Documentation for TreeLD, version 1.0

Algorithm by Sebastian Zöllner* and Jonathan K. Pritchard

Code by Sebastian Zöllner, Xiaoquan Wen and Jonathan K. Pritchard

January 2004

*Adress for correspondence: Department of Human Genetics, 920 E 58th Street CLSC 505, Chicago Il 60637, email: szoellne@genetics.bsd.uchicago.edu.

Contents

1 Introduction

The program TreeLD implements a statistical method for analyzing genotype data in association/linkage disequilibrium studies. It can be applied both for fine-mapping the location of a disease mutation and for performing tests for association. The underlying method is based on making inferences about the presence and location of a disease mutation by reconstructing the ancestry of a set of sequences. The software is designed for SNP-data and can accommodate both case-control data and chromosomes from a TDT test. Our package is named TreeLD, to reflect the emphasis on estimating the ancestral trees that relate the sampled chromosomes at different positions across a region of interest. The computational algorithm underlying TreeLD 1.0 is named LATAG. LATAG and the philosophy behind our method are described in Zöllner and Pritchard (2005).

The Program produces two kinds of results:

- p-values for significance testing for the presence of disease mutation(s) in a region
- estimates and credible regions for the location of a disease mutation

In addition, the program outputs additional information such as the estimated trees relating sampled chromosomes. These can be used to help visualize the subset of the data containing the where the signal.

This document describes the use of the program TreeLD, which is free and comes with no warranty whatsoever. Downloads for the software will be made available via

$$http://pritch.bsd.uchicago.edu/software.html$$

This webpage provides a version of the program for Windows NT, for the Unix Solaris and for Linux. Please send bug reports and requests for new features to Sebastian Zöllner at szoellne@genetics.bsd.uchicago.edu. We ask downloaders to complete the software registration form on the web page.

2 Overview

2.1 Philosophy

The aim of an association study is to analyze the genetic variation in one or more regions of interest and to detect nonrandom association between alleles and the studied phenotype. This genetic variation is generated by a complex stochastic process that includes mutation, genetic drift, recombination, and sometimes natural selection. In population genetics, this stochastic process is typically modeled using the so-called "ancestral recombination graph", or ARG (reviewed by Nordborg, 2001). In the ARG the ancestry of each individual locus can be described as a bifurcating tree (a coalescent tree). TreeLD uses the entire information in the marker data to infer these trees at selected loci in the region of interest. Once the ancestry of a locus is known, we can assess the likelihood that a disease-causing allele arose on this ancestry by looking at the distribution of cases and controls among the tips of the tree.

Figure 1 illustrates the utility of this approach. In the displayed tree, the individuals that show a disease phenotype are clustered, therefore providing good evidence that this tree describes the ancestry of a locus containing a disease mutation. If the individuals carrying the disease were randomly distributed among the tips of the tree, it would be a strong indication for the absence of a disease

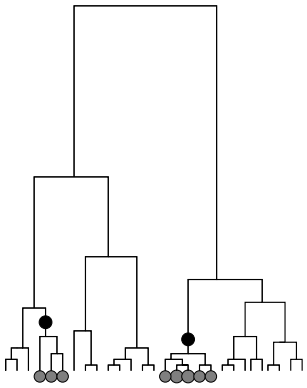


Figure 1: *Hypothetical example of a coalescent genealogy for a sample of 28 chromosomes, at the locus of a disease susceptibility gene. Each tip at the bottom of the tree represents a sampled chromosome; the lines indicate the ancestral relationships among the chromosomes. The two black circles on the tree represent two independent mutation events producing susceptibility variants. These are inherited by the chromosomes marked with gray filled circles. Individuals carrying those chromosomes will be at increased risk of disease. This means that there will be a tendency for chromosomes from affected individuals to cluster together on the tree, in two mutation-carrying clades. The degree of clustering depends in part on the penetrance of the mutation.*

mutation. Thus we can use the ancestry of a locus as an indicator for the presence of one or more disease mutations at the locus of interest. For the purpose of this model, a locus is not a single basepair in the sequence, but a short region of a few kb.

2.2 Overview of the algorithm

In the following section, a quick overview of the application flow will be given; a more detailed description on how to run TreeLD is presented in section 6. Figure 2 also provides a flow-diagram of an analysis with TreeLD.

TreeLD organizes data analysis into projects; each project is associated with a single dataset. The analysis performed in each project usually consists of two steps that are controlled by the user. First the ancestry of the dataset is estimated based on the information in the input file, then the ancestry is analyzed for evidence about the locus of disease mutation.

The first step, estimating the ancestry, is performed by focusing on individual loci (referred to as focal points) along the sequence and inferring the ancestry of each focal point individually with a Markov Chain Monte Carlo (MCMC) algorithm. To account for the uncertainty in this estimate, multiple trees that provide feasible ancestries are generated for each focal point. The performance of the MCMC algorithm depends on the user's choice of parameters for the burn-in and the number of trees that are generated. Suitable choices for these parameters are discussed in section 8. The output of one run of the MCMC for all focal points is called a tree set.

As a second step, the trees sampled at each focal point are analyzed for a signal of the presence of one or more disease mutations. By combining the resulting posterior likelihoods over multiple focal points, the posterior distribution for the locus of disease mutation is generated and an estimate for the position of the disease mutation(s) together with the credible region is obtained. Based on the same tree set, a test for the presence of a disease mutation can also be performed.

