LDhat: A package for the population genetic analysis of recombination

Gil McVean
mcvean@stats.ox.ac.uk

www.stats.ox.ac.uk/~mcvean/LDhat.html

Department of Statistics, Oxford, OX1 3TG, UK

Last updated: July 22, 2004

Contents

1 Introduction

LDhat is a package of programs written in the C language for the analysis of recombination from population genetic data. The key feature of the package is the estimation of population recombination rates using the composite likelihood method of Hudson [6], adapted to finite-sites models (applicable to diverse genomes such as those of some viruses and bacteria) [9] and to estimate variable recombination rates [10]. The method accommodates both phased or haplotype and unphased or genotype data, with arbitrary levels of missing data.

Within the package there are a number of programs. Brief descriptions of the programs are given below.

- **convert**. A simple manipulation program that generates files in the appropriate format for subsequent analyses, allows a number of options to be selected (e.g. minimum allele frequency cutoff, missing-data cutoff, inclusion/exclusion of sites with 3 or more alleles, data subsampling), and summarises the input data. The files generated by the program are *sites*, which contains the sequence data, and *locs*, which contains the physical location of SNPs.

- **pairwise**. Estimates the population recombination rate for the region analysed assuming a constant recombination rate over the region and either a crossing-over or gene conversion model. Takes as input the *sites* and *locs* files generated by convert (or the user) and (usually) a lookup table for the appropriate number of sequences, theta per site, and grid size. For haploid or phased data with no missing information **pairwise** will, if prompted, generate a minimal lookup table within the program. This option allows efficient analysis of small data sets. However, other options within the program (such as conditional simulation - see below) require the exhaustive lookup file generated by **complete** or **lkgen**.

- **interval**. Estimates a variable recombination rate using a Bayesian reversible-jump MCMC scheme [3] under the crossing-over model (only). As for **pairwise**, the *sites*, *locs*, and lookup table files are required.

- **stat**. Summarises the output from **interval**.

- **complete**. Generates a lookup table required for all analyses (except under certain circumstances, see **pairwise**). Input number of sequences, theta per site (assumes a two-allele symmetric mutation model), and the grid size for $4N_e r$ (in terms of maximum value and number of points in the grid). The program calculates the coalescent likelihood of all possible two-locus haplotype configurations using the importance sampling method of Fearnhead and Donnelly [1].

- **lkgen**. Generate a lookup table from an existing one - for a smaller number of sequences than the existing one, with the same theta per site and grid size. Because of the computational cost of calculating the lookup table, pre-computed tables for a range of parameter values are available from `www.stats.ox.ac.uk/~mcvean/LDhat.html`

2 Installation

The LDhat package is available as freeware from

`www.stats.ox.ac.uk/~mcvean/LDhat.html`

C source code is provided as a tarred, gzipped folder, LDhat.tar.gz for compilation using gcc on a Unix or Linux operating system (or under a proprietary C compiler). To install, type at the command prompt (indicated by the % sign)

```
% gunzip LDhat.tar.gz
% tar -xvf LDhat.tar
```

Move to the LDhat directory and type `make`. To remove intermediate files, subsequently type `make clean`.

3 Input format

The LDhat package accepts data in two formats: full sequence data, or SNP surveys. Full sequence data should be aligned and in a modified FASTA format, with the first line detailing the number of sequences/genotypes, the number of sites in the alignment and a flag (1 or 2) that details whether the data is haplotype/phased (1) or genotype/unphased (2). If full sequence data is used, **convert** should be used to generate the *sites* and *locs* files needed for subsequent analyses (these can be renamed). Sequence names should be no longer than 30 characters long, and data should have no more than 2000 characters per line (line breaks should be used to break up larger data sets).

The alternative format is to input segregating positions only, in which case two files (referred to as the *sites* and *locs* files) are required. The format of the *sites* file is identical to that for the full-sequence format (see example below). The *locs* file has on the first line the number of SNPs (segregating sites), the total length of the region analysed, and a flag (L or C) that details whether a model of crossing-over or gene conversion is fitted (see [9]). Following the header line, the file contains the relative or absolute location of SNPs/segregating sites in increasing order. For large SNP surveys it is recommended that positions are encoded in kb rather than bp.

