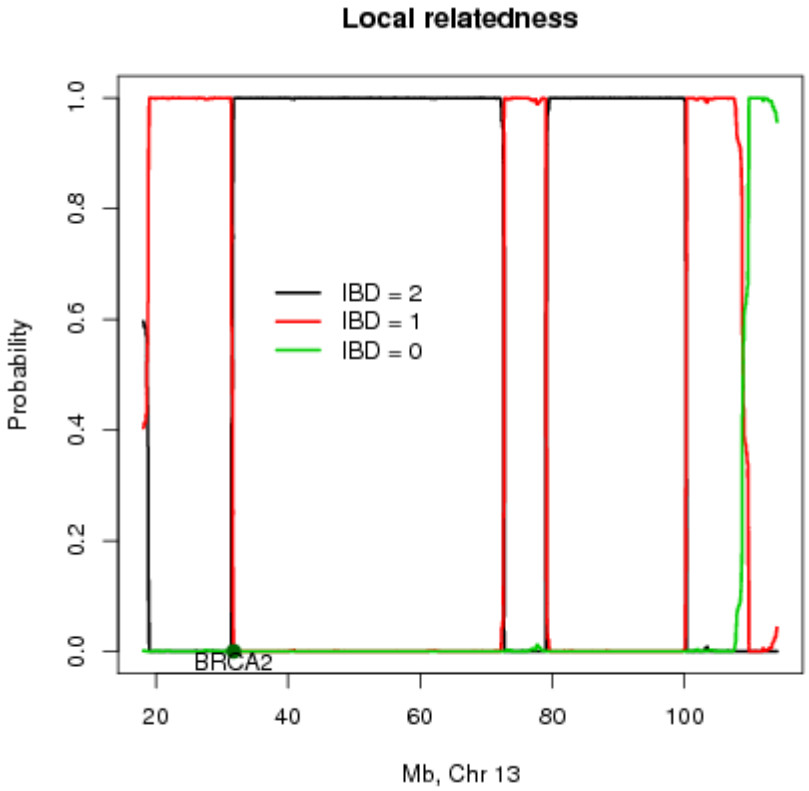
Manual for Relate v0.992
Software for estimating local relatedness across the genome and for performing linkage mapping for distantly related individuals with unknown pedigrees.

Anders Albrechtsen
albrecht@binf.ku.dk
Thorfinn Sand Kornliussen
thorfinn@binf.ku.dk

April 4, 2010



Contents

1 Introduction

Check for latest version of program and this manual at http://staff.pubhealth.ku.dk/~ande/web/software/relate.html.

This method estimates the probability of sharing alleles *identity by descent* (IBD) across the genome using single nucleotide polymorphism (SNP) data and can also be used for mapping disease loci using distantly related individuals. These individuals will often be seemingly unrelated but if they share the same founder mutation then they will be distantly related. The method is based on a continuous time Markov model with hidden states. The hidden states are the IBD states between two diploid chromosome pairs. We assume that the individuals are not inbreed and thus the individuals can share 0, 1 or 2 alleles IBD. The SNPs are allowed to be in linkage disequilibrium (LD). To accommodate LD the method needs SNPs from several individuals in order to estimate the allele frequencies and the pairwise LD. The method return the posterior probabilities of the IBD states across the genome and the overall IBD sharing. The estimates for all pairs of individuals can be combined into a score that will show linkage peaks across the genome and using a permutation procedure a significance threshold can be set. NB! only diploid chromosomes are used (Do not use X chromosomes unless only females are used).

2 Software

Relate is supplied as a R-package for both windows and unix. The program is licensed under the gpl licence http://www.gnu.org/licenses/gpl.html. And comes under no warranty.

The program can be used as a standalone c++ command line program without having R installed on the system.

Program options are changed via an options file for the command line version, or function arguments for the R version.

For performing the linkage analysis only the command line version is implimented.

2.0.1 Requirements - tested platforms

R should be installed, or ansi compliant c/c++ compilers. The program has been tested on all R versions from 2.6-2.8. and on gcc/mingw compilers 4.1.2 to 4.3.2 and on Intel compiler 11. This includes both 32bit and 64bit platforms.