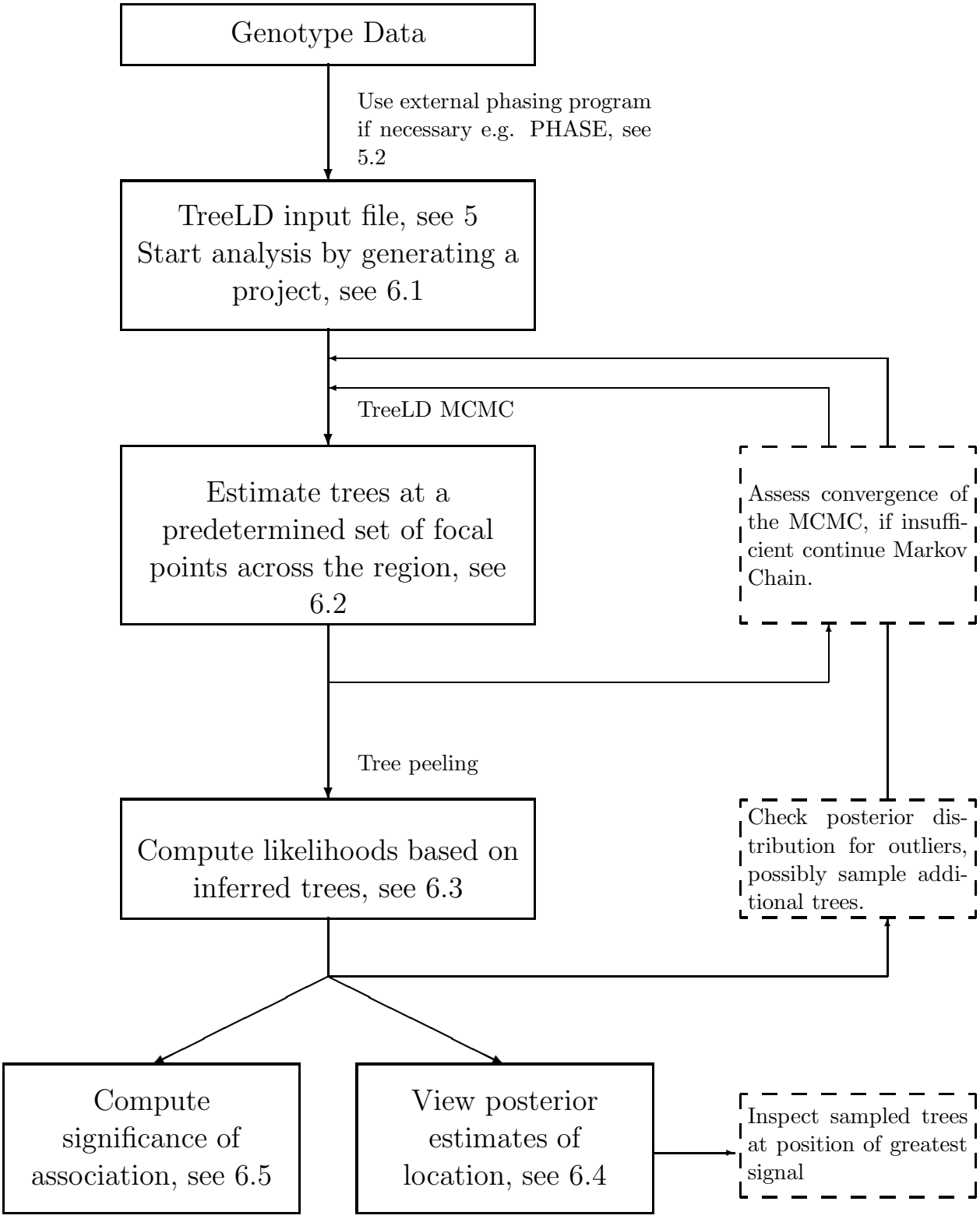


Figure 2: *Flowchart of analysis with TreeLD.*

The resolution of the posterior distribution depends on the number of focal points that have been selected, as does the power of the test for association.

2.3 Technical capabilities and limitations

The algorithm that has been implemented in TreeLD is very computationally intensive. Therefore the available computational resources pose a limit on the size of the dataset that can be analyzed. The largest dataset we have worked with consists of 250 individuals, with 130 markers typed in each individual. A thorough analysis of this dataset took 48 hours on 10 processors in a Linux-cluster. The parts of the analysis that are computationally most intensive are the treebuilding step (see 6.2 and the generation of p-values by permutation (see 6.5).

The major determinant of the total time an analysis takes depends strongly on the initial burn-in of the MCMC. Section 8.1 provides an overview for these times. In section 8.4, we provide some concepts on how to apply TreeLD to large datasets and receive satisfying results.

3 Installation instructions

3.1 Windows version

To install the Windows version of TreeLD, download TLsetup_win.exe to your computer and run the program. Follow the instructions to complete the installation.

3.2 Linux version

To install the Unix/Linux version of TreeLD, download the package for your platform and put the file into an appropriate working directory. Then, unzip the file ("gzip -dc <filename> | tar xvf - "), where <filename> is the name of the downloaded file. Run the program by typing "./TreeLD" in the appropriate directory. On a Unix system, you can also install the program into a standard directory for programs, e.g., /usr/local/TreeLD. You then need to set an environment variable e.g., TreeLD_PATH to the TreeLD package installation directory. For bash users, the syntax is: TreeLD_PATH="/usr/local/TreeLD" export TreeLD_PATH in the .profile file; for csh/tcsh users put setenv TreeLD_PATH "/usr/local/TreeLD" in .tcshrc or .cshrc. You can then run the program by typing TreeLD_PATH/TreeLD.

4 The interface

After starting the program the user sees a window that should look similar to figure 3. The display consists of four main windows and additional pop-up displays that can be activated during the analysis. The four windows are as follows:

File Window. This window contains a tree structure with clickable nodes. Clicking on **Project summary** shows the summary information of the current project and all established tree sets. Clicking on a tree set node loads the selected tree set to the front end. Only one tree set can be loaded at a time.