Haplotype/phased sequence data can either be coded in as standard DNA letters (including ambiguous bases, N or ? for missing data and - for gaps), or as 0/1/2/3. For genotype/unphased data, the convention used is 0 and 1 for the two homozygotes, and 2 for heterozygotes. For computational efficiency, all subsequent analyses are restricted to those sites that have 2 alleles segregating (i.e. monomorphic and sites with 3 or more alleles are excluded).

Example *sites* file format for haplotype full sequence data.

```
4  10  1
>SampleA
TCCGC??RTT
>SampleB
```

```
TACGC??GTA
>SampleC
TC?-CTTGTA
>SampleD
TCC-CTTGTT
```

Example *sites* and *locs* file format for genotype data.

```
**Sites file**
 4 10 2
>GenotypeA
122110?000
>GenotypeB
1111201100
>GenotypeC
011111?112
>GenotypeD
2112210100

**locs file**
10  1200 L
1 57 180 187 223 250 438 509 878 1034
```

An example data set, *lpl.sites* and *lpl.locs*, is included in the package.

4 Using the programs

This section provides a detailed description of the input and output generated by each program and the options available within them. Where command-line arguments are given, arguments in square parentheses indicate options. Due to computational load, at present there is a limit of 300 sequences that can be analysed within the program. If more sequences are included, only the first 300 will be analysed.

4.1 convert

To run **convert** from the command line type

```
% ./convert [input_file] [locs_file]
```

If no file is specified, the user is prompted for the input file name within the program. If the name of a *locs* file is also included on the command-line, the program will use this to provide the relative position of each SNP (this option cannot be accessed from within the program), otherwise it is assumed that sites are contiguous.

The program reads in the input file(s), and produces a summary file *freqs.txt* that details the frequency of each allele at each site. Next, the user is prompted

to choose the subset of SNPs to output for further analysis. Options available are

- Analyze all SNPs or those with 2 alleles? (1/2). Although only those sites with 2-alleles are analysed in **pairwise** and **interval**, outputting all segregating sites (option 1) may be of interest and can be used to estimate a finite-sites estimate of Watterson's theta per site [9]. If the option for only those sites with two-alleles is selected (option 2), the user can specify a minimum minor allele frequency (MAF) cutoff (e.g. 0.2 specifies those sites with a MAF$>$ 20%). If 0 is inputted, no MAF cutoff is implemented.
- Frequency cut-off for missing data. Sites with large amounts of missing data slow subsequent analyses. Specify the frequency of missing data above which sites are excluded from subsequent analyses (e.g. 0.2 specifies 20%).
- Print all sites? (1/0). To print only a subset of the data select option 0, and specify the start and end position (or SNP) to output.

The program ends by summarising the input data with four statistics. These are

- The number of segregating sites
- The average pairwise differences (PWD) between sequences
- Watterson's infinite-sites estimator of the population-scaled mutation rate $\theta = 4N_e\mu$ for diploid species, or $\theta = 2N_e\mu$ for haploid species. Note that this estimate applies to the whole region (i.e. is not per site), **pairwise** will also estimate a theta, but this is a *per site* estimate based on a finite-sites approximation to Watterson's estimate [9].
- Tajima's D statistic [14]. A measure of departure from neutral expectation. Significance levels under the assumption of no recombination can be found in Tajima's paper [14]. Significance levels with recombination can be estimated by Monte-Carlo simulation, e.g. using Hudson's makesamples program [7], or DNAsp [13].
- Fu and Li's D* statistic [2]. Another summary of the departure of the frequency spectrum from neutral expectations, that focuses on the number of singletons (sites at which only a single sequence differs from the rest of the sample).

The program generates two files, *sites* and *locs*, which may be renamed and used in subsequent analyses. These contain the data at the sites selected and the position of the sites.

4.2 pairwise

Performs estimation of the population-scaled recombination rate $\rho = 4N_e r$ for diploid species, or $\rho = 2N_e r$ for haploid species, where N_e is the effective population size and r is the genetic map distance across the region analysed (the product of the physical distance and the per site rate of recombination across

the region). The composite likelihood method of Hudson [6] is used, although in contrast to Hudson, a finite-sites model is used to estimate the coalescent likelihood of two-locus haplotype configurations [9]. The use of a finite-sites model enables the use of the composite-likelihood method in species, such as viruses and bacteria, where sites may have experienced multiple mutations in the history of the sample; see McVean et al. [9] for more details. It should be noted that the model also assumes uniformity of the mutation rate across sites. Extreme rate heterogeneity can cause homoplasy that mimics the effects of recombination. For this reason, in highly diverse genomes, analysis of codon positions 1 and 2 apart from position 3 is recommended. However, the nonparametric tests employed in the program to test the hypothesis of no recombination are robust to rate heterogeneity and complex mutation models [9].