2.1 Function overview

2.1.1 Using R

The following functions has been implemented in R, if you want to analyze all pairs of individuals, you should consider using the pure c++-version of the program.

getPed This function will read in data from a native snpMatrix object. snpMatrix is an R package available from bioconductor (www.bioconductor.org).

The function takes as argument the returnvalue from the snpMatrix function read.snps.pedfile

getPlink This function is used to read in the full data from the plink binary format. This should consist of 3 files with the following suffix. .bed, .bim and .fam files. Note that standard plink format is only used for human data. Thus if your sample has more than 22 autosomes then do not use the plink format.

runHmmld Run the main program with all function arguments.

runHmmld.plink Run the main program using the filenames of binary plink files. This will avoid having to read the entire plink binary file into R.

runHmmld.ped Run the main program using a snpMatrix object.

ld.snp3 This will calculate the ld's of a given matrix. But also the haplotype frequencies. This is a standalone version of the snpMatrix function *ld.snp*, that works for general R matrices. This method is implemented using the gsl method for root finding (http://www.gnu.org/software/gsl/).

plot R generic function used for plotting the result of a single-pair analysis from an object of the HMMrelate class.

print R generic function used for getting general info of a single-pair analysis from an object of the HMMrelate class.

In depth information of these functions is available through the R help system.

2.1.2 Using commandline (pure c++)

The commandline version can take 2 kinds of input data, either flat text files (described in section (3) and section 2.3) or binary plink files (see http://pngu.mgh.harvard.edu/~purcell/plink/). Note that we do not support all function when using the plink format.

The pure c-version has 3 possible ways of running.

1. Single-pair analysis: This is the same as the R version. Here only one pair of individuals are analysied. Set allpair to 0 in the option file.

2. All-pairs analysis: This is a fast method for running all combinations of individuals. Set allpair to 1 in the option file. This is much more efficient than running the single-pair analysis, since the haplotypes frequencies and pairwise ld's only has to be calculated once. Expect this cut down the runtime to around 20% compared with running the single-pair version consecutively.

3. Selected individuals: There is also a custom option where you can supply a file(with the -d flag) telling the program which individuals that you want to run the analysis on, this will be on all combinations. This is useful when you only want to analyse a smaller sample and only include other individuals to get better estimate of the haplotypes and allele frequencies. This method uses a variant of the all-pairs implementation, and is also faster than running the singl-pair analysis.

2.2 Installation

Get the latest version of the package on the website. Windows users should download the windows version and unix users the unix version. This is an R-package that is also used to build the commandline version of the program. The user however doesn't need to have R installed to build the commandline version.

The directory structure is quite simple. The "src" contains the c/c++ source files, the "data" subfolder contains some test data for both the R and C/c++ version, but also some example options files, only used by the commandline version. The rest of the folders "man" and "R" are specific for the R version.

2.2.1 Install package for R

To install the R-package for R using command line, you should go to the folder containing the R-package and type in

Installation for the R version

```
R CMD INSTALL Relate_version.tar.gz
```

If you don't have superuser privileges you may need to use `-l` option. And the *.libPaths()* from R accordingly.

Windows users can use the menu → Packages → Install packages from local zip file.

The program generates two warnings during installation, that are due to the fact that it's also a standalone program. These warnings are the options files in the data subfolder, and an installation script in the src subfolder.

2.2.2 Install package for command line

We will only describe how to install the package on a system that has the gnu toolchain installed or a build system that supports this toolchain. This means that you have to have a working '**make**' program installed. It is possible to install without. But this will not be explained in this manual in this version.

Unpack using your favorite unpack program. `tar xfvz Relate_version.tar.gz` or `unzip Relate_version.zip`. Goto the subdir of the unpacked file called src. from here rename Makefile_old to Makefile and type make. The program should now be compiled using the g++/gcc compilers.

Installation for the command line version

```
tar xfvz Relate_version.tar.gz
cd Relate/src
./install.sh
```

This will compile the program with the executables '**relateHMM**' and '**HMMtest**'. These will be located in the src subfolder.

If you however want to install with a different compiler e.g the Intel compilers you can specify these to the '**make**' command by using the following command.

