TreeDT: Tree Pattern Mining for Gene Mapping

Petteri Sevon, Hannu Toivonen, and Vesa Ollikainen

Abstract—We describe TreeDT, a novel association-based gene mapping method. Given a set of disease-associated haplotypes and a set of control haplotypes, TreeDT predicts likely locations of a disease susceptibility gene. TreeDT extracts, essentially in the form of haplotype trees, information about historical recombinations in the population: A haplotype tree constructed at a given chromosomal location is an estimate of the genealogy of the haplotypes. TreeDT constructs these trees for all locations on the given haplotypes and performs a novel disequilibrium test on each tree: Is there a small set of subtrees with relatively high proportions of disease-associated chromosomes, suggesting shared genetic history for those and a likely disease gene location? We give a detailed description of TreeDT and the tree disequilibrium tests, we analyze the algorithm formally, and we evaluate its performance experimentally on both simulated and real data sets. Experimental results demonstrate that TreeDT has high accuracy on difficult mapping tasks and comparisons to other methods (EATDT, HPM, TDT) show that TreeDT is very competitive.

Index Terms—Biology and genetics, nonparametric statistics, nonnumerical algorithms and problems.

1 Introduction

The discovery of new disease susceptibility genes can have immense importance for human health care. The gene and the proteins it produces can be analyzed to understand the disease-causing mechanisms and to design new medicines. Further, gene tests on patients can be used to assess individual risks and for preventive and individually tailored medications.

Gene mapping aims at discovering areas in the genome that have a statistical connection to a given trait. In association studies, genetic markers and haplotypes along chromosomes are used to discover associations between phenotypes (e.g., diseased-associated versus control) and chromosomal regions (i.e., potential disease gene loci) in samples of haplotypes from patients and control individuals. The growing number of available markers, hundreds of thousands now being available on off-the-shelf chips, opens new opportunities, but also emphasizes the importance of computationally scalable methods.

We introduce TreeDT, a novel method for gene mapping. It analyzes the observed haplotypes by constructing *haplotype trees*, tree-structured string patterns that reflect the genetic history of potential disease susceptibility (DS) gene loci. Roughly, the idea is as follows: At a specific locus, TreeDT uses haplotype trees as estimates for the genealogy of the sample of haplotypes. In the genealogy at (or near) a true DS gene locus, disease-associated chromosomes are overrepresented in one or more subtrees in which all chromosomes have inherited the disease predisposing mutation from a common ancestor. We call this phenomenon *tree disequilibrium*. At locations far from the DS locus, the genealogical trees are expected to be in equilibrium. The TreeDT method estimates and tests the disequilibrium of genealogical trees at a number of loci and predicts a DS gene to be where the most significant disequilibrium is observed. The idea of TreeDT mapping was preliminarily reported in reference [1].

The contributions of TreeDT are the following:

1. a novel approach to gene mapping using the analogy between the concepts of genealogical trees and haplotype trees,
2. specification of a statistical test for assessing tree disequilibrium at a given location, for comparing the results across several loci, and for estimating the overall statistical significance of the best finding, and
3. an efficient algorithm for the tests, including an algorithm for carrying out multiple nested permutation tests at the cost of a single permutation test.

In the next section, we review the gene mapping problem. In the following sections, we specify the proposed method and then give an algorithm for it. After a brief review of related work, we experimentally evaluate and compare the performance of the algorithm. Finally, we conclude with a discussion.

2 Problem Background

Let us assume the goal is to locate a disease-susceptibility gene for a given disease. We next briefly review the genetic background; without loss of generality, we restrict the discussion in this paper to one chromosome. In case one has several chromosomes to analyze, the results for different chromosomes are independent and Bonferroni correction [2] can be applied to the p values obtained by TreeDT for individual chromosomes.

A genetic *marker* is a short polymorphic region in the DNA, denoted here by M1, M2, ... The different sequence variants (*alleles*) at the marker locus are denoted in our examples by positive integers. The number of alleles per marker is small: typically, less than 10 (for microsatellite

- *P. Sevon and H. Toivonen are with HIIT BRU, Department of Computer Science, PO Box 68, FI-00014 University of Helsinki, Finland. E-mail: {petteri.sevon, hannu.toivonen}@cs.helsinki.fi.*
- *V. Ollikainen is with Helsinki Polytechnic Stadia, PO Box 4020, FI-00099 City of Helsinki, Finland. E-mail: vesa.ollikainen@stadia.fi.*

Manuscript received 8 Mar. 2005; revised 4 July 2005; accepted 12 Sept. 2005; published online 1 May 2006.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBB-0011-0305.

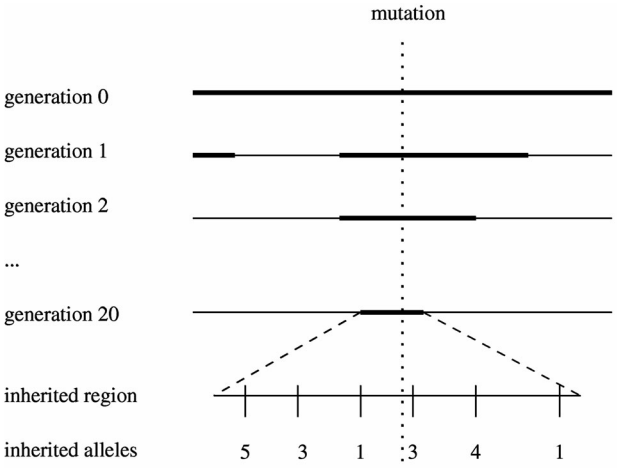

Fig. 1. A carrier of the mutation in generation 20 has inherited alleles from the ancestral chromosome in generation 0 around the gene locus.

markers) or exactly two (for SNP markers). The collection of markers used in a particular study constitutes its *marker map* and the corresponding alleles in a given chromosome constitute its *haplotype*. For the purposes of this paper, the input data consists of haplotypes of diseased and control individuals.

All of the current carriers of a disease predisposing mutation have inherited it from one or more ancestral chromosomes that have introduced it into the population. They also have inherited segments of varying length around the mutation (Fig. 1). With sufficiently dense marker maps, it is possible to observe *linkage disequilibrium* (LD), nonrandom association between nearby markers. Due to sharing of segments inherited from a common mutation-carrying ancestral chromosome, there is also LD between the unobserved DS gene and markers near it and, consequently, the markers can be used as surrogates for the DS gene in genetic analyses.

In diseases with a reasonable genetic contribution and especially in population isolates into which a few founders have introduced the mutation, affected individuals are likely to have higher frequencies of alleles and haplotype patterns inherited from a common ancestor (e.g., a founder of a population isolate) near the DS gene locus than control individuals. This is the starting point of association-based mapping methods: Where in the genome are haplotype patterns strongly associated to the disease? While the principle is simple, the problem is far from trivial. The recombination history is stochastic; mutation carriers often only have a higher risk of being diseased than noncarriers and, in a case-control study, both groups are usually mixes of carriers and noncarriers; finally, there is missing information.

Our gene mapping framework belongs to the family of association/LD-based mapping methods. It is applicable to case-control settings, where the input consists of disease-associated and control haplotypes. Each individual contributes a chromosome pair, so the number of chromosomes is twice the number of individuals. We ignore the fact that chromosomes come in pairs and simply consider the input data as consisting of a set of independent haplotypes. The

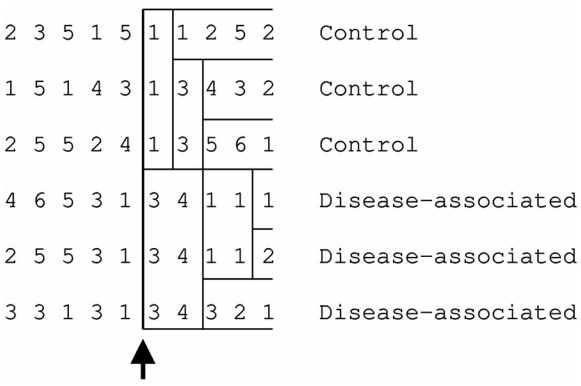

Fig. 2. Set of haplotypes A-F, sorted by the partial haplotypes to the right from the location pointed to by the arrow.

task is now to predict the location of a disease susceptibility gene on the map, given a set of disease-associated haplotypes and a set of control haplotypes.