Map Window. This window displays the distribution of markers in the region of interest. The spacing of markers in this widow is to scale.

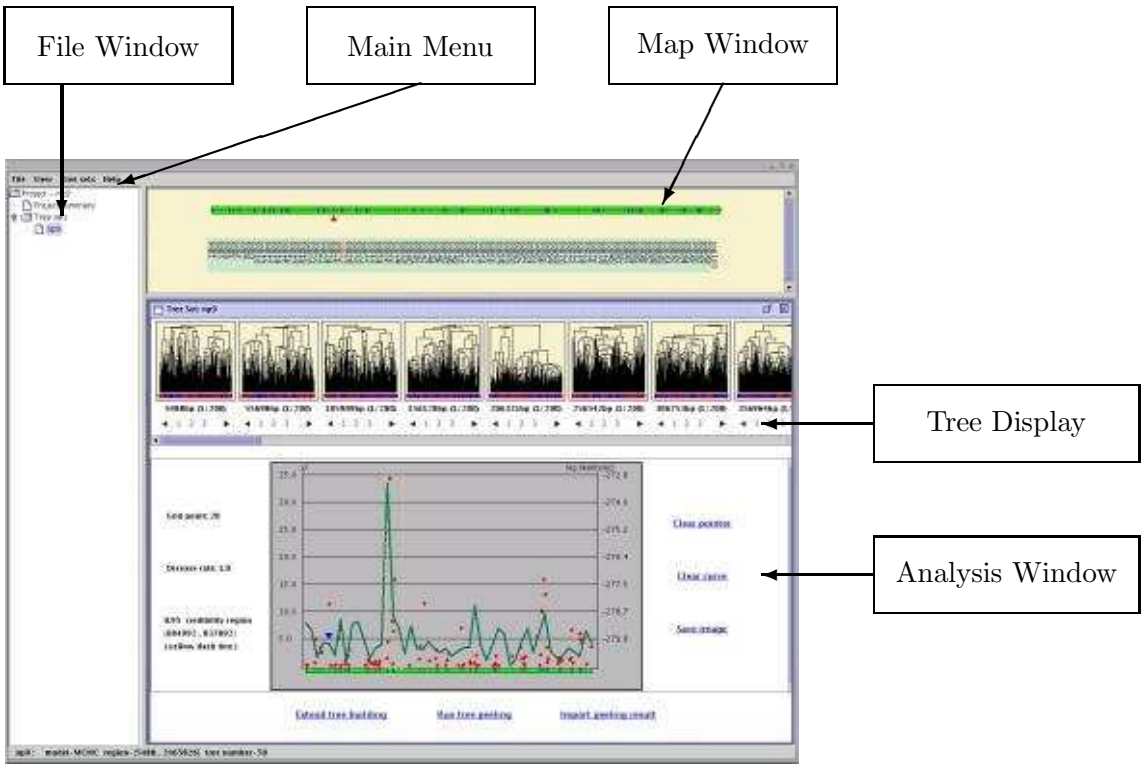


Figure 3: *Screen shot of the main screen of TreeLD*

Tree Display. Here the trees that have been generated by the tree building step of the analysis are displayed. For each focal point, the first generated tree is indicated, with two arrow icons for scrolling to the other trees. Clicking on one of the tree-images opens a new window that shows a full image of the tree. In this image, the phenotypes of the terminal nodes are indicated by a color-scheme.

Analysis Window. This window displays the LD-evidence that is present in the data. The individual dots show the χ^2 of the individual markers, while the connected line indicates the posterior likelihood for the position of the disease mutation. The window provides four items for further analysis of the data:

Extend tree building Builds additional trees in a new MCMC, starting at the last tree from the prior Markov chain. See 6.6 for details.

Tree peeling Starts the analysis of generated trees with treepeeling as described in 6.3.

Import peeling results Loads saved tree peeling result, re-draw the posterior distribution plot in the front end.

Clear Curve Clears the posterior distribution curve that is currently displayed.

Clear pointer Clears the blue triangle pointer that is currently displayed as described in 6.4.

Save Image Saves current plot as a jpeg file.

The basic control of the program is provided by the menu bar. It helps user to manage the projects, tree sets and analyze the data by providing the following four menus:

File. This menu provides functionalities to manage projects. Projects are saved on disk with a .tpj extension.

New project Creates a new project, as described in 6.1.

Open project Opens an existing project file. It is not possible to have more than one open project at a time.

Close project Closes current working project, and saves all the changes.

Recent projects Displays a list of recently visited projects. Open one of the projects by clicking on it.

Exit Exits the TreeLD program.

View→ χ^2-plot. This menu provides utilities to generate a plot of the individual χ^2-values of the markers.

Tree sets. This menu contains functions to help the user create and manage tree sets. It provides the following functions:

New tree set Generates a new tree set as described in 6.2.

Remove tree set(s) Activates a pop-up window for the removal of trees. Click on unwanted tree sets and use the right arrow button move them to the "to be deleted" list.

Generate tree building scripts Provides the tools to produce scripts to run TreeLD on multiple computers. Refer to 7.2 for detail.

Help. This menu provides basic help for the use of TreeLD. If you have any questions that neither this function nor the documentation can address, please contact treeldhelp@bsd.uchicago.edu.

5 Input format

The input file supplied by the user should indicate the position of the markers used, the phenotypes of the individuals studied and the phased genotypes of the sample. A schematic for the input file is given by figure 4 where the quantities are as follows:

Comments All lines in the beginning of the file that is started by a hash (#) will be ignored by the program and can therefore be used for comments.

P The line that designates the map-positions of the markers starts with a P (upper case). The positions should be separated by a single whitespace.