To run the program from the command line type

```
% ./pairwise [input_file] [locs_file] [lookup_table_file] [1]
```

If no file-names are specified, the user will be prompted within the program. If no lookup table is available, one specific to the data set will be generated within the program. The option [1] sets the program to concise mode (reduces the amount of output files). The various options within the program are

- Use an existing likelihood file? If yes (option 1), the name of the lookup table should be inputted followed by whether the lookup table is exact (specific to the data set, including missing data and unphased/genotype information) or not. Otherwise (option 0), the program will ask the user to input parameters to estimate the lookup table specific to the data. In order, these are: A) The theta per site to use (with a suggestion made from the finite-sites version of Watterson's estimate); B) The maximum value of $4N_e r$ (equivalent to $2N_e r$ for haploid organisms) for the grid, with a suggested value of 100; and C) The number of points on the grid (with a suggestion of 101). The choice of parameters for the grid (which is uniformly spaced) specifies the accuracy and computational load of the calculation; large values for the maximum $4N_e r$ and points will take longer and give more accurate results. However, it is suggested that the max $4N_e r$ value should be in the range 20-100 (values greater than 100 are treated as 100), and the number of points should be in the range 21-201. The estimation of the lookup table may take up to a day on a standard desktop. It is therefore recommended that wherever possible existing lookup tables that include all possible two-locus haplotype configurations are used or generated from existing ones (see **lkgen**). Furthermore, for unphased/genotype data, and data sets with missing data, such an exhaustive lookup table is required. It is worth noting that minor changes in the value of theta per site used do not seem to have a large influence on the estimated recombination rate.

- Estimation of the recombination rate. Once the lookup table has been generated/read in, the user will be prompted for whether to change the default values for the grid over which the population recombination rate (for the whole region) should be changed (option 1 = yes, option 0 = no). Initially, the defaults should be accepted, but if the estimated value is at the extreme of the grid, subsequent analyses should change the defaults. If

the grid over which likelihoods have been calculated is large and fine (e.g. max $4N_e r = 100$, no. points = 101), there is no restriction in estimating rates over the region analysed which extend into the 1000s. If possible, it is worth having in mind a rough figure for the value you expect over the region. Generates a file *outfile.txt*.

- Calculating fits (no user prompting - see below for details). Generates a file *fits.txt*.

- Sliding windows analyses. An equivalent estimation procedure can be carried out in a sliding-windows fashion to look at recombination rate variation in the region (though this analysis is largely superseded by **interval** (see below). If this option is selected (option 1), the user is prompted for the number and length of the windows (the degree of overlapping is determined by these options). Generates a file *window_out.txt*.

- Rmin values. Uses Hudson and Kaplan's estimator of the minimum number of recombination events [8] to describe the evidence for recombination across the region. Option 1 simply prints to screen the minimum over the region (which is a lower bound on the true minimum in the infinite-sites model). Option 2 prints both the total, but also generates a file *rmin.txt*, which details the estimated minimum between all pairs of SNPs using the dynamic programming algorithm of Myers and Griffiths [11].

- Moment method. Uses Wakeley's [15] adaptation of Hudson's [4] moment method to estimate the population-scaled recombination rate across the region from the variance in the pairwise differences between sequences.

- Test for recombination. Nonparametric permutation tests for the influence of recombination. The distribution of four statistics of the data (maximum composite likelihood, summed difference between all pairs of sites that show evidence of recombination by the four-gamete test, correlation between the r^2 measure of linkage disequilibrium (LD) and physical distance and correlation between the $|D'|$ measure of LD and physical distance) are calculated under random permutation of the physical position of the SNPs (1000 permutations). The values observed in the unpermuted data are compared to these null distributions to describe the evidence for a non-zero rate of recombination. The power and properties of these tests are described in [9]. Currently, these tests are only available for phased/haplotype data. Generates a file *rdist.txt*.