3 Method

3.1 Haplotype Trees

TreeDT is based on estimating and evaluating the genealogy of the observed haplotypes. At any given location, the (unknown) genealogical tree consists of the chromosomes in the sample as the leaves and each internal node corresponds to the *most recent common ancestor* (MRCA) chromosome of its descendants in the tree. Due to recombinations, different parts of a chromosome have different histories and genealogical trees vary across the chromosome.

In the genealogical tree for a DS gene locus, all the chromosomes in one or more subtrees—depending on how many times the mutation has been introduced to the population (sample)—have inherited the disease predisposing mutation. These subtrees should then have more disease-associated chromosomes than other parts of the tree. In the genealogical tree for a random locus, in contrast, disease-associated and control chromosomes are randomly distributed. Since the expected number of historical recombination events between two locations increases with the distance between them, the closer a location is to the DS gene, the more similar its genealogical tree is to the genealogy of the DS gene.

True genealogies are not known, but haplotypes can be used to estimate them. Given a location in the chromosome—a potential DS gene locus—the haplotypes to the right of the location can be organized into a tree by their shared prefixes (Fig. 2 and Fig. 3), as can haplotypes to the left. TreeDT uses the right and left trees as two different estimates of the genealogical tree at the location. At a given locus, a pair of chromosomes shares a region, inherited from their MRCA, around the locus. Inside this region, no crossovers have occurred in either of the lineages descending from the MRCA to the two chromosomes. The more distant the MRCA, the more recombination events have taken place in the two lineages and, consequently, the shorter the expected length of the shared region is. The use of haplotype trees to estimate genealogies is based on the converse of the observation above: The expected time to the

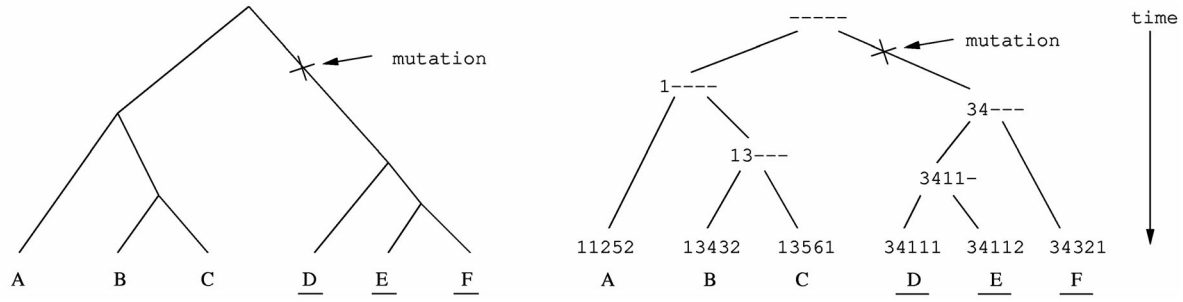

Fig. 3. On the left, the true (unobserved) genealogy for the haplotypes at the pointed location in Fig. 2. On the right, the haplotype tree used as an estimate of the true genealogy. The disease-associated haplotypes are underlined. In this example, the tree is constructed at the DS gene locus and the mutation-carrying haplotypes form a single subtree {D, E, F}, even though the order of internal nodes of subtree {D, E, F} is incorrectly estimated.

MRCA of a pair of haplotypes decreases as the length of haplotype sharing between them increases. The structure of a haplotype tree perfectly complies with this observation and, in this sense, it gives the most likely genealogy. The most typical error is a wrong order for internal nodes, but this often has a small effect or no effect at all on the accuracy of TreeDT (Fig. 3).

3.2 Tree Disequilibrium Test

TreeDT constructs the left and right haplotype trees between each pair of consecutive markers. For each tree, it then assesses tree disequilibrium: Are disease-associated haplotypes overrepresented in a set of subtrees, suggesting that the tree has been constructed at or near a DS locus? Our test considers all different cardinalities of subtree sets (denoted by k throughout the text) up to the number of haplotypes in the data set, or possibly limited by some maximum value $k_{\max}$.

We introduce a novel tree disequilibrium test, intended for predicting DS gene locations. The vicinity of the location for which the test gives the lowest p value (the most significant disequilibrium) is the most likely candidate area for the DS gene location. The method also computes a corrected overall p value for the most significantly disease-associated location, which can be used for predicting whether there is a DS gene in the studied chromosome at all or not.

The tests performed by TreeDT are organized in a four-level hierarchy. All the tests are conditional on the estimated genealogical trees and only test the distribution of the disease-association statuses of the haplotypes in the trees. The null hypotheses effectively state that there is no genetic effect in the observed data, i.e., the disease-association statuses are randomly assigned to the observed haplotypes.

We first give an overview of the testing hierarchy and then describe each level in more detail, including definitions of the null hypotheses and test statistics. The levels of the hierarchy are, in bottom-up order:

0. For a fixed set S of nonoverlapping subtrees of a haplotype tree, we define statistic $Z_k(S)$ (1), $k = |S|$, corresponding to null hypothesis $\mathrm{H}_0(S)$, to be used as a measure of disequilibrium. No p values are estimated at this level.
1. For each haplotype tree T in the set of left and right-side haplotype trees at all tested locations and each number of subtrees k, TreeDT computes test statistic $ZMAX_k(T)$ (2), the maximum of $Z_k(S)$ over all sets S of k nonoverlapping subtrees of T, and estimates the respective p value $p_k(T)$ under null hypothesis $\mathrm{H}_1(T, k)$.
2. For each tested location l, TreeDT estimates a local p value $local\_p(l)$ using a test statistic derived from $p_k(T_{\mathrm{L}})$ and $p_k(T_{\mathrm{R}})$ for all k, where T_{L} and T_{R} are the left and right-side haplotype trees at location l (statistic $d(l)$ given in (3); null hypothesis $\mathrm{H}_2(l)$).
3. At the top level, TreeDT estimates an overall p value $overall\_p$. It is corrected for multiple testing over tested locations using the smallest local p value as the test statistic (null hypothesis H_3).

3.2.1 Level 0: Measuring Disequilibrium of a Given Set of Subtrees

At the bottom level of the testing hierarchy, we consider null hypothesis $\mathrm{H}_0(S)$ for assessing the disequilibrium of a fixed set S of subtrees of a given haplotype tree.

Null hypothesis $\mathbf{H_0}(S)$. Haplotypes in set S are not disease-associated at a higher probability than haplotypes in the entire data.

For measuring the disequilibrium, we define an ad hoc Z_k statistic, which is a sum of k standard Z statistics. For a set S of k nonoverlapping subtrees $T_1, \ldots, T_k$,

$$Z_k(S) = \sum_{i=1}^{k} z_i, \qquad \text{where } z_i = \begin{cases} \frac{a_i - n_i p}{\sqrt{n_i p(1-p)}}, & \text{if } n_i > 1 \\ 0, & \text{if } n_i = 1, \end{cases} \tag{1}$$

where a_i is the number of disease-associated haplotypes and n_i the total number of haplotypes in subtree $T_i \in S$, and p is the proportion of disease-associated haplotypes in the whole sample. Each z_i measures the departure of the observed number of disease-associated chromosomes (a_i) from the expectation ($n_i p$) in standard deviations ($\sqrt{n_i p(1-p)}$) in subtree T_i under the assumption of binomial distribution with parameters n_i and p and is approximately normal if n_i is large. Subtrees consisting of a single leaf do not contribute to the test statistic since it is only possible to extract localization information from two or

more haplotypes sharing a region. We are interested only in subtrees in which the proportion of disease-associated haplotypes is greater than expected, $z_i > 0$ (i.e., we assume the mutation to increase the risk of the disease).