Position(i) Each of those numbers indicates the position of a marker relative to an arbitrary point of reference in basepairs. The loci must be in consecutive order along the chromosome (i.e. the positions have to be increasing). Position(i) should be separated from Position(i+1) by a single whitespace.

For each individual i in the sample, the phenotype and genotype information must be provided. The first line indicates the phenotype, the second and possibly third line contain the genotype information.

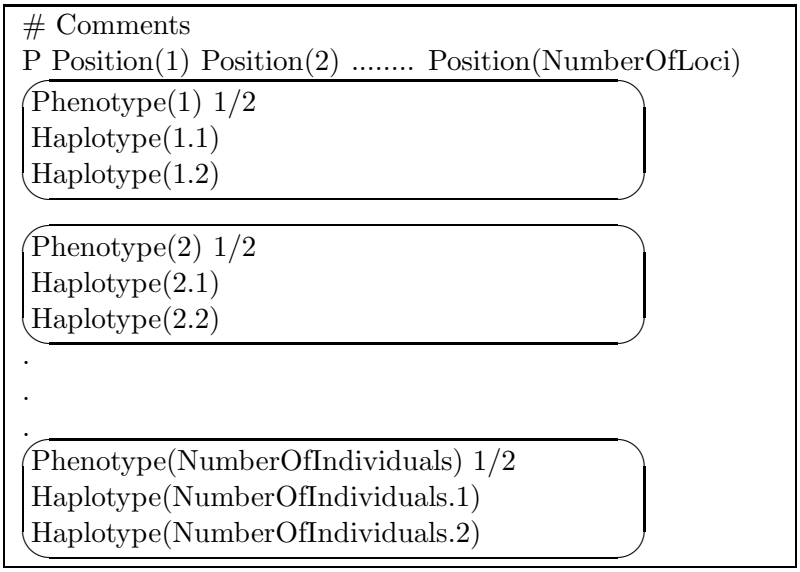


Figure 4: *Schematic of an input file. Each oval box indicates the input information for one individual. Figure 5 shows a specific example.*

Phenotype(i) This floating point number indicates the phenotype of the individual i. For a QTL-study, this can be the measured quantitative trait. In a case-control study, the phenotypes should be indicated as 1.0 for cases and 0.0 for controls.

1/2 This entity should designate how many chromosomes share the phenotype designated in the same line. For case-control studies on autosomal loci this will be a 2, while for male X-chromosomal loci or for non-transmitted haplotypes in a TDT this will be a 1.

Haplotype(i.x) In this line(s) the input file contains the one or two haplotypes as indicated by the line with the phenotype information. The state of each SNP is indicated by a 1 or a 2 without a space between the individual characters. It is not important which allele of the SNP is assigned to which number, as no information about ancestral state is used. The number of markers displayed in each of this lines must match the number of marker positions provided in the line starting with a P.

An example for a simple input file is given in the file *SampleInput.txt*, which is shown in figure 5.

Please note that the chromosomes in the input file are assumed to be unrelated. Thus, data that are generated in a case-control study can be used without modifications. See 5.3 for instructions how to generate an input file for trios.

5.1 Genetic map distances

As the haplotype pattern in the sample is the result of recombination events in the ancestry of the sample, specifying the precise recombination rate (genetic distance) between markers improves the quality of the results. While the physical location of the markers provides the simplest estimate, it is now becoming clear that recombination rates can vary greatly over short physical distances. Therefore, we suggest the use of a statistical model to estimate the intermarker recombination rates and apply those rates instead of the physical distances (for example Li and Stephens, 2003; McVean

```
# SampleInput.txt, 5 SNPs, 3 cases, 3 controls
P 820 22312 34200 55333 82290
1.0 2
21122
21211
1.0 2
21111
21122
1.0 2
21122
21112
0.0 2
11122
12111
0.0 2
22211
11112
0.0 2
11122
11111
```

Figure 5: *Example of an input file. The first line is a comment that is ignored by the program. The second line indicates the location of the 5 markers at bp 820, 22312,..., 82290 relative to an arbitrary starting point. The first three individuals in the sample are the cases, designated by the phenotype of 1.0. The 2 after the phenotype indicates that these are diploid individuals. The last three individuals are controls as indicated by the phenotype 0.0.*

et al., 2004). These programs will provide relative recombination rates between markers. By replacing the intermarker distances with these relative rates, a more precise genetic map of the region of interest will be used. As TreeLD estimates the regionwide recombination rate as a nuisance parameter, it is not necessary to provide the absolute genetic distances in the sample, as long as the relative genetic distances are correct.

5.2 Missing information and unknown phase

This release of TreeLD does not support the analysis of sequences with missing data or unknown phase. To impute missing data and phase information, we recommend using PHASE 2.1 (Stephens et al., 2001), which can be obtained from *http://www.stat.washington.edu/stephens/*. To obtain the best results from PHASE, it is advisable to run the program on one combined dataset of cases and controls. The perlscript PH2TL.pl included with the TreeLD package transforms the output of PHASE into an input file for TreeLD. Usage is

$$PH2TL.pl < PHASE.out > \ < Phenotypes.txt > \ < TreeLD.in >$$

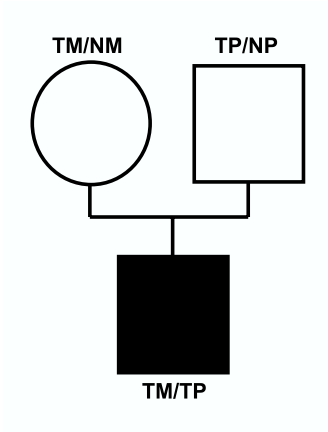
where$< PHASE.out >$ designates the name of the output file that is generated by PHASE, $< TreeLD.in >$ is the name of the input file for TreeLD and $< Phenotypes.txt >$ is a text file that

contains the phenotype information,comprised of one phenotype per line in the same order as the input file for PHASE.