- Parametric simulation. Monte Carlo coalescent simulations are carried out using the estimated population recombination rate and the inputted value of theta per site, which condition on the location and approximate allele frequency of SNPs in the sampled data. These simulations are used to generate the sampling distribution of the point estimate of the population-scaled recombination rate and to test the hypothesis of a constant recombination rate over the region. If simulations are to be performed, an exhaustive lookup table must be used. Currently only available for phased/haplotype data. Generates a file *sim_out.txt*.

A brief description of the contents of the output files.

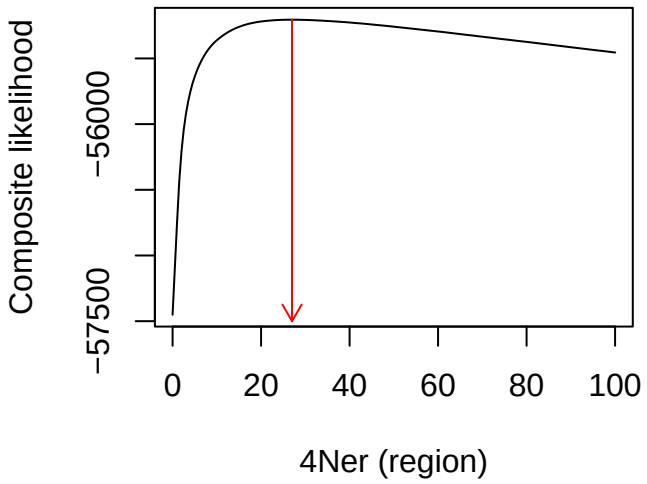

Figure 1: Composite-likelihood curve for the LPL data set from Finnland (48 haplotypes) [12]. The maximum occurs at $4N_e r = 27$ for the 9.7 kb region

- *outfile.txt* contains the point estimate of the population recombination rate for the region and the composite likelihood surface. The likelihood surface for the sample data set is shown in Figure 1.
- *new_lk.txt* contains the lookup table specific to the data set. This should be renamed and used in future analyses, e.g. for **interval**.
- *fits.txt* contains a upper-diagonal matrix with the marginal likelihood-ratio test statistic for each pair of sites. For each pair of SNPs, i and j, the reported value represents the difference in log composite-likelihood between the best-fitting model that assumes a constant recombination rate over the entire region, and the marginal maximum for the pair of sites. For example, suppose there is a 1kb region for which the maximum composite likelihood for the entire data is achieved at $4N_e r = 100$. A pair of SNPs at 250bp and 750bp would therefore have a recombination distance of $4N_e r = 50$, but if we were looking at just those sites, maybe the likelihood would be maximised at $4N_e r = 10$. The difference in log likelihood between these estimates is then reported, multiplied by $+1$ if the marginal maximum is achieved at a higher $4N_e r$ than the estimated value or -1 if the marginal maximum is achieved at a lower $4N_e r$ than the estimated value. Also calculated is the sum of the composite log likelihood ratio test statistics across pairs. The matrix can be used to visualise the discordance between the fitted (constant) rate patterns in the data; see Figure 2.

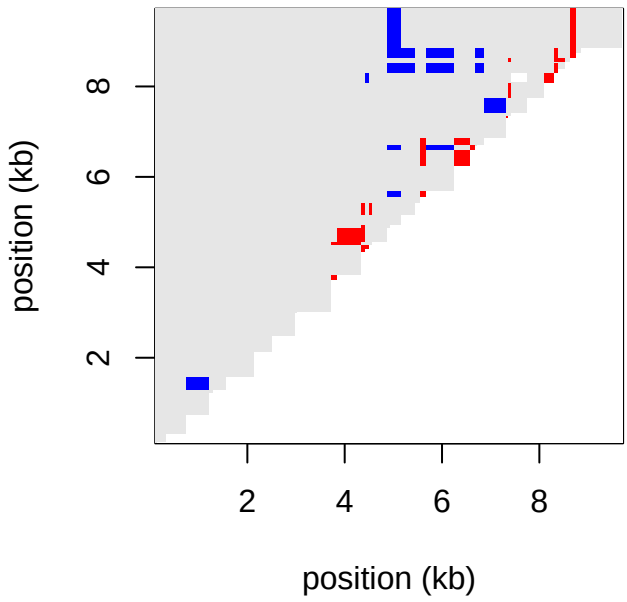

Figure 2: For the LPL data, a plot showing which pairs of sites have unusual patterns of LD given the assumption of a constant recombination rate. Pairs with a marginal likelihood ratio greater than 2.0 are coloured (red for pairs with less LD than expected, blue for pairs with more LD than expected, gray for other pairs). The block of red in the centre suggests the presence of a recombination hotspot