3.2.2 Level 1: Estimation of a p Value for a Given Haplotype Tree and Number of Subtrees

We do not test against $\mathrm{H}_0(S)$ directly for sets S of nonoverlapping subtrees of a given tree T. Instead, we perform a single test for each k (number of subtrees) against null hypothesis $\mathrm{H}_1(T, k)$.

Null hypothesis $\mathbf{H_1}(\boldsymbol{T}, \boldsymbol{k})$. There is no set of k nonoverlapping subtrees of T in which haplotypes are disease-associated at a higher probability than haplotypes in the entire data.

As the test statistic for given T and k, we use the maximum value of Z_k over all sets of k nonoverlapping subtrees of T, denoted by $ZMAX_k(T)$:

$$ZMAX_k(T) = \max\{Z_k(S) | S \text{ is a set of } k \text{ nonoverlapping subtrees of } T\}. \quad (2)$$

We chose Z_k as the base statistic because it can be efficiently maximized in the space of all possible subtree sets simultaneously for all k, using the recursive algorithm given in the Algorithms section. Efficiency of the maximization procedure is important because it is performed many times, often millions, during the execution of TreeDT.

Given a tree T, the $ZMAX_k(T)$ statistic has different distributions for different numbers k of subtrees. In order to get a comparable measure for all values of k, TreeDT estimates a p value $p_k(T)$ for each k under null hypothesis $\mathrm{H}_1(T, k)$ using a permutation test, as will be described later. We are aware that the maximization of the test statistic introduces a selection bias favoring smaller subtrees, but, at this point, we have not taken any actions to compensate for it.

3.2.3 Level 2: Estimation of Local p Values

In order to get a single p value representing the disequilibrium at location l, we need to combine the information from the trees to the left and to the right of the location. As a combined measure, we use $d(l)$, the product of the lowest p values over all k from each side:

$$d(l) = \min_k\{p_k(T_{\mathrm{L}}(l))\} \cdot \min_k\{p_k(T_{\mathrm{R}}(l))\}, \quad (3)$$

where $T_{\mathrm{L}}(l)$ and $T_{\mathrm{R}}(l)$ are the left and right haplotype trees at location l. Since the values of $d(l)$ are not directly comparable across the chromosome, e.g., due to varying marker density and background LD, a new local p value, $local\_p(l)$, for the combination is estimated under null hypothesis $\mathrm{H}_2(l)$. The local p values are comparable across different loci.

Null hypothesis $\mathbf{H_2}(\boldsymbol{l})$. There is no set of (at most $k_{\max}$) nonoverlapping subtrees in the haplotype tree for either side of location l in which haplotypes are disease-associated at a higher probability than haplotypes in the entire data, i.e., there is no gene effect present at location l.

The output of TreeDT is essentially a list of p values for the tested locations. A point prediction for the gene location is obtained by taking the best location; alternatively, a (potentially fragmented) region of length x is obtained by taking the best locations until a length of x is covered. A direct link between the p value and the probability that the gene is indeed close to the location cannot be established. The local p values are used simply as a method of ranking the locations.

3.2.4 Level 3: Estimation of a Corrected Overall p Value

A single corrected p value for the best finding can be obtained with a third significance test using the lowest local p value as the test statistic (H_3).

Complete null hypothesis $\mathbf{H_3}$: There is no set of (at most $k_{\max}$) nonoverlapping subtrees in the haplotype tree for either side of any tested location in which haplotypes are disease-associated at a higher probability than haplotypes in the entire data, i.e., there is no genetic effect present in the entire data.

The resulting overall p value can be used to answer the question whether there is a gene in the investigated area in the first place or not: The null hypothesis is rejected if the resulting p value is lower than a predetermined significance level.

Since there are dependencies among tests at levels 0-2, the null distributions of test statistics at levels 1-3 are very complex (due to maximization and minimization), regardless of how we define the null distribution at level 0. We therefore resort to resampling techniques (see, e.g., [3]) to estimate the joint null distributions of the test statistics and, consequently, the p values. Ignoring mutual dependencies and using, e.g., Bonferroni correction to obtain the overall p value would result in grossly inflated p values.

The complete null hypothesis effectively states that the disease-association statuses are randomly assigned to the haplotypes. In a typical study setting, the numbers of disease-associated and control haplotypes in the sample are predetermined and, therefore, random permutation of disease-association statuses of the haplotypes is the appropriate resampling method. A permutation test is conducted in the following way: 1) Draw a sample of resampled data sets, where the disease-association statuses of the haplotypes are randomly permuted, 2) compute the appropriate test statistic for each data set in the sample, and 3) estimate a p value by comparing the test statistic for the observed data to the empirical null distribution obtained from the permuted data sets.

Because null hypotheses $\mathrm{H}_2(l)$, $\mathrm{H}_1(T, k)$, and $\mathrm{H}_0(S)$ are restrictions of H_3 to location l, haplotype tree T and number k of subtrees, and set S of subtrees, respectively, tests against all null hypotheses H_0-H_3 can be carried out as permutation tests as described above.

4 Algorithms

The *TreeDT* algorithm can be decomposed into two subtasks: 1) construction of the left and right-side haplotype

Algorithm *MaximizeZ*(T)

Input: Haplotype tree T
Output: $ZMAX_k(T)$ for all k, $1 \leq k \leq n$, where n is the number of leaves in T

Method:

1 If T is a leaf, then $ZMAX_1(T) := 0$
2 else :
3 $F := \{\}$
4 For each immediate subtree T' of T :
5 Recursively call *MaximizeZ*(T')
6 For each k, $1 \leq k \leq a$, where a is the total number of leaves in forest F :
7 $ZMAX_k(F \cup \{T'\}) := ZMAX_k(F)$
8 For each k, $a < k \leq a + b$, where b is the number of leaves in T' :
9 $ZMAX_k(F \cup \{T'\}) := 0$
10 For each pair (i, j), $1 \leq i \leq b$ and $1 \leq j \leq a$:
11 $ZMAX_{i+j}(F \cup \{T'\}) := \max\{ZMAX_{i+j}(F \cup \{T'\}), ZMAX_i(T') + ZMAX_j(F)\}$
12 $F := F \cup \{T'\}$
13 For each k, $1 \leq k \leq b$:
14 $ZMAX_k(F) := \max\{ZMAX_k(F), ZMAX_k(T')\}$
15 For each k, $1 \leq k \leq n$, where n is the total number of leaves in T :
16 $ZMAX_k(T) := ZMAX_k(F)$
17 Calculate $Z_1(\{T\})$ (Eq. 1). $ZMAX_1(T) := \max\{ZMAX_1(T), Z_1(\{T\})\}$

Fig. 4. Algorithm for calculating $ZMAX_k(T)$ for all values of k.

trees for each location and 2) carrying out the permutation tests. These tasks are discussed in detail in the following subsections.

4.1 Constructing Haplotype Trees

The haplotype trees to the left and right from each analyzed location can be efficiently constructed using the textbook radix sort algorithm (see, e.g., [4]). The algorithm iterates over all markers from right to left. At each marker, the intermediate result is a sorted list of the partial haplotypes to the right from the marker. A right-side tree can be easily derived from the intermediate list because the haplotypes belonging to one node form a continuous block in the sorted list (see Fig. 2 and Fig. 3 for an example). The left-side trees can be identified similarly by sorting the inverted haplotypes. The computational cost of constructing all the trees is linear both in the number of markers and the number of haplotypes and it is negligible compared to the cost of the permutation test procedure.

4.2 An Algorithm for Maximizing the Z_k Test Statistic

The first step of the testing procedure is computation of level 1 test statistic $ZMAX_k$ for all haplotype trees T and numbers k of subtrees; level 0 is completely embedded in this step. It is essential that the time complexity of the algorithm for maximizing the Z_k values is as low as possible because it must be executed for each haplotype tree and permutation in turn. An efficient recursive algorithm, MaximizeZ, which propagates the locally maximized Z_k values upwards in the haplotype tree, is presented in Fig. 4.