Note that while PHASE estimates the uncertainty for each estimated state, the script PH2TM.pl selects for each marker the allele that has the highest posterior probability according to PHASE and assigns this result as the state of this marker. Thus, using PHASE in this manner will reduce the uncertainty that is present in the data and therefore generate confidence intervals that are somewhat anti-conservative.

5.3 TDT-data

A TDT-dataset consists of trios where the parents are usually unaffected while the offspring is affected (Spielman et al., 1993). Therefore half of the chromosomes are observed twice in the sample, even though they occur only once independently. To apply TreeLD to TDT-data, those duplicated chromosomes must be removed from the sample. To this end, the user must include only the non-transmitted chromosomes of the parental generation in the input file, while the offspring generation is still presented as diploid individuals. For example the trio depicted in figure 6.1 would be included in an input file as shown in 6.2. It is not possible at present to apply TreeLD to families with more than one offspring.



6.1 *Trio*

```
0.0 1
Haplotype-NM
0.0 1
Haplotype-NP
1.0 2
Haplotype-TM
Haplotype-TP
```

6.2 *Input file for one trio.*

Figure 6: *Example of an input-file generated from a TDT-Trio. NM is the haplotype on the non-transmitted maternal chromosome, NP is the haplotype on the non-transmitted paternal chromosome, while TM and TP are the transmitted paternal and maternal chromosomes.*

6 Running TreeLD

To start the program under Windows NT, click on the TreeLD program icon. Linux/Unix users have to run the batch script "TreeLD" to initiate the program. Then a new window should be opened similar to the one described in section 4. To start the analysis of a new datafile, the user must generate a new project. For such a project, the analysis of a datafile consists of two major steps. The first step

is the inference of a set of trees consistent with the marker data. The second step is the analysis of the generated set of trees. In the following, each of these steps is discussed in more detail.

6.1 Create a project

As mentioned earlier, TreeLD organizes data analysis into projects. Each project is associated with a single input file. Within each project, the analysis program can be run multiple times with different sets of parameters. Within a project, each run of the treebuilding algorithm is saved as a tree set.

To create a new project, click on **File→Create project** and fill out all the information in the pop up window. This prompts a query for the path to the input datafile which must be formated according to section 5. After the dataset is loaded into memory, the marker map that is provided in the input file will be displayed in the Map Window. It is not possible to have more than one project open at a time.

To continue the analysis of an old project, click on **File→Open project**. After loading a project into memory, all tree sets that have been generated for this project will be displayed in the File Window. Clicking on one of these tree sets will load it into the computer memory. It will then appear in the Tree Display.

6.2 Generating trees

To start the first step in the analysis and generate a tree set, you have two options. You can either click on **Tree set→new tree set** and follow the instructions to drag the mouse to highlight the region of interest in the map window. Alternatively, you can mark the region of interest in the map window without using the Main Menu. Either step will open a pop-up dialog box where all the necessary model parameters can be set. This window asks for the beginning and the end of the studied region in bp and the number of focal points that should be evenly distributed in that region. After clicking on the button labeled **More options**, the burn-in length for the MCMC in steps of updates can be entered. Each step consists of 10,000 proposed updates to the tree. Some discussion on what burn-in to choose here is presented in section 8. Furthermore you are given the option of performing a reduced burn-in for every MCMC after the first. If this option is chosen, the MCMC does not use a random starting tree for the second and all further focal points. Instead the last tree that has been generated at the neighboring focal point is used as the first tree for the new Markov Chain. The advantage of using this option is that the burn-in for all but the first MCMC can be substantially lower, reducing computation time. This gain is maximized if the program is running on a single CPU. If the tree-building process is distributed to multiple processors (see 7.2), each of these processors runs one full-length MCMC as its first focal point and only subsequent Markov Chains profit from the reduced burn-in. The disadvantage of this option is that the tree sets at neighboring focal points are not entirely independent, which may make it harder to spot aberrant results as described in 8.2. See 8.1 for further discussion of this option.

You can also enter the number of trees that should be sampled after burn-in. Each tree is sampled after one step of updates, consisting of 10,000 proposed changes to the tree. A good starting value for the number of sampled trees analysis is between 100 and 200 trees. Larger numbers become computationally very burdensome and offer only a limited advantage. Section 6.6 describes how more trees can be sampled if the initial set of trees is insufficient. Finally, the seed to initialize the random number generator can be set from this window. If you want to generate several independent runs of the treebuilding algorithm, make sure that each run has a different seed.

After this setup of the treebuilding process has been finished, a button **Start running trees** appears in the Tree Display. Clicking this button will start running the tree-building.

6.3 Analysis of the trees with the peeling algorithm

After the set of trees is generated, the next step in the analysis is to analyze the set of trees. The tool that is provided in this version of TreeLD is called treepeeling, a fast and efficient algorithm that is described in detail in Zöllner and Pritchard (2005). This algorithm is deterministic: i. e. multiple runs of the algorithm on the same set of trees will result in the same output.

When a tree set has been generated, the tree peeling options will be made available to the user in the Analysis Window. After clicking on the **Tree peeling** button, a dialog box will appear. The program will then require the input of parameters for the rate of mutation at the disease locus and the range of penetrance parameters that should be explored by the peeling algorithm. The disease mutation rate β reflects the expected number of non-ancestral disease mutations in the ancestry of a randomly ascertained individual in a population of constant size and should range from 0.1 to 1. See Pritchard (2001) for a discussion about the implications for the choice of β. The choice of penetrance parameters depends on the analyzed phenotypes.