- *window_out.txt* contains the output of the sliding-window analysis. The columns are (in order): the position of the left-most SNP in the window, the position of the right-most SNP in the window, the number of SNPs in the window, the window estimate of the recombination rate per bp or kb (depending on whether the *locs* file is in bp or kb), the log composite likelihood ratio comparing the locally estimated rate to the rate estimated for the whole region, and Tajima's D statistic for the window. The log composite likelihood ratio is an indicator of the degree to which the window differs from the region as a whole, but should not be used as a standard likelihood ratio test statistic.

- *rmin.txt* contains the output from the Rmin analysis. For each pair of sites i and j ($j > i$), the upper diagonal element (i, j) is the minimum number of recombination events that occurred in the history of the sample between these positions, and the lower diagonal element (j, i) is that value divided by the physical distance between the SNPs. This matrix can also be used to derive graphical representations of the extent of recombination in the sequence, see Figure 3.

- *rdist.txt* contains the results of the permutation analyses designed to test for the presence of recombination. These data can be used to compare the distribution of the test statistic against the observed value.

- *sim_out.txt* contains details of the parametric simulations carried out to test for recombination-rate uniformity and calculate the sampling distribution of the point-estimate of the recombination rate. The columns in the file are respectively, the point estimate of rho from the simulation, the summed marginal log composite likelihood test statistics across pairs of SNPs (see above), the difference in log composite-likelihood between the maximum and that obtained at the value the data was simulated under (for assessing the null distribution of the log composite likelihood ratio test statistic), and a correction factor (CF), which is proportional to the log of the probability of the simulated data under the coalescent model. The final column could be used to weight simulations, however the test is carried out in the manner of Hudson [5], where all simulations are given equal weight.

4.3 interval

Performs estimation of variable recombination rates using a penalised likelihood within a Bayesian reversible-jump Markov chain Monte Carlo scheme (RJMCMC). Note, however, that because of the assumptions introduced by the use of composite-likelihood, the standard Bayesian interpretation of the output of the chain cannot be said to represent the posterior distribution of rate estimates. Rather, a summary of the output should be used to describe the analysis of the data, and comparison to simulations should be used to draw statistical inferences; see [10] for details of the method.

The program requires the same input files as **pairwise**, except that a lookup table is essential. To run it from the command line, type

```
% ./pairwise [input_file] [locs_file] [lookup_table_file] [1]
```

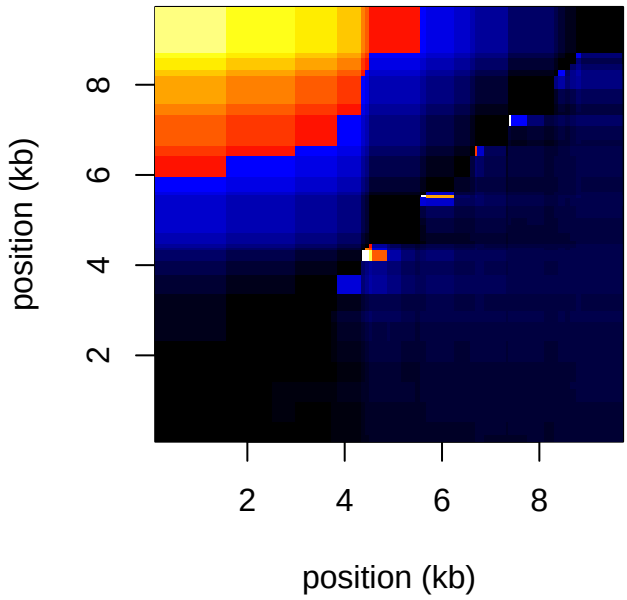

Figure 3: Detectable recombination events in the LPL gene (Finnish population) using the method of Hudson and Kaplan [8]. Above the diagonal, values represent the minimum number of detected recombination events between pairs of sites. Below the diagonal, the minimum number of recombination events is rescaled by the physical distance between pairs of sites to represent the local intensity of recombination. The cluster of recombination events in the centre of the gene again suggest a recombination hotspot

If no file-names are specified, the user will be prompted within the program. The option [1] sets the program to concise mode (reduces the amount of output files). The various options within the program are

- Is likelihood file exact? As with **pairwise** lookup tables can either be exhaustive (i.e. contain every possible two-locus haplotype configuration with no missing data for the number of sequences as generated by **complete** or **lkgen**), or exact (contain all two-locus configurations for the specific data set including missing-data and genotype data), as generated by **pairwise**.