Let us generalize $ZMAX_k$ to apply to forests (sets of trees) in the obvious way: $ZMAX_k(F)$ is the maximum value of $Z_k(S)$, where S is a set of k nonoverlapping subtrees of forest F. The algorithm is based on the recursive definition of $ZMAX_k$:

$$ZMAX_k(\{T_1, \ldots, T_m\}) = \max_{U \subseteq \{1,\ldots,m\}} \left\{ \sum_{i \in U} ZMAX_{k_i}(T_i) \middle| \sum_{i \in U} k_i = k \right\}$$

$$ZMAX_k(T) = \begin{cases} ZMAX_k(IS(T)), & \text{if } k > 1 \\ \max\{Z_1(\{T\}), ZMAX_1(IS(T))\}, & \text{if } k = 1, \end{cases}$$

where $IS(T)$ is the set of immediate subtrees of T (subtrees whose roots are children of the root of tree T).

For haplotype tree T that is not a leaf, the algorithm first computes $ZMAX_k$ statistics for the forest F consisting of all its immediate subtrees. The subtrees are processed one at a time: For each subtree, the algorithm first recursively calls itself and then computes $ZMAX_k$ statistics for the forest of all the previously processed subtrees and current subtree (lines 6-14). The $ZMAX_k$ values of tree T are identical to those of forest F, except for $k = 1$, where it is the maximum of $Z_1(T)$ and $ZMAX_1(F)$ (lines 15-17).

The time complexity of the algorithm is $O(n^2)$, both on average and in the worst case, where n is the number of leaves in the tree, i.e., the number of haplotypes in the data set. By setting an upper limit $k_{\max}$ for the size of the subtree sets, the time complexity can be reduced to $O(n)$ with a constant coefficient proportional to $k_{\max}^2$, $k_{\max}$ being typically small, ≤ 10. The only modification required in the algorithm is an additional condition $\leq k_{\max}$ for index k and sum $i + j$ at lines 6, 8, 10, 13, and 15 of the algorithm. When LD-mapping is applicable, the majority of mutation carriers are concentrated in only a few subtrees of the haplotype trees at the DS gene locus and using this prior information to restrict the number of subtrees may slightly increase the power of the method, as shown in the Experiments section. In the experiments for this paper, we use an upper limit of three subtrees.

Algorithm *NestedPermutationTests*

Input: Set L of tested locations, left and right haplotype trees $T_{\mathrm{L}}(l)$ and $T_{\mathrm{R}}(l)$ for each location $l \in L$, number q of permutations
Output: Local p value $local\_p_0(l)$ for each tested location, overall corrected p value $overall\_p$

Method:

// Level 1: Compute p values for each haplotype tree T and number k of subtrees
For each haplotype tree $T \in \{T_{\mathrm{L}}(l), T_{\mathrm{R}}(l) \mid l \in L\}$:
  For each $i \in \{0, \ldots, q\}$:
    If $i = 0$, then $P_i(T) := T$ // Consider the observed tree as "permutation 0"
    else generate tree $P_i(T)$ by permuting the disease-association statuses of the haplotypes in T into some (pseudo-random) order that is a function of i // We want the ith permutation to be the same for each T
    Call $MaximizeZ(P_i(T))$, which returns $ZMAX_k(P_i(T))$ for each number k of subtrees
  For each number k of subtrees :
    For each permutation $i \in \{0, \ldots, q\}$:
      Estimate p value $p_k(P_i(T))$ by comparing $ZMAX_k(P_i(T))$ to its empirical null distribution $ZMAX_k(P_j(T))$, $j \in \{0, \ldots, q\} \setminus \{i\}$ // See text for details
  For each permutation $i \in \{0, \ldots, q\}$: $min\_p(P_i(T)) := \min_k\{p_k(P_i(T))\}$

// Level 2: Compute local p values
For each location $l \in L$:
  For each $i \in \{0, \ldots, q\}$: $d_i(l) := min\_p(P_i(T_{\mathrm{L}}(l))) \cdot min\_p(P_i(T_{\mathrm{R}}(l)))$
  For each $i \in \{0, \ldots, q\}$:
    Estimate local p value $local\_p_i(l)$ by comparing $d_i(l)$ to its empirical null distribution $d_j(l)$, $j \in \{0, \ldots, q\} \setminus \{i\}$ // See text for details

// Level 3: Compute the corrected overall p value
For each $i \in \{0, \ldots, q\}$: $min\_local\_p_i := \min_l\{local\_p_i(l)\}$
Estimate overall corrected p value $overall\_p$ by comparing $min\_local\_p_0$ to its empirical null distribution $min\_local\_p_i$ over all permutations $i \in \{1, \ldots, q\}$ // See Eq. 4

Fig. 5. Algorithm for nested permutation tests.

The space complexity is $O(n)$ if the number of subtrees is not restricted. The average complexity reduces to $O(\log n)$ for the restricted case. Proofs for the correctness and time and space complexities of the algorithm are given in Appendix A, which can be found on the Computer Society Digital Library at http://computer.org/tcbb/archives.htm.

4.3 An Efficient Algorithm for Multiple Nested Permutation Tests

With permutation tests, p values are estimated by comparing the test statistic to a sample drawn from its null distribution obtained by randomly permuting the observed data set. The proportion of permutations for which the test statistic has at least as extreme a value as the observed value is an unbiased estimate for the p value:

$$p = \frac{1}{q}|\{i \in \{1, \ldots, q\} | s_i \geq s_o\}|, \tag{4}$$

where s_o is the observed value of the statistic, s_i is the value of the statistic from the ith random permutation, and q is the number of permutations. If the test statistic is being minimized instead of maximized, as is the case with p values as test statistics, then $s_i \geq s_o$ should be replaced with $s_i \leq s_o$.

With nested permutation tests, the upper level test statistics are based on p values from the lower level tests. In order to draw a sample from the null distribution at the upper level, these lower level p values must also be computed for permuted data sets by sampling repermuted data sets from each original permuted set. The straightforward algorithm for a two-level nested permutation test using nested loops would have time complexity proportional to q^2, where q is the number of permutations at each level, as for each of the q permutations at the top-level, q permutations are generated at the bottom-level.

With TreeDT, there are three levels of nested permutation tests (levels 1-3, no p values were estimated at level 0) and the time complexity of the simple algorithm is proportional to q^3. The test would be intractable already with rather low permutation counts. However, the time complexity can be drastically reduced using the same set of permutations at each level of the test and comparing each permutation to all other permutations and the unpermuted data, thus only maximizing the Z_k values q times instead of q^3 times for each haplotype tree. This is possible because any random permutation of a permuted data set is also a random permutation of the original data. The improved algorithm, NestedPermutationTests, is presented in Fig. 5.

The problem with computing p values for a permuted data set is that there are only $q-1$ other random permutations to compare to, resulting in different granularities of p values for the permutations and the observed data. There are two possible solutions: either generate an extra permutation or include the observed data in the comparison. We chose the latter approach because it guarantees consistency of p values: If the observed value of the test statistic is more extreme than the value for a permuted data set, then the p value for the observed data set is guaranteed to be smaller than the p value for the permuted data set. This would not be the case if an extra permutation was generated for comparison.

If a null hypothesis is not true, then the observed data is not drawn from the null distribution and the respective p values for permuted data sets that are larger than the p value for the observed data are conservative (by less than $1/q$). This may result in slightly anticonservative local and overall p values for the observed data; however, this only happens if the null hypothesis should be rejected and, thus, does not affect the Type I error rate, but may increase statistical power.

Due to the finite number of permutations, the precision of the p values given by a permutation test may not be sufficient for accurate localization. In some situations, even a very large number of permutations do not produce any values for the test statistic as extreme as the observed values for several consecutive locations. For this reason, the p values returned by the first and second level permutation tests are determined slightly unconventionally. At levels 1 (line 8) and 2 (line 13), the returned p value is interpolated linearly between the p values corresponding to the next lower and higher values for the test statistic obtained by permutations. At level 1, if the observed value s_o of the $ZMAX_k$ statistic is higher than the highest value s_p obtained from the permutations, the p value is extrapolated using the ratio of the two scores, $p = \frac{s_p}{qs_o} \leq \frac{1}{q}$. At level 2, if the observed value of the d statistic is lower than the lowest obtained from the permutations, a lower boundary value of zero for d is used to interpolate the p value. Finally, the top-level test returning the overall p value is implemented in the usual conservative manner (4).