From the src subdir in Relate

```
make CC=icpc C=icc
```

2.3 Usage

2.3.1 Using R

After starting R load the package using the library function:

Help in R

```
>library(Relate)
>help(package=''Relate'')
```

The R documentation can at anytime be accessed typing the above box so only the command line version will be explained here.

2.3.2 Using the commandline

The command should be written on one line, and has only been split up for improved readability. The order of the arguments doesn't matter. First is the program syntax for using flat textfiles.

command line options for flat files

```
./relateHMM -o options.txt -v verbose
            -g geno.txt -c chromo.txt -p pos.txt
            -post postoutput -k koutput
             -d individuals.txt
```

The "`-g -p`" parameters are requried, the rest are optional.

If you want to run the program using plink binary data you should use.

simple plink command line options

```
./relateHMM -o options.txt -v verbose
            -plink data
            -post postoutput -k koutput
             -d individuals.txt
```

NB! When the plink format is used the physical positions are used since the genetic positions are optional.

The program then assumes that the 3 files generated by plink are called `data.bim`, `data.bed` and `data.fam`. The program will also accept "-plink data.bim", or another one of the plink file-extensions. If however your plink files doesn't have the same name name you can specify with.

advanced plink options

```
./relateHMM -o options.txt -v verbose
            -plink-bed file1 -plink-bim file2
            -plink-fam file3 -post postoutput -k koutput
             -d individuals.txt
```

The program will not try to manipulate the filenames if all three are specified.

-g Genotypefile with individuals as rows and columns as SNPs. Delimiters can be ANY of {,;: } including tabs. Missing are denoted by NA or 0. see Section (3) data for more elaborate info.

-c Chromosome file. Delimiters can be ANY of {,;: } including tabs and newlines.

-p Positions file, Delimiters can be ANY of {,;: } including tabs and newlines. See Section (3) position for more elaborate info.

-o options file, this file should be newline separated and tab separated. An example can be seen in the appendix, the options will be explained in the next chapter.

-v Integer. Amount of info printed. 0 :no info will be printed, 1 some info will be printed, 2 even more info, mostly useful for debugging.

-post Filename used for dumping the posterior probabilities for the hidden IBD states. If the probability for sharing IBD k2 is set to zero then only k0(one row) will be dumped, since k1=1-k0. Otherwise two rows(k2,k1) will be dumped for each pair of individuals. Values of -1 denotes non-used SNP's, SNP's that haven't been used. If the filename exist a new file will be created by appending an integer if you are running all-pairs. Otherwise the data will be appended

-k Filename used for dumping the stationary probabilities for IBD sharing. If k2 value is set to zero, only 1 value will be dumped, otherwise 2. If a file exists with the given filename a new file will be created by appending an integer, if you are running all-pairs. Otherwise the data will be appended

-d A file containing individuals to test. Delimiters can be ANY of {,;: } including tabs. 1 indicates individiduals that should be pairwise tested. 0 indicates the individuals won't be tested. Program will then test all pairs in the same order as described in (section 6.0.1), this flag will override the allpairs option in the options file. This means that if the option file is set to run single-pair, the program will still run all-pairs.

-plink Generic plink filename. If a name with a known file extension ".bim,.bed,.fam" is supplied it will try to guess the other 2 filenames. That is, removing the suffix and appending the other ones. If the parameter given doesn't have a plink specific extension it will append ".bim,.bed,.fam", and try to use these.

-plink-bed The binary plink file.

-plink-bim This is not a compressed file but the first 6 columns of a ped file. The program will extract the first column for chromosomes and the 4th column for positions.

-plink-fam This is some ancestral scheme. NB! The pedigree information will not be used by the program. It is used solely for getting the number for individuals.

It's not possible to run the program with both flat files and plink files. The program simply stop if you try to.

2.4 Run examples

In this section we will present toy examples for running the '**relateHMM**' for more realistic example data see section 7.4. All necessary files are located in data subfolder of the package.

This should be run root folder
```
cd data
```

2.4.1 Examples using the program from the commandline using flat text files

First we will use the flat text files called `500.geno`, `500.pos` and `500.chr`.