Case-control data. For case- control data, Treepeeling calculates the likelihood of the phenotypes conditional on two penetrance parameters, the probability of a chromosome with a mutant allele to be carried by a case individual in the sample and the same probability for a wildtype allele. The algorithm can either be run for a known set of penetrance parameters, or alternatively it can average over a grid of penetrance values. As the penetrance of a single locus in a pre-ascertained sample of cases and controls can differ from penetrance parameters ascertained through other means, it is usually advisable to integrate over the space of penetrances. The algorithm performs this analysis on a grid of penetrances for the wildtype and the mutant allele, the density of that grid must be decided on by the user. The number of gridpoints are then evenly distributed by the program. If the penetrance parameters are unknown, we suggest using the maximal gridsize of 20x20 points.

Quantitative data. For quantitative data, the underlying model is that the phenotype of a chromosome is drawn from one of two normal distributions with the same standard deviation and differing means, dependent on whether it carries the disease mutation. As those means and standard deviation are usually unknown, the program calculates the posterior likelihood on a grid of values and averages over the grid. When starting the treepeeling step, the user is asked for a minimum and a maximum value for mean and standard deviation and the number of gridpoints that are to be used between those two values. To make sure that the resulting likelihood is independent of the selected grid density, the user may want to experiment with different densities. If the phenotype data is clearly not normal, it may be worth attempting transformations of the phenotype data to make it more normal.

After providing this information, the process will be started by clicking the **Start tree peeling** button in this dialog box. This will open a window displaying the output from the peeling algorithm for each focal point as they are being generated, indicating the progress of the program. Buttons on the bottom of this window provide the following options to the user:

Save data Saves the generated likelihoods to disk. If this option is not used, the peeling results will be discarded when TreeLD is closed.

Plot data Opens a window that plots the peeling likelihood at each focal point.

Compute CI Calculates a credible interval base on the result of the peeling algorithm. Clicking on this button opens a window that allows you to enter the percentile for the credible region. After this calculation is finished, a box appears, displaying the beginning and the end of the credible interval.

Done Closes the peeling output window and displays the results as a graph in the Analysis Window.

Cancel Interrupts the treepeeling analysis or the computation of the credible interval.

When this process is finished, and the window is closed, the estimated posterior distribution is displayed as a graph in the Analysis Window.

6.4 Generating the posterior distribution

To estimate the position of the disease mutation and the confidence interval, TreeLD creates the posterior distribution of the locus of the disease mutation based on the output of the treepeeling step. To calculate the likelihood of a tree under the disease mutation model, the program averages over the likelihood for each set of penetrance variables. By then averaging over all trees at one focal point, the posterior likelihood of that focal point to be the locus of disease mutation is calculated. This averaging process can be overly influenced by outliers. Section 8.2 describes the impact of outliers in more detail and offers advice on how to reduce their impact.

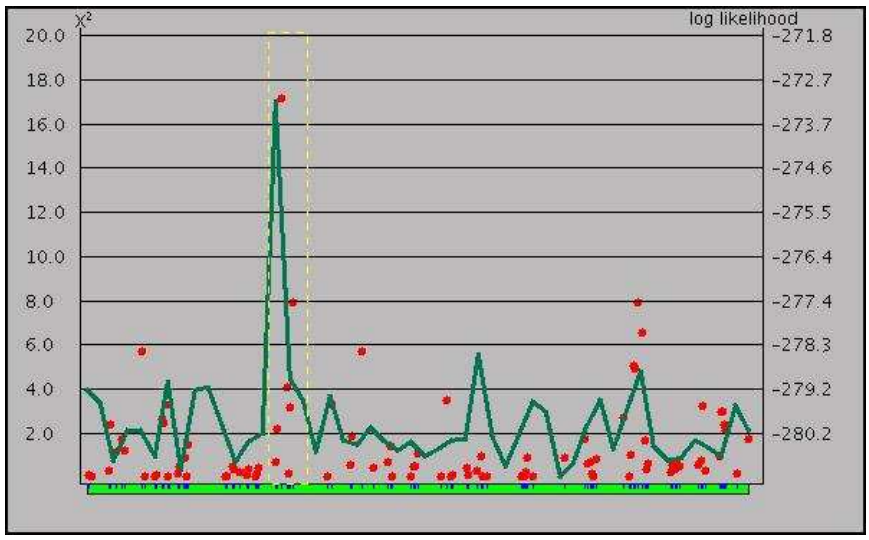


Figure 7: *Example of an Analysis window after a treepeeling run for a case-control dataset. The red dots indicate the χ^2-values of the individual markers as indicated by the left vertical axis, the continuous line indicates the posterior loglikelihood to carry the disease locus as indicated by the right vertical axis and the dashed yellow box marks the credible region.*

Under most circumstances, the location with the highest posterior likelihood is the best estimate for the location of the disease mutation. (This might not be true if this maximum is an isolated peak,

see 8.2 for further discussion.) If a credible region has been in generated as described in 6.3, it is displayed in the Analysis Window as a dashed box.

It should be pointed out that the displayed continuous distribution is an estimate from values that have been generated at different focal points. Therefore the resolution of this distribution is dependent on the number of focal points it is based on. A posterior distribution that is based on more focal points will usually result in a more precise estimate of the location of disease mutation and a tighter credible region. See 8.3 for some suggestions about the optimal densities.