The time complexity of the loops at lines 7 and 12 is $O(q \log q)$ using an algorithm which first sorts the values of the test statistic for all the permutations. The time complexity of the algorithm, predominated by calls to MaximizeZ, is $O(qn^2s)$ (or $O(qns)$ if an upper limit $k_{\max}$ is given), where n^2 (or n) is the time complexity of the MaximizeZ algorithm and s is the number of markers (or tested locations) in the chromosome.

4.4 Algorithm TreeDT

As a summary, we give an informal description of the TreeDT algorithm. As input it takes a marker map and a set of disease-associated and control haplotypes. The parameters for the algorithm are the number of permutations and, optionally, the maximum or exact number of subtrees to be tested. Its output consists of local p values for all tested locations and an overall corrected p value as returned from the NestedPermutationTests algorithm. The set of locations to be tested is the set of intervals between adjacent markers and the flanking regions on both sides of the marker map.

The haplotypes are sorted using the radix sort algorithm. Construction of haplotype trees and level 1 of Nested PermutationTests are interleaved: After each iteration of radix sort, level 1 of the permutation procedure is performed for the right-side haplotype tree implicit in the intermediate sorting result. One only needs to store the smallest p value over all numbers of subtrees for each haplotype tree and each permutation. In a second pass, the same is done for the inverted haplotypes and the left-side haplotype trees. All there is left after this point is straightforward execution of levels 2 and 3 of the permutation algorithm.

5 Related Work

During the past decade, several parametric statistical gene mapping methods have been proposed that model LD around the DS gene [5], [6], [7], [8], [9], [10]. Some more recent methods explicitly model the genealogy of the sample at the DS locus [11], [12], [13]. These methods typically estimate a number of parameters, most importantly, the location of the DS gene and the respective confidence intervals. However, the models are based on a number of assumptions about the inheritance model of the disease and the structure of the population, which may be misleading for the statistical inference. The methods tend to be computationally complex, especially with respect to the number of markers.

TreeDT belongs to the class of nonparametric gene mapping methods, which typically scale well to very large data sets. The simplest nonparametric test for case-control haplotype data tests all haplotypes of adjacent markers up to some maximum length for disease association using, e.g., Fisher's exact test or the χ^2-test for independence.

Transmission/disequilibrium tests (TDT) [14] are an established way of testing whether alleles or haplotypes are transmitted from heterozygous parents to affected offspring more often than expected. We performed multipoint TDT analyses with microsatellite data using the GENEHUNTER2 software package [15]. With SNP data, we used the Exhaustive Allelic TDT (EATDT) of Lin et al. [16], which performs the TDT test for all haplotypes up to a given maximum length.

Haplotype Pattern Mining (HPM) [17] analyzes the disease association of each marker based on the set of haplotype patterns, essentially strings with wildcard characters, overlapping it. HPM has been extended for detecting multiple genes simultaneously [18] and to handle quantitative phenotypes and covariates [19]. Extending HPM to analyze unphased genotypes instead of haplotypes and to compute overall corrected p values has been discussed in [20].

Recently, methods have been proposed that explicitly account for allelic heterogeneity at the DS gene [21], [22], [23].

An alternative approach for LD-based mapping is linkage analysis. The idea is to analyze families and to find out at which markers alleles tend to be co-inherited with the disease. Linkage analysis does not rely on linkage disequilibrium, so, in that respect, it is more widely applicable than LD-based methods. The downside is that estimates are rough (due to the smaller effective number of meioses utilized), that it has weaker power to detect genes with moderate effect, and that collecting information from larger families is more difficult and expensive.

The nested permutation test procedure in TreeDT is closely related to the maxT (level 1) and minP (levels 2 and 3) methods by Westfall and Young [3], intended for adjusting p values for multiple simultaneous tests. For example, the overall p value returned by TreeDT is the same as the most significant local p value adjusted using the minP method. However, Westfall and Young did not give a computationally efficient algorithm for the method nor did they consider more deeply nested tests than two-level permutation. Ge et al. [24] have independently proposed a

fast algorithm for two-level permutation testing that, like TreeDT, uses the same set of permutations on both levels.

6 Experiments

We next empirically evaluate the performance of TreeDT and compare it to other methods suitable for the analysis of very large data sets: single marker TDT, multipoint TDT of GENEHUNTER2 (hereafter m-TDT for short) using haplotypes of up to four markers, EATDT, and HPM, our earlier proposal based on haplotype patterns. We evaluate the methods on simulated microsatellite and SNP data and a real Type 1 diabetes data set.

6.1 Data Sets

6.1.1 Simulated Data

We designed several different test settings, with variation in the strength of the genetic effect, in the number of founders who introduced the mutation to the population, in the amount of missing information, and in the sample size. One set of fixed values for these parameters will be used as a baseline setting, against which the effects of the parameters will be compared. For statistical analyses, we created 100 independent artificial data sets in each test setting. Great care was taken to generate realistic data; a detailed description of the simulation procedure is given in Appendix B, which can be found on the Computer Society Digital Library at http://computer.org/tcbb/archives.htm.

We simulated an isolated population that had grown from 100 founders to 100,000 individuals over a period of 20 generations. A fixed number of sources (founder chromosomes) of the mutation were selected for each data set. In the baseline setting, there is one mutation-carrying founder chromosome with 2 percent frequency in the final population.

Our disease model was fully penetrant; all the mutation carriers were affected. We selected a challenging disease model where only a small proportion A (in the baseline setting $A = 10\%$) of the disease-associated chromosomes carries the disease-predisposing mutation, a complication often encountered in the analysis of common diseases.

For each data set, we sampled random affected individuals (100 in the baseline setting) from the final generation and their parents. For each individual in the sample, the simulation produced an unordered allele pair (*genotype*) for 101 microsatellite markers spaced equidistantly over a 100 cM chromosome. By looking at the genotypes of the offspring-parents-trio, we can infer which alleles have been transmitted to the offspring. The haplotypes constructed from the alleles transmitted to each affected individual in a sample were labeled disease-associated, whereas the haplotypes constructed from the nontransmitted alleles of the parents can be conveniently used as controls. However, in a rare case of three similar heterozygote genotypes at a marker (both parents and the offspring), the parental origin of the inherited alleles is ambiguous. We chose to take this into account by setting the alleles unknown in these cases. Thus, 3.7 percent of alleles are missing, making the mapping task more difficult, but also more realistic. The final output of the simulation procedure is a collection of 100 data sets, consisting of 200 disease-associated and 200 control haplotypes in our baseline setting.

We also included one setting with dense SNP data from a general population with no recent founder effect. We used data provided by Lin et al., simulated for the original EATDT article [16]. There were 50 data sets, each consisting of a 5 Mb (= 5 cM) sequence simulated using an improved version of Hudson's algorithm for coalescent model with recombination [25]. The number of markers varied from 504 to 571 and the average D' between adjacent markers was ~0.76. The frequency of the disease predisposing mutation was 0.1. The risk of becoming affected was three times larger for mutation carriers (nine times larger for double carriers) than noncarriers and prevalence was 3.3 percent. For our experiments, we sampled 100 trios from the original data sets with about 450 nuclear families. With these data, we compared TreeDT to EATDT, which infers haplotypes with no missing alleles from trios using a method by the same authors [26]. We used the haplotypes inferred by EATDT also as input for TreeDT and HPM.

6.1.2 Real Type 1 Diabetes Data

In order to test and demonstrate the real-life performance of TreeDT, we analyzed a Type 1 diabetes data set [27]. We have used exactly the same data set in the article introducing HPM [17]. The study subjects were genotyped for 25 markers covering 14Mb, including the HLA-region. There is a known major susceptibility locus located in the center of the marker map. The data consists of 385 affected sib-pairs and their parents, out of which 100 randomly chosen families were used. One child was randomly selected from each family. The nontransmitted parental haplotypes were used as controls and the transmitted as cases.