This should be run from the data folder
```
../src/relateHMM -g 500.geno -p 500.pos -c 500.chr
 -o options.single.pair.txt  -post single.post -k single.k -v 0
```

The "-v 0" flag will make the program quite silent, if you want info printed out at runtime change to a larger value(integer). The program will generate two files called `single.post` and `single.k`. If these files exists the values will be appended to these. This is different from the all-pairs version, which will create new files.

If we instead want to run the allpairs version we will instead use the `options.all.pair.txt` file.

This should be run from the data folder
```
../src/relateHMM -g 500.geno -p 500.pos -c 500.chr -o
 options.all.pair.txt -post single.post -k single.k -v 0
...
-> Filename: single.post exists will instead use filename:single.post1
-> Filename: single.k exists will instead use filename:single.k1
->ind1:10ind2:11
```

The program tells us that it will use the filenames `single.post1`. If this file already existed it would have been called `single.post2`. Futher more standard output will tell you which individuals are being analysed.

We can analyse a subset of the individuals, these individuals are listed in the file called `500.d`, This file should contain the same number of individuals as the genotype file. That is the number of rows.

This should be run from the data folder
```
../src/relateHMM -g 500.geno -p 500.pos -c 500.chr
 -o options.single.pair.txt -d 500.d -post single.post
 -k single.k -v 0
```

```
...
-> Will run a specified joblist
-> number of pairs to run: 15
-> Filename: single.post exists will instead use filename:single.post2
-> Filename: single.k exists will instead use filename:single.k2
-> ind1:4        ind2:5  15/15
```

Notice that we use the `options.single.pair.txt`, the "-d" will override any all-pairs/single pair parameter. The output also tells us that we will perform 15 pair analysis's and also count how many single-pairs the program has performed. The individual id's are zero indexed. This means that the first individual will have id 0.

2.4.2 Examples using the program from the commandline using plink binary files

Plink binary files consists of 3 files a `.bed,.bim,.fam`, we will use the testfiles in the src subdir.

From the package root

```
cd data
```

Then execute this command on a single line

Example of command line execution of relateHMM

```
../src/relateHMM -o options.single.pair.txt -plink 500
-post test.post -k test.k -v 0
```

2.5 All-pairs

As mentioned in the short introduction to this section all-pairs is only implemented as a commandline version. Change the first line in the options file to 1. You also have the option to run multipairs. This is done by specifing with the -d parameter. Then the program will run all combinations of the pairs specified by the order in section (6.0.1).

2.5.1 Run time

The program is pretty CPU intensive and so far we have not used the software for more than 500 individuals. The run time is $O(mn^2)$, where n is the number of individuals and m is the number of markers. If the rate of change between states is estimated from the overall IBD sharing then the program runs much faster. If IBD=2 is not possible (or very unlikely) then fixing k2=0 improve run time greatly.

3 options

This section will describe the various options that can be used in the program.

data Integer Matrix (flat genotype file for the -g option). A matrix with SNP genotypes where NA or 0 denotes missing data, 1 for AA, 2 for Aa and 3 for aa. The number of individuals is the number of rows and the number of SNPs is the number of columns. (see table 1)

3	1	1	3	3
1	1	1	3	2
1	0	2	2	3
2	2	2	0	3

Table 1: example of the genotype file with four individuals each with five SNPs.

position The position of each SNP in centi Morgan (or mega bases). If centi Morgan is used then phi should be set to 0.01.

pair Integer vector of length two with the row numbers of the two individuals where relatedness is to be estimated. Only applies when all pairs are set to 0. Note that the first individual is denote 0 and the second individual is denote 1.

par Optional numeric vector c(a,k0,k1,k2) of parameters used instead of optimization.

min.maf The minimum minor allele frequency allowed.