6.5 Testing for the presence of a disease mutation

TreeLD also implements a significance test for the presence of a disease locus in the region of interest based on the output of the treepeeling program. During the tree peeling step, the likelihood of a disease gene being present is obtained by finding the focal point and the set of penetrance parameters that generate the highest likelihood. With this a likelihood ratio (LR) is calculated by dividing this by the likelihood of the null hypothesis. While asymptotic theory suggests that this LR is approximately χ^2-distributed, our simulation studies have shown that applying this distribution is conservative. Nevertheless, the p-value that is generated by this distribution can act as an indicator for the strength of the association signal. Therefore, the program provides two p-values from the LR-test, one that is the unmodified p-value and a second one that is corrected for multiple tests by a Bonferroni-correction. Please note that for a tight grid of focal points, the signal at adjacent focal points is highly correlated and the Bonferroni correction is very conservative. Nevertheless, it provides useful starting point for the analysis. If the resulting p-values are indicative but not significant, a more appropriate p-value can be obtained by permuting the phenotypes among the individuals in the sample. This shuffling occurs over all trees at once and thus generates a overall p-value that does not need correction for multiple testing. To perform this analysis, rerun the peeling algorithm and check the **Run permutation** box. The p-value is then displayed in the text console window at the end of the run. The random phenotypes that are generated by the shuffling are dependent on the seed the program is started with. Thus if the user wants to apply the same random phenotype-distributions to multiple datasets, the program has to be restarted with the same seed in every analysis.

As described before, the likelihood of a focal point will depend on its distance to the location of the disease mutation. Therefore, if the focal points are distributed on a tighter grid in the region of interest, the power of a test for association may be increased.

6.6 Generating additional trees

After performing an initial analysis, you may want to sample additional trees. To do so, click on **Extend tree building** in the Display Window. This will cause a dialog box to appear, which will query for the additional burn-in that is to be performed before new trees are sampled, the number of trees to be sampled and whether to discard the currently saved trees. If you are using this option because the MCMC did not converge in the first chain and you want to run it for more steps to assure convergence, then you should discard the first set of trees. On the other hand, if you want to increase the number of trees that are sampled from the posterior, you may want to keep the trees that are already sampled. The program also provides the option to discard only a subset of the already sampled trees. If that option is selected, trees are deleted in the order that they have been sampled in.

After setting up the tree building, you will be given the option to start the tree building, or to generate job scripts that can be exported to other machines.

7 Additional options

7.1 Visual analysis of trees

An advantage of the analysis performed by TreeLD is that the LD-signal is visualized in a unique way by the generated tree set. Clicking on a tree set will cause a pop-up window to appear, showing the tree at the location of interest. If the input is a case-control sample, the tips in this tree will be colored to indicate the case/control status of the tip. This allows the user to identify clades in the tree that have an overabundance of cases among their descendants.

It is specially informative to view the trees at the location with the highest posterior likelihood. For this purpose, you can click on the focal point with the highest likelihood in the displayed graph. This will cause a blue triangle to appear, as shown in figure 7. It will also highlight the tree set that has been generated at this spot for further inspection.

Analyzing the trees at this location provides an informal way to estimate the number of disease mutations, as every such cluster is the result of one disease mutation. Furthermore the user can assess which chromosomes carry which disease mutation by identifying the clade it belongs to. Clicking on any internal node in the tree will open a window, showing the length of sequence that is transmitted to all terminal decedents of that node. This visualizes the region that is most likely to carry a disease mutation.

7.2 Creating tree building scripts

Running tree building on a single computer can be very time intensive. To take advantage of parallel processing, the user could set up tree building programs running on multiple computers simultaneously. The procedure is described as follows:

1. Install the tree building program (mcmc_tree.exe) on each computer. (Note: it is not required to install the whole software package on those machines)
2. Create a tree set as described in 6.2.
3. Click on **Tree sets→Generate tree building scripts**, follow the guide of pop up dialog boxes and save the generated scripts.
4. Copy the script and data file on each machine.
5. Start the script onto each machine
6. Copy the outputs from each machine back to the project directory if necessary.

7.3 Generating tree peeling scripts

The user may also want to create scripts for performing the treepeeling step on multiple nodes at once. As the treepeeling step is quite fast, this may only be necessary for generating p-values for a large study with many focal points. In this case, it is necessary to make sure that all permutation p-values are obtained with the same seed for the random number generator.

The procedure for generating the peeling scripts is similar with creating tree building scripts. To get the batch scripts from the front end, click on **Generate scripts only** button when finish setting up the tree peeling options

7.4 Exporting images

Every graphic that is displayed in the TreeLD program can be exported as a jpeg file. To export images, click on the **Save image** button in corresponding image window/frame. This will open a window where the path and the name if the generated file can be entered.

8 Choosing parameters for the Markov Chain Monte Carlo

Obtaining reliable results from TreeLD depends on multiple factors. Some, like the informativeness of the marker map cannot be influenced. On the other hand, the burn-in of the MCMC, the number of trees generated and the number of focal points chosen are important parameters for the analysis of a dataset that the user assigns. While the optimal value for each parameter depends on the individual dataset, the following describes some considerations for a reasonable starting point for an analysis.

8.1 Choosing burn-in

The MCMC algorithm that is the basis of the treebuilding program starts out from a random tree and through successive iterations improves the tree until it converges to a set of trees that are likely given the marker data. This process is called burn-in. Each burn-in step consists of 10,000 updates to the tree and takes between 3 and 40 seconds on a 2.4 Ghz processor with 512 Mb of RAM. To ensure convergence of the MCMC, a sufficiently high number of burn-in steps must be selected. This number depends mostly the size of the analyzed dataset but other factors can influence it, such as the informativeness of the selected markers.

The following table shows some approximate convergence times that we have observed for datasets that we analyzed. These numbers only provide a starting point for the analysis of individual datasets.

Sample size	Number of markers	burn-in
100	25	100
100	50	250
100	100	325
250	25	300
250	50	650
250	100	825
500	25	600
500	50	1125
500	100	1400

Table 1: *Expected number of update steps for the treebuilding algorithm until a sufficient degree of burn-in is achieved. Each step consists of 10,000 updates. The Sample size is displayed as the number of diploid individuals in the sample.*

To monitor convergence of a Markov Chain Monte-Carlo run, a timeplot of likelihood of the estimated variables conditional of the haplotype data can be displayed by right-clicking on the trees of interest in the Tree Display, and selecting **Show MCMC time series plot** in the pop up menu. This plot displays the probability to observe the marker data conditional on the tree (Pr(Data|Tree)).