6.2 Analysis of TreeDT

First, we assess the prediction accuracy of TreeDT with different values of A, the proportion of disease-associated chromosomes that actually carry the mutation (Fig. 6a). The results are reported as curves that show the percentage of 100 data sets where the gene is within the predicted region, as a function of the length of the predicted region. In other words, the x-coordinate tells the cost a geneticist is willing to pay, in terms of the length of the region to be further analyzed, and the y-coordinate gives the probability that the gene is within the region. For $A = 20\%$ or 15% the accuracy is very good and, with lower values of A, the accuracy decreases until with $A = 5\%$ only in 20-30 percent of data sets can the gene be localized within a reasonable accuracy of 10 cM. We remind the reader that the test settings have been designed to be challenging, and to test the limits of the approach.

Next, we evaluate the effect of the only parameter of TreeDT, the number of mutation-carrying subtrees that are searched for in each tree. An upper limit of $k_{\max} = 3$ subtrees, used in the previous test, is evaluated against fixed numbers of one, two, or three subtrees, with a varying number of founders that introduced the mutation (Fig. 6b). As we increase the number of founders, evidence about the gene location becomes more fragmented and, accordingly, the performance degrades. While the differences between different numbers of subtrees are not large, it is interesting

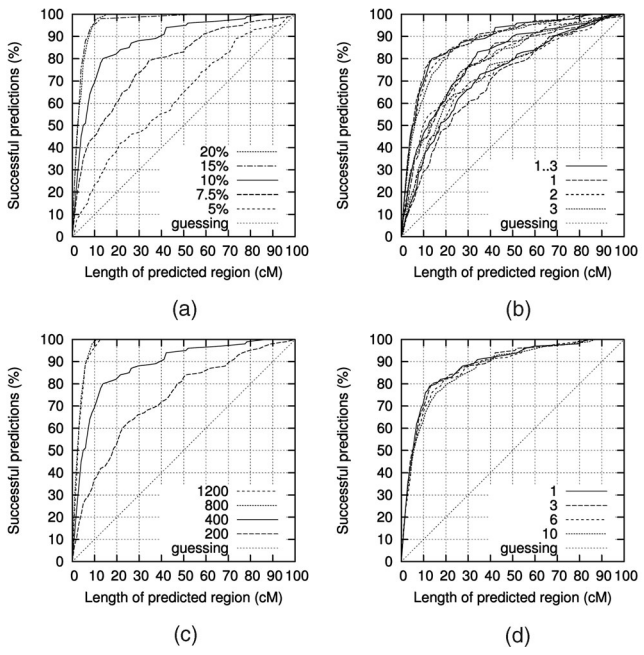

Fig. 6. Analysis of the performance of TreeDT. (a) Gene localization accuracy with different values of A, the proportion of disease-associated chromosomes that actually carry the mutation. (b) Gene localization accuracy with different numbers of subtrees (parameter of the method, given in the legend) and different numbers of founders (population parameter: 1 for the highest set of curves, 2 for the curves in the middle, and 3 for the lowest set of curves). (c) Gene localization accuracy with different sample sizes. (d) Gene localization accuracy with different maximum numbers of subtrees.

to note that, for each number of founders, the same number of subtrees gives marginally the best result. The upper limit of three subtrees gives consistently competitive results, so we continue using it in the following experiments.

As expected, the localization accuracy improves with increasing sample size, although increasing the sample sizes from 800 to 1,200 chromosomes does not significantly improve the results any more (Fig. 6c). We also evaluated the effect of the maximum number of subtrees in the baseline setting, where there was a single mutation-carrying founder. Decreasing the maximum number does improve the accuracy, as shown in Fig. 6d, because the number of tests on the lowest level decreases with it. However, tests with different numbers of subtrees are highly correlated and the differences in results are rather small. Furthermore, the number of mutation-carrying founders for a real data set is usually unknown, so it might not be wise to set too low a limit.

In gene mapping studies like the ones simulated above, it is assumed that a disease susceptibility gene is known to be present in the analyzed area, based on a preliminary analysis, e.g., by linkage analysis. TreeDT has the important advantage over plain gene localization methods that it can also be used to predict whether the analyzed region contains a disease susceptibility gene at all or not. The overall p value TreeDT produces indicates the corrected statistical significance of the most significantly disease-associated location and, by setting an upper limit for its value, TreeDT can be used to classify data sets to ones that do or do not contain a mutation. In order to verify the correctness of the permutation procedure, we generated 100 data sets where the disease-association statuses were randomly chosen for each individual, that is, there is no genetic contribution from the simulated chromosome. For these data sets, TreeDT should produce overall p values as well as local p values from the uniform distribution in $[0, 1]$. Fig. 7a shows the cumulative distribution of the observed overall p values on these data sets; for only 100 data points, the deviation from the diagonal is within expectations. The 10,200 local p values follow the uniform distribution very convincingly (Fig. 7b).

Smaller thresholds for the overall p value result in fewer false positives, but also in fewer true positives. Fig. 7c shows the experimental relationship, in the form of an ROC curve, between power (ratio true positives/all positives) and overall p (ratio of false positives/all negatives). For higher values of A, the classification accuracy is extremely good. However, for $A = 5\%$, the power to detect the DS gene is no better than random guessing, although the localization accuracy for a gene known to exist is still adequate in 20-30 percent of the cases (Fig. 6a).

6.3 Comparison to Other Methods

TreeDT, HPM, and m-TDT have practically identical performance in localizing the DS gene in the baseline setting (Fig. 8a). The parameters we used for HPM were χ^2-threshold 7, maximum pattern length 7, and one gap of at

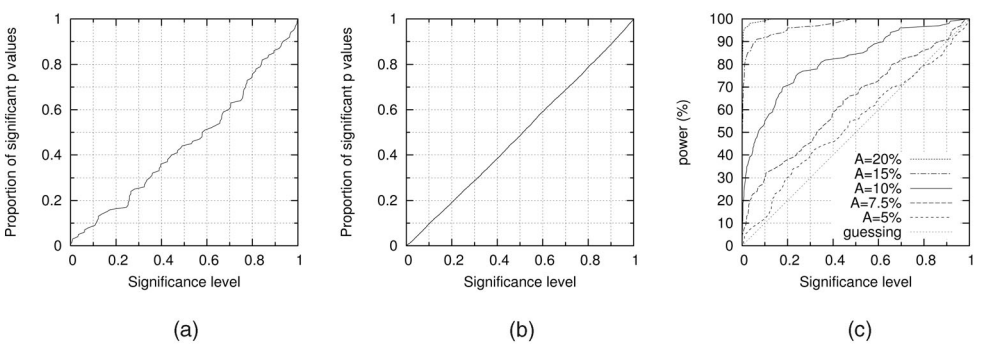

Fig. 7. (a) The cumulative distribution of overall p values on 100 data sets in which there were no DS genes. (b) The cumulative distribution of 10,200 local p values on the same data (pooled over all 102 tested locations per data set). (c) Statistical power to detect the existence of a disease susceptibility gene.