LD The measure used to select the previous SNP to condition on, ("D'","D" or "rsq2").

epsilon The allelic error rate.

back The number of previous SNPs that can be conditioned on (see details for recommendations) .

alim The allowed range for the rate of change between IBD states.

start Optional starting point for the optimization. This only applies to the R version of the program.

prune The maximum value allowed for pairwise LD. If 0 then no pruning is performed.

ld_adj Logical. use the pairwise emission probabilities to correct for LD.

fix.a Numeric. Fix the a parameters to this value.

fix.k2 Numeric. Fix the k2 parameter to this value.

chr A vector of chromosomes numbers (only relevant when multiple chromosomes are used).

calc.a Estimate the a parameter from the overall IBD sharing. Appropriate for distantly related individuals or individuals who are related through one or two paths of the same length.

phi Numeric. The recombination rate in Morgans per Mega base (m/Mb).

timesToRun Integer. The maximum number of times the optimization is run.

timesToConverge Integer. The number of times the optimization should reach the same optimum.

giveCrap Integer. If non zero then emission probabilities (given the unobserved state and allele phase) and the haplotype probabilities are returned. Also more runtime information is given.

convTol Numeric. The tolerance for stating that the likelihood have reached the same likelihood.

back2 Integer. This is only included for debugging purposes. A value that determines the back internally in the program when running the fast implementation of all-pairs "-choose 0". If value is twice the value of "back", the program will produce the same results as running the slow version version of all-pairs "choose 1". So twice the value of back is"recommended", and is also the default.

4 Method

4.1 Parameters a, $k0$, $k1$, $k2$, ϵ(epsilon)

k0 The fraction of the genome where the pair of individuals share no chromosomes IBD (unrelated).

k1 The fraction of the genome where the pair of individuals share 1 chromosome IBD.

k2 The fraction of the genome where the pair of individuals share 2 chromosomes IBD.

a A parameter for the rate of change between IBD states.

ϵ The error rate. The method assumes that each allele has ϵ chance of being the other allele than the observed one.

4.2 The likelihood

The likelihood for the pairwise relatedness between individuals j and k assuming a first order Markov chain is

$$P(G^{j,l}|k,a) = \sum_{\mathbf{x}} \left(\prod_{i=0}^{m} P(G_i^{j,l}|X_i = \mathbf{x_i}) \right) \left(\prod_{i=1}^{m} P(X_i = \mathbf{x_i}|X_{i-1} = \mathbf{x_{i-1}}, k, a) \right) P(X_0 = \mathbf{x_0}|k) \tag{1}$$

where $k = \{k_0, k_1, k_2\}$, X_i is the IBD state of the ith marker, m is the number of markers, $G_i^{j,l} \in \Phi^2$ are the genotypes for individual j and l at position i and $\mathbf{x}$ is all possible IBD paths through the chain. This is generalized for multiple autosomal chromosomes by assuming an infinite distance between markers from different chromosomes.

The instantaneous rate matrix, Q is

$$Q = \begin{pmatrix} -ak_1 & ak_1 & 0 \\ ak_0 & -a(k_0+k_2) & ak_2 \\ 0 & ak_1 & -ak_1 \end{pmatrix}, \tag{2}$$

where the rows and columns corresponds to the states $IBD = 0$, $IBD = 1$ and $IBD = 2$ so the instantaneous rate of going from $IBD = 1$ to $IBD = 0$ is ak_0. The matrix is parametrized, so that the stationary distribution is given by the Jacquard coefficients and a is a parameter that decides how fast the chain changes states.

5 Details and recommendations

5.1 Choosing the number of markers to condition on (back)

The number of SNPs that LD extents over depends on the number of SNPs on the chip. If a low number of SNPs are used i.e. 10,000-50,000 then there won't be a lot of LD and you should choose a low number of SNPs to condition on – like back=5. We found that for 250,000 SNPs back=25 was enough and twice that for the 500k chips. However you should use the R package to investigate if LD is not accommodated sufficiently. To do that run the pairwise relatedness estimation for a couple of pairs of individuals. If the IBD pattern is erratic then choose a higher back. If the picture shows fewer changes in IBD pattern (like the picture on the first page of this manual) then LD is accommodated by the method.

5.2 Estimating a from the overall allele sharing (calculateA)

If the individuals are related through only one parent (or parental pair) then a can be estimated from the overall allele sharing. Therefore if the relationship between individuals is assumed to be simple (i.e. nuclear family or distantly related through only on individual) then this assumption will correct. We recommend using this assumption for better and faster convergence. The assumption can also be used to estimate the number of generations between individuals through the following relation

$$a = -Mlog(1-\phi), \tag{3}$$

where M is the number of generation (must be divided by two if the individuals are related through a parental pair). Note however, that if you use individuals selected based on their disease status (and they share a genetic variation IBD) then this estimation will be biased.