As long as the Markov Chain is still converging, this plot is increasing, after convergence it should move horizontally.

If during treebuilding the option is taken to pick the tree at a neighboring locus as the starting tree of the MCMC and not a random tree, then the main purpose of burn-in is to make sure that the trees sampled are independent from the starting tree (This is the case if an unequal number of burn-in steps is chosen during treebuilding, see 6.2). In this case, the time series plot does not serve as an indicator for sufficient burn-in, as the probability of the starting tree conditional on the data may not be a lot smaller than the probability of a tree that is sampled after convergence.

If this analysis indicates, that the elected burn-in had been insufficient, it is possible to restart the MCMC from the last generated tree as described in 6.6.

Incomplete convergence will result in a reduced signal from the data, so if individual markers show association but the posterior distribution generated by the treepeeling step is basically flat this may be a sign that the burn-in is insufficient.

8.2 Effect of outliers among the sampled trees

Due to computational restraints, only a limited number of trees can be drawn from the posterior distribution. This makes the posterior likelihood that is calculated at each location susceptible to being influenced by outliers with unusually high likelihoods. These occur when individual trees with a low posterior likelihood but high support for the presence of the disease location are sampled. This may result in the posterior likelihood being overestimated at that location. If the map of focal points is sufficiently dense (see 8.3, influence of an outlier may be indicated by a single spike in the posterior distribution, where focal point i has a high posterior likelihood, but neither focal point $i-1$ nor $i+1$ have an elevated signal. If a focal point is really close to the locus of disease mutation, then the neighboring focal points will also be in the proximity of the disease mutation and therefore also show a increased posterior likelihood. On the other hand, if the increased likelihood at a focal point is due to an outlier, adjacent focal points will not show an increased likelihood for the presence of a disease mutation. While the impact of outliers can be reduced by sampling more trees from the posterior distribution, some outliers may have a likelihood that is orders of magnitude higher than the likelihood of all other trees that are sampled at the same focal point. Thus it may be not computationally viable to sample enough trees to control for the effect of outliers.

On the other hand, a signal that is generated by an outlier in the tree-distribution will usually not be repeated if a second set of trees is generated and analyzed. Therefore, it is advisable to verify any peak in the posterior distribution by generating additional trees at locations where a signal is generated. To make sure that this new set of trees is independent from the first set of trees, it may be necessary to restart the analysis from a random tree and to repeat the MCMC, including burn-in.

8.3 Density of focal points

The tree-reconstruction step of TreeLD estimates the ARG of a region of interest by sequentially estimating the marginal trees at a number of focal points. Based on this estimate, the posterior distribution for the locus of disease mutation is generated. The quality of this estimate depends on the number of focal points it is based on. As the computation time of TreeLD increases linearly with the number focal points, it is important to select a number that is sufficient to provide good results. While theoretical calculations suggest that increasing the grid-density to up to 1,000 focal points/cM may still yield an improved result, it is questionable that using more focal points than there are

markers in the dataset will improve the results noticeably. For our analyses we use between 10 and 50 focal points per cM.

8.4 Strategy

For large datasets, doing a thorough analysis of the dataset with TreeLD can be very computationally intensive. Therefore, it is advisable to perform an approximate run as a first pass analysis with a low number of focal points (maybe 1 focal point per 0.1cM), a small number of sampled trees and a low burn-in. This analysis may be sufficient to identify interesting areas in the sequence of interest, which then can be analyzed by a tighter grid of focal points and a more extensive MCMC. The generated trees in this first pass will only be rough approximations of the true ancestry of the locus, but they may already contain some signal about the presence of a disease mutation. If those approximate trees are insufficient and only little signal is generated, treebuilding can be restarted from the last sampled tree and the original trees can be discarded (see 6.6). It should be pointed out that running the program this way should only be done for exploratory analysis of the data as repeated approximate runs generate a multiple-testing problem that can create false positives.

As it is much easier in this version of the program to extend the burn-in or to sample additional trees than it is to add focal points, it may be a sensible approach to plan the analysis accordingly.

9 How to cite this program

In publications that include results generated with this program please cite Zöllner and Pritchard (2005).

References

Li, N. and Stephens, M. (2003). Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. *Genetics*, 165:2213–2233.

McVean, G. A. T., Myers, S. R., Hunt, S., Deloukas, P., Bentley, D. R., and Donnelly, P. (2004). A coalescent approach to study linkage disequilibrium between single-nucleotide polymorphisms. *Science*, 304:581–584.

Nordborg, M. (2001). Coalescent theory. In Balding, D., Bishop, M., and Cannings, C., editors, *Handbook of statistical genetics*, pages 179–212. Wiley.

Pritchard, J. K. (2001). Are rare variants responsible for susceptibility to common diseases? *Am. J. Hum. Genet.*, 69:124–137.

Spielman, R. S., McGinnis, R. E., and Ewens, W. J. (1993). Transmission test for linkage disequilibrium: the insulin gene region and insulin-dependent diabetes mellitus (IDDM). *Am. J. Hum. Genet.*, 52:506–513.

Stephens, M., Smith, N. J., and Donnelly, P. (2001). A new statistical method for haplotype reconstruction from population data. *Am. J. Hum. Genet.*, 68:978–989.

Zöllner, S. and Pritchard, J. K. (2005). Coalescent-based association mapping and fine mapping of complex trait loci. *Genetics*, [To Appear].

Index