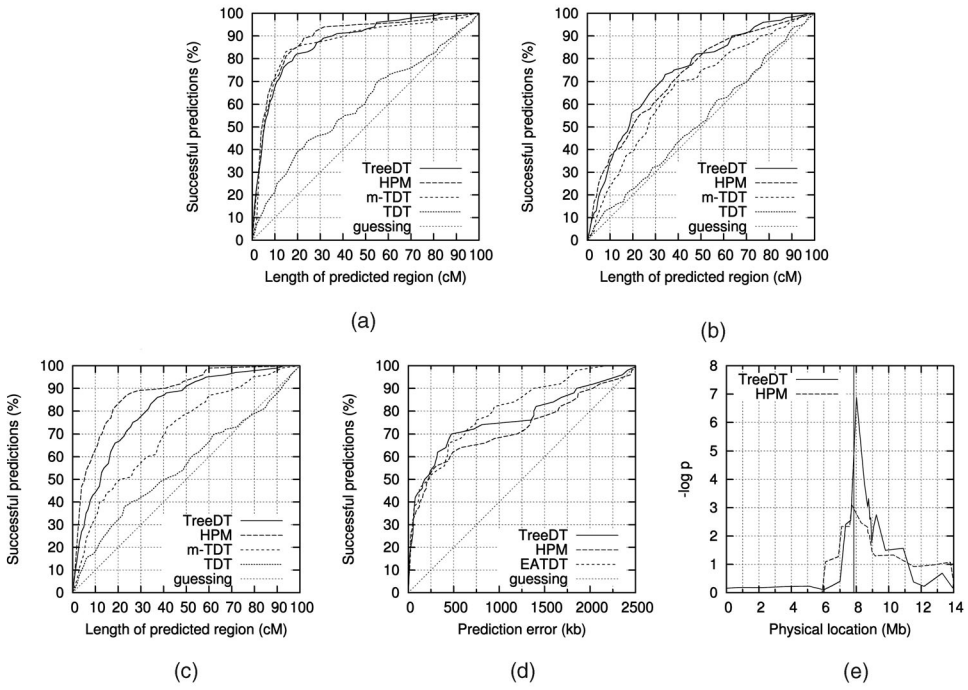

Fig. 8. Comparison of the gene localization accuracy of different methods. (a) The baseline test setting. (b) The baseline setting with three founders. (c) The baseline setting with 15 percent missing data. (d) SNP data of Lin et al. (e) Comparison of TreeDT and HPM on real Type 1 diabetes data. The known DS locus is denoted with a vertical line.

most one marker. Single marker TDT is clearly inferior compared to the other methods. Tests with other values of A or other sample sizes give similar results (Appendix C, which can be found on the Computer Society Digital Library at http://computer.org/tcbb/archives.htm).

In a test setting with three founders who introduced the mutation to the population, differences between the three best methods start to appear (Fig. 8b). TreeDT has an edge over HPM, which in turn performs better than m-TDT. TDT barely beats random guessing.

Next, we compare the methods with a large amount of missing data (Fig. 8c). We randomly removed 5 percent of the alleles in the genotype data in family trios, adding further complexity to the determination of parental origin. Consequently, almost 15 percent of the allele information in the resulting haplotypes is missing. Expectedly, HPM is most robust with respect to missing data since it allows gaps in its haplotype patterns. Surprisingly, TreeDT is not much weaker than HPM, although no actions have been taken in it to account for missing or erroneous data: unlike HPM and TDT, TreeDT considers missing allele value as just another allele. The performance of m-TDT degrades much more clearly as the implementation in GENEHUNTER2 does not include trios with missing alleles within a haplotype in the TDT test for the haplotype. Using a simple method for imputing values for the missing alleles improved the results of TreeDT to be on a par with HPM (Appendix C, which can be found on the Computer Society Digital Library at http://computer.org/tcbb/archives.htm). We were not able to run m-TDT on the imputed haplotypes using GENEHUNTER2, but, in principle, a similar improvement should be seen.

With the SNP data of Lin et al., we compare TreeDT to EATDT by the same authors and HPM. Because EATDT only reports the location of the most significant haplotype, we could only report the mapping performance as a function of the localization error for the point estimate. TreeDT and EATDT have equal performance up to ~700 kb prediction error, HPM being slightly less accurate (Fig. 8d). The result suggests that, with a reasonably dense marker map, TreeDT performs well also with bi-allelic SNP data and with data from general populations.

On the real Type 1 diabetes data, TreeDT pinpoints the known DS locus very convincingly using 10,000 permutations (Fig. 8e). HPM with the same number of permutations is able find the locus as well, but local p values given by TreeDT are much smaller and the extrapolation mechanism further highlights the predicted location. None of the permutations gave at least as small a lowest local p value as the lowest for the observed data. The overall corrected p value is thus $< 10^{-5}$.

The execution time of TreeDT for a simulated data set with 400 haplotypes is about one minute using 1,000 permutations on a 1400 MHz Pentium 4. The respective time for HPM with permutations is over 4 minutes.

7 Discussion and Conclusions

We have introduced TreeDT, a novel method for gene mapping. It is based on genealogical analysis of haplotypes and detects tree disequilibrium in the haplotype trees to the right and left of the disease susceptibility gene location. We have shown how tree disequilibrium can be efficiently evaluated between every pair of consecutive markers and

be subsequently tested for statistical significance using multiple, nested permutation tests with the cost of one.

Nowadays, the focus of gene mapping methodology is on complex diseases, where there are several genes and possibly environmental factors contributing to susceptibility. With complex diseases, the associations of individual genes are diluted by the effects of the other factors. Therefore, methods looking for individual susceptibility genes must be able to detect rather weak genetic effects. Toward this end, in the simulated data sets used in the experiments, only a small fraction (5-20 percent) of chromosomes carried the mutated allele, creating a mapping challenge similar to that of mapping individual genes for a complex disease.

One of the problems with real haplotype data is that there are missing alleles due to problems in allele calling and ambiguities in determination of the haplotypes. The algorithm as applied in this paper regards a missing allele as just another allele symbol. Although TreeDT does reasonably well even with 15 percent of the alleles missing in the haplotypes, we recommend that methods for inferring the missing alleles [26], [28], [29] be applied to the data prior to analysis, if possible.

Reconstruction of genealogies from haplotype data is a difficult problem. In TreeDT, the use of haplotype trees to estimate genealogical trees is based on the fact that the expected length of the haplotype sharing decreases as the time to the most recent common ancestor increases. However, the length of haplotype sharing has large variance and, consequently, the predictions of tree structure are not very reliable. As a result, the mutation-carrying subtree in the true genealogical tree may be split into a few subtrees in the estimated tree. Or, on the other hand, it may be merged with other subtrees, diluting the observable disequilibrium in the combined subtree. The results from our experiments suggest that, in most cases, the structure of the estimated tree is close enough to that of the true genealogical tree to allow for testing the disequilibrium with good power.

The tree disequilibrium test itself is not bound to any particular method for estimating the genealogical trees, so more elaborate estimation techniques could be used. Also, the test statistic for a given location (level 2) can be easily modified to accommodate any number of genealogy estimates for a location, e.g., a sample of estimates produced by some stochastic method can be used.

Our experiments show that TreeDT is effective in extreme conditions typical for current mapping projects: with lots of noise (only 10-20 percent of affected chromosomes carry the mutation) and with small sample sizes (200 affected and 200 control chromosomes). We also showed a successful mapping result using a real Type 1 diabetes data set. However, the highest potential of the method lies with data intensive tasks, such as genome-wide analyses with larger samples and larger number of markers, due to its low computational complexity.

While TreeDT was developed with isolated populations in mind, we showed that, with a sufficiently dense marker map, it can be effectively applied to data from general populations as well. We compared TreeDT to two other nonparametric methods suitable for analysis of genome-wide data: HPM and TDT. TreeDT is most competitive, especially in the presence of allelic heterogeneity. TreeDT can also be used to predict whether a gene is present at all or not. Unlike family-based association tests, such as TDT and its variants, TreeDT can also be applied to case-control samples of unrelated individuals for which haplotypes have been inferred using population-based haplotyping methods (e.g., [30], [31]). These features make it promising for haplotype mapping from the first, genome-wide analyses to more fine-grained analysis.

ACKNOWLEDGMENTS

The authors thank Päivi Onkamo, Juha Kere, Petteri Hintsanen, and Lauri Eronen for support and cooperation during this research. See http://www.cs.helsinki.fi/group/genetics/ for instructions on obtaining TreeDT. This work was supported in part by the Helsinki Graduate School in Computer Science and Engineering (HeCSE), Graduate School in Computational Biology, Bioinformatics, and Biometry (ComBi), and Tekes.