- Starting value for $4N_e r$ ($2N_e r$ for haploid species). The user must specify an initial starting point for the RJMCMC scheme, which is achieved through assuming a constant recombination rate across the region analysed. The user should ideally have an idea of the approximate value expected (e.g. from a knowledge of genetic map distance and effective population size of the species). For example, in humans a starting point is the number of kb times 0.4 (assuming a 1cM/Mb recombination rate and effective population size of 10^4).

- Block penalty. The method works by fitting a piece-wise constant recombination rate to the data, where changes in rate can only occur at SNPs. To avoid over-fitting, a penalty is introduced, which is multiplied by the number of change-points in the estimate genetic map and added to the log composite likelihood. Large penalties identify positions at which there is very strong evidence for changes in recombination rate, but will loose detail. Small penalties reveal detail, but also introduce noise. Calibration of the method by simulation has shown that a penalty of 20 is adequate for the analysis of large data sets in humans, but it is recommended that the user uses a series of penalties in the range 0-50. Monte Carlo simulations can potentially be performed (not available within the LDhat package) to assess the performance of the method under different conditions; see [10] for full details.

- Number of updates for MCMC. The Markov Chain can be thought of a random walk through parameter space that biases towards parameter values that have high likelihood (and prior weight). To sample the space adequately it is necessary to run the chain an indefinite amount of time. For practical purposes, it is recommended that no few than a million iterations should be performed, with some longer runs, and runs at different starting points to check for convergence.

- Number of updates between samples. For efficiency, it is often preferable not to keep the results from every iteration of the Markov chain. It is recommended that sampling every 2000-5000 iterations is adequate.

Once initiated, the Markov chain is left to run. At each sample, a summary is printed to screen that details the iteration number, the current likelihood of the data (excluding change-point penalties), the number of blocks with uniform recombination rate (the number of change points plus one) and the total population recombination rate over the region analysed (map length). Efficient mixing of the algorithm can be observed if the summaries change frequently.

Two files are generated as the Markov Chain progresses. *rates.txt* is the output from each sample detailing the recombination rate (expressed in $4N_e r$ per kb or bp depending on the format of the *locs* file) between each SNP, and the corresponding likelihood of the data. *bounds.txt* details the position of the change points along the sequence at each sample. The program **stat** should be used to summarise these two files.

At the end of the run, acceptance rates for each of the proposed moves in the RJMCMC scheme are detailed. Because of the composite-likelihood scheme, these rates are often very low (of the order of 1% or less), indicating that the chain needs to be run for a long period of time for appropriate sampling.

Also generated is a file *new_lk.txt*. As for **pairwise**, this file should be renamed and used in future analyses with **interval** as an exact lookup table. Note however, that this file cannot be used for **pairwise**.

4.4 stat

Summarises the output from **interval** in terms of the average, median, 2.5th percentile and 97.5th percentile of the estimated recombination rate between each pair of SNPs. To run the program from the command-line type

```
% ./stat [input_file] [burn_in]
```

If no file is specified, the user will be asked to input the file name. Files should be of the format of *rates.txt* and *bounds.txt*, with the first line detailing the number of samples and the number of elements at each sample. The burn-in specifies the number of entries to be ignored in the summary. It is recommended that the first 100,000 iterations of the RJMCMC scheme are excluded. The output file, *res.txt*, is a summary of the samples from the RJMCMC chain. An example of the summary is shown in Figure 4 for two different values of the penalty (5 and 20). Note that the first entry in the *rates.txt* file corresponds to the total population genetic distance across the region (the map length). Visual inspection of the estimates from the RJMCMC should be used to check for convergence, along with multiple runs starting from different points (see Figure 5).

4.5 complete

Generates a lookup table with the coalescent likelihoods for every possible two-locus haplotype configuration for a given sample size, using a user-defined theta per site and a grid of recombination rate values. The importance sampling method of Fearnhead and Donnelly [1] is used. Note, that because the importance sampling method is Monte-Carlo (i.e. simulation-based), different runs of the program may give very slightly different results.