5.3 assuming k2=0 (fixK2)

If the individuals are distantly related then it is very unlikely that a pair of individuals will share two alleles IBD. Applying this assumption to the model is done by fixing k2 (the fraction of loci that share two alleles IBD) to zero. This is done by setting fixK2 to 1 and fixk2_value to zero.

5.4 Convergence

The method uses numeric optimization to get the maximum likelihood estimates of the parameters. In some instances the method does not find the global maximum because of multiple local maximum. Therefore the method is run using multiple starting points. The number of optimizations needed depends on the choice of parameters.

calc.a=TRUE,fix.k2=0 Here only one parameter is optimized (k0) and the method always finds the maximum. Choose times_to_converge=1, times_to_run=1. This will be the standard setting for performing linkage analysis using distantly related individuals.

calc.a=FALSE We recommend choosing times_to_converge=5, times_to_run=10, Using these setting most data will converge to the right maximum.

6 Output files

6.0.1 post

The outfile for the posterior probabilities estimated of the IBD sharing will be a matrix where the rows are each pair of individuals and each column is a SNP position. If the fix.k2=0 option is used then the post file will have $n(n-1)/2$ lines (one for each pair of individuals). If their are four individuals and fix.k2=0 then rows will belong to the pairs of individuals :

Individual 1	Individuals 2	posterior for IBD=0
Individual 1	Individuals 3	posterior for IBD=0
Individual 1	Individuals 4	posterior for IBD=0
Individual 2	Individuals 3	posterior for IBD=0
Individual 2	Individuals 3	posterior for IBD=0
Individual 3	Individuals 4	posterior for IBD=0

Table 2: the pairs of individuals with fixed k2

if k2 is not fixed then

Individual 1	Individuals 2	posterior for IBD=0
Individual 1	Individuals 2	posterior for IBD=1
Individual 1	Individuals 3	posterior for IBD=0
Individual 1	Individuals 3	posterior for IBD=1
Individual 1	Individuals 4	posterior for IBD=0
Individual 3	Individuals 4	posterior for IBD=1
Individual 2	Individuals 3	posterior for IBD=0
Individual 2	Individuals 3	posterior for IBD=1
Individual 2	Individuals 4	posterior for IBD=0
Individual 2	Individuals 4	posterior for IBD=1
Individual 3	Individuals 4	posterior for IBD=0
Individual 3	Individuals 4	posterior for IBD=1

Table 3: the pairs of individuals without fixed k2

6.0.2 k

Just like the posterior output for the local IBD sharing the output for the overall IBD sharing k will have n(n-1)/2 lines if k2 is fixed - one for each pair of individuals. If k2 is not fixed then each pair will have two lines.

7 Linkage analysis

Linkage mapping can be performed using case control information for each individual. The testing is performed using a permutation procedure to obtain p-values for whether the cases are more related in a region than controls. This provides a local p-value without any correction for multiple testing, and a global p-value that tests if any region is more related within cases than within controls (thus a value that is corrected for multiple testing).

7.1 Installation of HMMtest

The program is installed per default if you used the method described in section (2.2.2).

7.2 Input data and options

The input and output from the HMMrelate all pairs should be used for the HMM test program. Also a file that specifies cases and controls.

Posterior Matrix (-P) This is the posterior distribution for being related a loci as described in section 7.3 where each row is a pair of individuals and each column is a SNP position.

Position file (-p) This is a list of positions along the genome as used as input for the relateHMM program.

disease vector (-D) A file containing a description of the individuals disease status where 0 defines controls and 1 defines cases. If denotes as -1 then this individuals is excluded from the analysis. Delimiters can be ANY of {,;: } including tabs.