REFERENCES

[1] P. Sevon, H.T.T. Toivonen, and V. Ollikainen, "TreeDT: Gene Mapping by Tree Disequilibrium Test," *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* pp. 365-370, 2001.

[2] R. Miller, *Simultaneous Statistical Inference.* New York: McGraw-Hill, 1966.

[3] P. Westfall and S. Young, *Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment.* New York: Wiley, 1993.

[4] D. Knuth, *The Art of Computer Programming, Volume III—Sorting and Searching.* Reading, Mass. : Addison-Wesley, 1975.

[5] B. Devlin, N. Risch, and K. Roeder, "Disequilibrium Mapping: Composite Likelihood for Pairwise Disequilibrium," *Genomics,* vol. 36, pp. 1-16, 1996.

[6] L. Lazzeroni, "Linkage Disequilibrium and Gene Mapping: An Empirical Least-Squares Approach," *Am. J. Human Genetics,* vol. 62, pp. 159-170, 1998.

[7] M. McPeek and A. Strahs, "Assessment of Linkage Disequilibrium by the Decay of Haplotype Sharing, with Application to Fine-Scale Genetic Mapping," *Am. J. Human Genetics,* vol. 65, pp. 858-875, 1999.

[8] S. Service, D. Temple Lang, N. Freimer, and L. Sandkuijl, "Linkage-Disequilibrium Mapping of Disease Genes by Reconstruction of Ancestral Haplotypes in Founder Populations," *Am. J. Human Genetics,* vol. 64, pp. 1728-1738, 1999.

[9] J. Terwilliger, "A Powerful Likelihood Method for the Analysis of Linkage Disequilibrium between Trait Loci and One Ore More Polymorphic Marker Loci," *Am. J. Human Genetics,* vol. 56, pp. 777-787, 1995.

[10] A. Morris, J. Whittaker, and D. Balding, "Bayesian Fine-Scale Mapping of Disease Loci, by Hidden Markov Models," *Am. J. Human Genetics,* vol. 67, pp. 155-169, 2000.

[11] A. Morris, J. Whittaker, and D. Balding, "Fine-Scale Mapping of Disease Loci via Shattered Coalescent Modelling of Genealogies," *Am. J. Human Genetics,* vol. 70, pp. 686-707, 2002.

[12] B. Rannala and J. Reeve, "High-Resolution Multipoint Linkage-Disequilibrium Mapping in the Context of Human Sequence," *Am. J. Human Genetics,* vol. 69, pp. 159-178, 2001.

[13] J. Lam, K. Roeder, and B. Devlin, "Haplotype Fine-Mapping by Evolutionary Trees," *Am. J. Human Genetics,* vol. 66, pp. 659-673, 2000.

[14] R. Spielman, R. McGinnis, and W. Ewens, "Transmission Test for Linkage Disequilibrium: The Insulin Gene Region and Insulin-Dependent Diabetes Mellitus (IDDM)," *Am. J. Human Genetics,* vol. 52, pp. 506-516, 1993.

[15] L. Kruglyak, M. Daly, M. Reeve-Daly, and E. Lander, "Parametric and Nonparametric Linkage Analysis: A Unified Multipoint Approach," *Am. J. Human Genetics,* vol. 58, pp. 1347-1363, 1996.
[16] S. Lin, A. Chakravarti, and D. Cutler, "Exhaustive Allelic Transmission Disequilibrium Tests as a New Approach to Genome-Wide Association Studies," *Nature Genetics,* vol. 36, pp. 1181-1188, 2004.
[17] H. Toivonen, P. Onkamo, K. Vasko, V. Ollikainen, P. Sevon, H. Mannila, M. Herr, and J. Kere, "Data Mining Applied to Linkage Disequilibrium Mapping," *Am. J. Human Genetics,* vol. 67, pp. 133-145. 2000.
[18] H. Toivonen, P. Onkamo, K. Vasko, V. Ollikainen, P. Sevon, H. Mannila, and J. Kere, "Gene Mapping by Haplotype Pattern Mining," *Proc. IEEE Int'l Symp. Bio-Informatics and Biomedical Eng.,* pp. 99-108, 2000.
[19] P. Onkamo, V. Ollikainen, P. Sevon, H. Toivonen, H. Mannila, and J. Kere, "Association Analysis for Quantitative Traits by Data Mining: QHPM," *Annals of Human Genetics,* vol. 66, pp. 419-429, 2002.
[20] P. Sevon, H. Toivonen, and P. Onkamo, "Gene Mapping by Pattern Discovery," *Data Mining in Bioinformatics,* J. Wang, M. Zaki, H. Toivonen, and D. Shasha, eds., Springer, 2005.
[21] D. Qian, "Haplotype Sharing Correlation Analysis Using Family Data: A Comparison with Family-Based Association Test in the Presence of Allelic Heterogeneity," *Genetic Epidemiology,* vol. 27, pp. 43-52, 2004.
[22] K. Yu, C. Gu, M. Province, C. Xiong, and D. Rao, "Genetic Association Mapping under Founder Heterogeneity via Weighted Haplotype Similarity Analysis in Candidate Genes," *Genetic Epidemiology,* vol. 27, pp. 182-191, 2004.
[23] J. Tseng, "Evolutionary-Based Grouping of Haplotypes in Association Analysis," *Genetic Epidemiology,* vol. 28, pp. 220-231, 2005.
[24] Y. Ge, S. Dudoit, and T. Speed, "Resampling-Based Multiple Testing for Microarray Data Analysis," *TEST,* vol. 12, pp. 1-77, 2003.
[25] R. Hudson, "Generating Samples under a Wright-Fisher Neutral Model," *Bioinformatics,* vol. 18, pp. 337-338, 2002.
[26] S. Lin, A. Chakravarti, and D. Cutler, "Haplotype and Missing Data Inference in Nuclear Families," *Genome Research,* vol. 14, pp. 1624-1632, 2004.
[27] S. Bain, J. Todd, and J. Barnett, "The British Diabetic Association-Warren Repository," *Autoimmunity,* vol. 7, pp. 83-85, 1990.
[28] D. Qian and L. Beckmann, "Minimum-Recombinant Haplotyping in Pedigrees," *Am. J. Human Genetics,* vol. 70, pp. 1434-1445, 2002.
[29] J. Li and T. Jiang, "Efficient Inference of Haplotypes from Genotypes on a Pedigree," *J. Bioinformatics and Computational Biology,* vol. 1, pp. 41-69, 2003.
[30] M. Stephens, N. Smith, and P. Donnelly, "A New Statistical Method for Haplotype Reconstruction from Population Data," *Am. J. Human Genetics,* vol. 68, pp. 978-989, 2001.
[31] L. Eronen, F. Geerts, and H. Toivonen, "A Markov Chain Approach to Reconstruction of Long Haplotypes," *Proc. Pacific Symp. Biocomputing,* pp. 104-115, 2004.

Petteri Sevon received the MSc and PhD degrees in computer science from the University of Helsinki in 2000 and 2004, respectively. He is a postdoctoral researcher in the Department of Computer Science at the University of Helsinki, Finland. His research interests include data mining and computational biology.

Hannu Toivonen received the MSc and PhD degrees in computer science from the University of Helsinki in 1991 and 1996, respectively. He is a professor of computer science at the University of Helsinki, Finland. His research interests include knowledge discovery, data mining, and analysis of scientific data, with applications in genetics, bioinformatics, ecology, and mobile communications. He is an action editor of *Data Mining and Knowledge Discovery*. He regularly serves on the program committees of all major data mining conferences and many machine learning and artificial intelligence conferences. He was a program committee cochair for the ECML/PKDD conferences in 2002 and he is a founding cochair of the BIOKDD workshop series on data mining in bioinformatics (2001-).

Vesa Ollikainen received the MSc and PhD degrees in computer science from the University of Helsinki in 1997 and 2002, respectively. He currently works as a senior lecturer in information technology at Helsinki Polytechnic Stadia, Finland. His research interests include disease gene mapping, with a focus on simulation techniques.