To use the program, type (at the command-line)

```
% ./complete
```

The user is prompted for the number of sequences, theta per site and grid properties (see **pairwise** for a discussion of the grid). The file generated *new_lk.txt* should be renamed before further analyses.

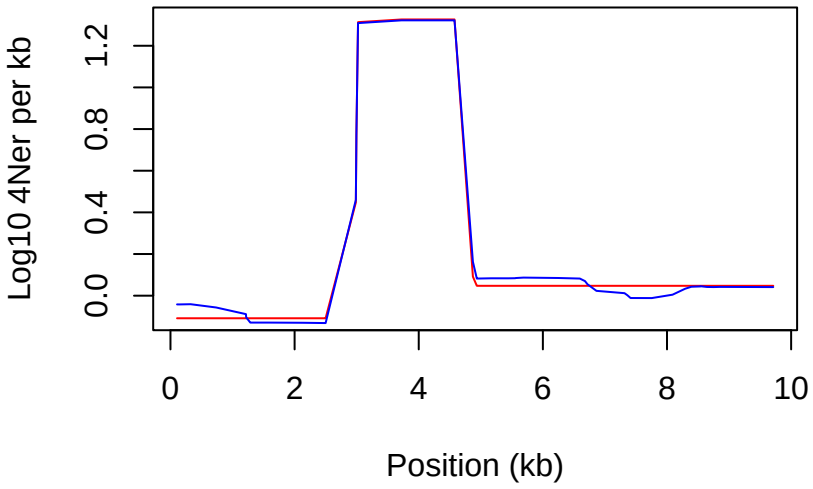

Figure 4: The estimated recombination rate across the LPL gene from the Finnish sample for penalties of 5 (blue) and 20 (red). The point estimate of the rate is given by the average rate from the posterior

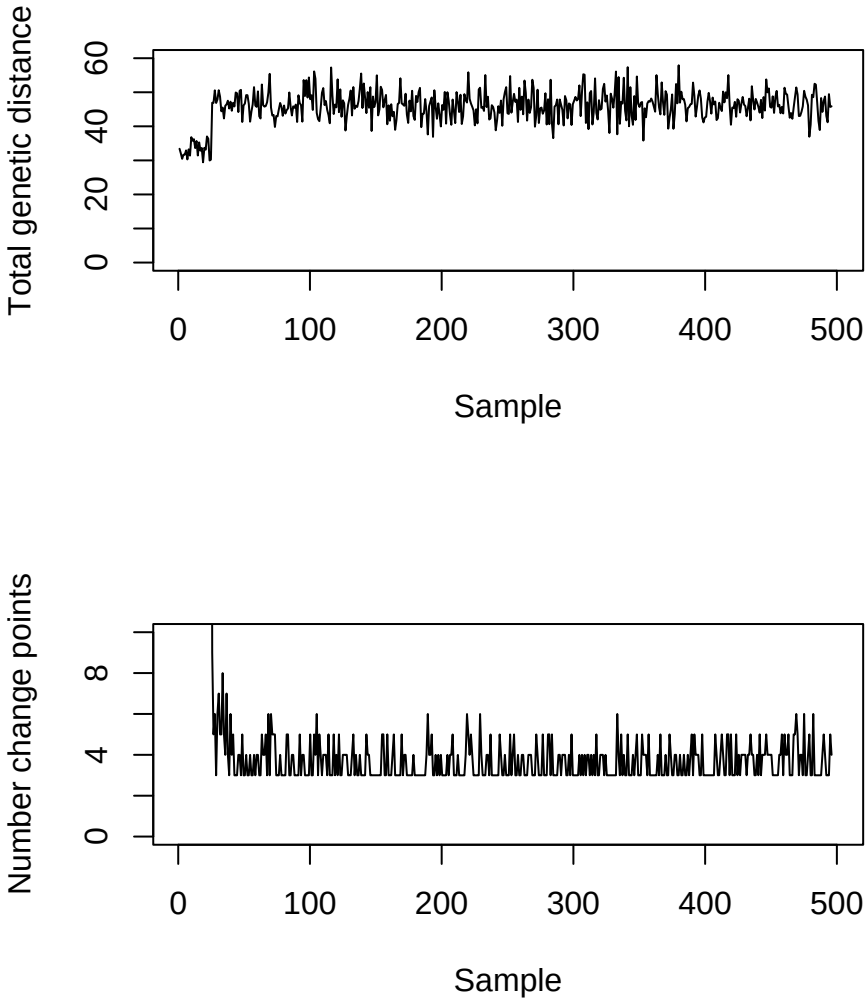

Figure 5: Samples from the RJMCMC for the Finnish LPL data with a penalty of 20. The top panel shows the total genetic distance over the region over a short run of 1000000 updates. For the same run, the lower panel shows the number of change points at each sample (taken every 2000 updates)

