists of the the trios in the pedigree file, with 19 or 24 added columns:

- OH_<parent1>, OH_<parent2>: the count of opposing homozygous SNPs between the focal (offspring) individual and , and between the focal individual and <parent2>. <parent1> and <parent2> are the column names of the 2nd and 3rd column in the input file. Missing value = -9
- ME_pair: number of Mendelian Errors between the focal individual and the parent pair. Missing value = -9
- prob_PO_PO : the probability that both the dam and the sire have a Parent-Offspring relationship with the focal individual. Missing value = 999.0000
- prob_.._.. : as for prob_PO_PO, where the first abbreviation denotes <parent1> and the second <parent2>, and
  - PO = parent - offspring
  - FS = full siblings (if not `--noFS`)
  - GP = grandparent or full avuncular
  - HA = half avuncular or other 3rd degree relationship
  - UU = unrelated.

With option `--LLR` log10-likelihoods (LL) are returned instead of probabilities, scaled for each trio by the LL that both parents are Unrelated, i.e. results are not scaled by $P(R)$.

Read output into R

To deal with the special missing values code, use the `na.strings` option of `read.table()`:

```
PedOUT <- read.table('Pedigree_OUT.txt', header=TRUE,
                     na.strings=c('NA', '-9', '999.0000', 'NaN'))
```

Disclaimer

This is a beta version.

While every effort has been made to ensure that this program provides what it claims to do, there is absolutely no guarantee that the results provided are correct. Use is entirely at your own risk.