Number of permutations (-Nsim) The number of permutations used to estimate a p-value.

case control design (-ccALL) A number that defines which design is used for the case control study. If 0 then within cases IBD are compared with within controls IBD, 1 within controls IBD are compared to within cases IBD and between cases and controls IBD, 2 within cases IBD are compared to whitin controls IBD and between cases and controls IBD.

missing genotypes (-infer) 0 The missing IBD probabilities due to missing genotypes are removed from the analysis (The non-missing IBD probabilities for the same SNP position are not removed). 1 (recommended) the missing IBD probabilities are inferred from the adjacent IBD probabilities.

covariance matrix (-c) The filename of the output file for the estimated covariance matrix. If a covariance matrix of the same name exists this will be used. (only applied if the -c option is used).

Local statistics (-file) The output file for the statistic at every loci.

Max statistic (-Xmax) The maximum statistic for every permutation including the unpermuted one (the first one).

infered posterior (-o) The name of the output file for posterior probability matrix with the inferred probabilities included.

significance (-sig) Threshold for the significance. The program will provide a threshold for the maximum statistics such that all values higher than this number are significant.

7.3 Output

7.3.1 Output to screen

pmaxsim This is the simulated p-value, adjusted for multiple testing, for the highest linkage peak. Note that the p-value can never be lower than one divided by the number of permutations.

psigsim This is the significance threshold for the statistics at the level specified (-sig). Any position with a statistics higher than this value is significant after correction for multiple testing.

7.3.2 Output files

Local statistics (-file) A file with the statistic for each position.

Max statistics (-Xmax) A file with the maximum statistics for each permutation.

covariance matrix (-c) A file with the covariance matrix. Creating this matrix is the most times intensive part of the testing and the matrix will can reused by the program if specified.

infered posterior (-o) A file with the posterior probabilities where the missing data have been infered based on the adjacent sites.

7.4 Run example for linkage analysis

In this section we will present a complete commandline example of running 'relateHMM' and the 'HMMtest' program, we will presume both are installed in the scr subdir and that the testfiles are in the data subfolder of the package.

From the package root folder

```
cd data
```

For a linkage analysis we want to run all pairs so after inspecting the options file so that all pairs is set to 1, calculateA is set to 1 and fixK2 is set to 1. We will run the program.

in the data subfolder

```
../src/relateHMM -o options.linkage.txt -file dat_small -v 0
 -d dat_small.keep -post dat_small.post -k dat_small.output
```

The "-v 0" still tells the program only to give basic information during runtime. Such as which individuals are being analysied. After the program has finished we want to run HMMtest on the output files `post.outpu` and `k.output`).

Example of HMMtest

```
../src/HMMtest -P dat_small.post -D dat_small.cc -infer 1
-p stripped.pos -Nsim 1000
```

To plot the results use the Rscript command

Example of HMMtest plot

```
Rscript linkagePlot.R perm=x.txt stat=stat.txt pos=stripped.pos
chr=stripped.chr alpha=0.01,0.05,0.001
```

this will create a pdf file

see other plotting options by typing

Example of HMMtest plot

```
Rscript linkagePlot.R
```

8 Contact and bugreporting

The authors can be contacted by email **albrecht@binf.ku.dk** and **thorfinn@binf.ku.dk**. Please provide your operating system, compiler version and R version, the Relate version, and of cause a small description concerning your problem.

Information to include

```
uname -a
gcc --version
R --version
```

A Option file for HMMrelate

```
1 #1=allpairs 0=normal run
8 #pair[0]
9 #pair[1]
0 #double recombination
0 #LD=0=rsq2 LD=1=D //allright everythings gone be allright
0.025 #   min
0.001 # alim[0]
5.0 # alim[1]
0 #doParameter calculation (pars)
0.3 # par[0] = a  this is only used if doParameter is set to 1
0.25 # par[1] = k2 this is only used if doParameter is set to 1
0.5 # par[2] = k1 this is only used if doParameter is set to 1
```

```
1 #ld_adj
0.01 #epsilon
50 #back
0 #doPrune
0.1 #prune_value
0 #fixA
0.0 #fixA_value
1 #fixK2
0.0 #fixk2_value
1 #calculateA
0.013 #phi_value
0.1 #convergence_tolerance
3 #times_to_converge
8 #times_to_run
100 #back2
```