NOTE. The computational demands of the algorithm implemented in **complete** mean that this program may take days (or longer!) to run for large sample sizes. For this reason, a series of precomputed tables are available from

`www.stats.ox.ac.uk/~mcvean/LDhat/html`

The program **lkgen** can be used to generate a lookup table for any number of sequences less than that in the original file.

4.6 lkgen

Generates a new lookup table from an existing one using the value of theta per site and grid parameters. To use at the command-line, type

```
% ./lkgen [input_lookup_file]
```

If no file is specified, the user is prompted. The user then inputs the number of sequences to calculate the lookup table for. On completion, the output file, *new_lk.txt* should be renamed before future analyses are carried out.

5 Bugs and idiosyncracies

This code is not written by a professional programmer and neither looks nor behaves like it. However, it is generally stable and should function appropriately if the instructions are followed. Please report any bugs, errors or suggested changes to mcvean@stats.ox.ac.uk.

6 Acknowledgements

Many thanks to Paul Fearnhead for the use of his code in the programs. See `www.maths.lancs.ac.uk/~fearnhea/software` for information about Paul's programs for estimating recombination rates and rate variation. The RJMCMC scheme owes much to discussions with Simon Myers. Thanks also to Philip Awadalla, Chris Spencer and many others who have tested the code. Please report any bugs to mcvean@stats.ox.ac.uk.

References

[1] P. Fearnhead and P. J. Donnelly. Estimating recombination rates from population genetic data. *Genetics*, 159:1299–1318, 2001.

[2] Y. X. Fu and W. H. Li. Statistical tests of neutrality of mutations. *Genetics*, 133:693–709, 1993.

[3] P. J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82:711–732, 1995.

[4] R. R. Hudson. Estimating the recombination parameter of a finite population model without selection. *Genet. Res.*, 50:245–250, 1987.

[5] R. R. Hudson. The how and why of generating gene genealogies. In A. G. Clark and N. Takahata, editors, *Mechanisms of Molecular Evolution*, pages 23–36. Japanese Scientific Societies Press, Tokyo, 1993.

[6] R. R. Hudson. Two-locus sampling distributions and their application. *Genetics*, 159:1805–1817, 2001.

[7] R. R. Hudson. Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics*, 18:337–338, 2002.

[8] R. R. Hudson and N. Kaplan. Statistical properties of the number of recombination events in the history of a sample of DNA sequences. *Genetics*, 111:147–164, 1985.

[9] G. McVean, P. Awadalla, and P. Fearnhead. A coalescent-based method for detecting and estimating recombination rates from gene sequences. *Genetics*, 160:1231–1241, 2002.

[10] G. A. T. McVean, S. Myers, S. Hunt, P. Deloukas, D. Bentley, and P. Donnelly. The fine-scale structure of recombination rate variation in the human genome. *Science*, 304:581–584, 2004.

[11] S. R. Myers and R. C. Griffiths. Bounds on the minimum number of recombination events in a sample history. *Genetics*, 163:375–394, 2003.

[12] D. A. Nickerson, S. L. Taylor, K. M. Weiss, R. G. Clark, A. G.and Hutchinson, J. Stengard, V. Salomaa, E. Vartiainen, E. Boerwinkle, and C. F. Sing. DNA sequence diversity in a 9.7-kb region of the human lipoprotein lipase gene. *Nature Genetics*, 19:233–240, 1998.

[13] J. Rozas, J. C. Sánchez-DelBarrio, X. Messeguer, and R. Rozas. DnaSP, DNA polymorphism analyses by the coalescent and other methods. *Bioinformatics*, 19:2496–2497, 2003.

[14] F. Tajima. Statistical method for testing the neutral mutation hypothesis by DNA polymorphism. *Genetics*, 123:585–595, 1989.

[15] J. Wakeley. Using the variance of pairwise differences to estimate the recombination rate. *Genet. Res.*, 69:45–48, 1997.