Pacific Symposium on Biocomputing 14:415-426 (2009)

TREEQA: QUANTITATIVE GENOME WIDE ASSOCIATION MAPPING USING LOCAL PERFECT PHYLOGENY TREES

FENG PAN[1], LEONARD MCMILLAN[1], FERNANDO PARDO-MANUEL DE VILLENA[2], DAVID THREADGILL[2] AND WEI WANG[1]

[1]*Department of Computer Science,* [2]*Department of Genetics*
University of North Carolina at Chapel Hill
Chapel Hill, NC, USA
E-mail: [1]{*panfeng,mcmillan,weiwang*}*@cs.unc.edu,* [2]{*fernando,dwt*}*@med.unc.edu*

The goal of genome wide association (GWA) mapping in modern genetics is to identify genes or narrow regions in the genome that contribute to genetically complex phenotypes such as morphology or disease. Among the existing methods, tree-based association mapping methods show obvious advantages over single marker-based and haplotype-based methods because they incorporate information about the evolutionary history of the genome into the analysis. However, existing tree-based methods are designed primarily for binary phenotypes derived from case/control studies or fail to scale genome-wide.

In this paper, we introduce TreeQA, a quantitative GWA mapping algorithm. TreeQA utilizes local perfect phylogenies constructed in genomic regions exhibiting no evidence of historical recombination. By efficient algorithm design and implementation, TreeQA can efficiently conduct quantitative genom-wide association analysis and is more effective than the previous methods. We conducted extensive experiments on both simulated datasets and mouse inbred lines to demonstrate the efficiency and effectiveness of TreeQA.

1. Introduction

Genome wide association (GWA) mapping locates genes or narrows regions in the genome that have significant statistical connections to phenotypes of interest. The discovery of these genes and regions offers the potential to increase understanding of biological processes controlling manifestation of phenotypes.

The most frequent genetic variants are single nucleotide polymorphisms (SNPs), in which a single nucleotide in the genome differs between individuals within a species. With the development of low-cost genotyping technologies, extensive SNP data can be cheaply and efficiently produced, which further increases the computational complexity of GWA mapping. Thus, there is an evident need for fast and effective GWA mapping methods.

Existing methods of association mapping look for similarities among samples (chromosomes, haplotypes, etc.) that are correlated with the phenotypes. If strong associations are present, the variance of the phenotype within groups of similar samples is substantially smaller than the variance over all samples.

For example, in single marker-based [17,5] and haplotype-based association mapping [10,4,12], samples are grouped according to their genetic variation at a

Pacific Symposium on Biocomputing 14:415-426 (2009)

single marker or a set of markers. For case/control phenotypes, markers that can divide samples into (almost) pure classes are reported. Though these methods employ different strategies for grouping samples, the derived groups are evaluated without further consideration of the intergroup similarities or alternate groupings.

In observation of this, tree-based association methods [14,18,13] utilize phylogenies constructed over the samples. The phylogeny tree is a rich yet compact representation of genetic similarities of the samples. It provides sensible groupings of samples at multiple resolutions. However, the existing methods either handle only case/control phenotypes [14,18] or do not scale to GWA mapping [13].

In this paper, we introduce TreeQA, a tree-based quantitative GWA mapping algorithm. TreeQA utilizes local perfect phylogeny trees constructed in genomic regions exhibiting no evidence of historical recombination by the 4-gamete test [2]. Given a perfect phylogeny, TreeQA evaluates all implied groupings and finds the strongest associations to the phenotype. Furthermore, TreeQA can identify and remove outliers during association analysis.

A brute-force implementation consists of a double loop: for every phylogeny tree, and for every grouping represented by the tree, we conduct a separate ANOVA test to measure its association to the phenotype, and keep track of the best groupings and trees. This approach is inefficient and prone to multiple test errors[1]. Both the number of trees and number of groupings per tree can be very large[a]. This large number of possible groupings requires many ANOVA tests, which is not only expensive computationally, but also gives rise to spurious associations[b]. Thus, permutation tests are necessary to ensure the statistical significance of the discovered associations, which will further increase the computational burden.

TreeQA exploits the following properties: (1) Groupings generated from the same tree obey a partial order, thus allowing reuse of intermediate computations; (2) A grouping may be derived from different trees, but only need to be evaluated once; and (3) Different phenotype permutations may share a substantial number of common computations that need to be computed only once. Thus, TreeQA employs two prefix-tree structures [21] to organize all observed sample subsets and groupings to facilitate the caching and retrieval of reusable computations and guide the enumeration and evaluation of groupings. As a result, TreeQA is able to handle quantitative GWA mapping very efficiently and is more effective and robust in association mapping than previous methods.

2. Related Work

Single-feature association mapping [17,5] considers the sample groupings induced independently by each single marker. Statistical tests such as χ^2 and F-tests are used to measure the association between the phenotype and each grouping. These methods are computationally efficient, however, they do not utilize the additional

[a] For example, the number of trees can exceed tens of thousands in a chromosome-wide association study. And there are up to 2^{2n-2} groupings that can be generated from a tree of n samples.

[b] With ε error rate, the risk of reporting at least one spurious association from x tests is $1-(1-\varepsilon)^x$.

Pacific Symposium on Biocomputing 14:415-426 (2009)

information content carried by haplotypes over single markers.

To address this shortcoming, haplotype-based methods have been developed. HAM [15] considers combinations of three consecutive SNPs along the genome. QHPM [8] uses frequent pattern mining methods to find haplotype patterns in the data, upon which sample groupings are created and evaluated. HapMiner [10] clusters samples using consecutive subsets of markers, and then assess the phenotype's association strength.

The utility of local phylogenies in association mapping has been recently explored in TreeLD [13], Blossoc [14], and TreeDT [18]. These methods use trees to represent sample similarities. Their approach is to exhaustively examine all possible groupings implied by the given phylogenies without explicitly excluding any outliers. Both Blossoc and TreeDT assume simple categorical (binary) phenotypes. TreeLD handles quantitative phenotypes but is not scalable to GWA analysis.

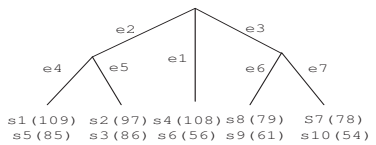
Some other work [6,7,16] uses a global phylogeny structure, e.g., ancestral recombination graph, over all markers in association mapping. However, because of the high computational cost of global phylogeny construction, these methods are not scalable to genome-wide analysis.

3. Preliminaries

We use a binary matrix $H = S \times M$ to represent a SNP dataset, where $S = \{s_1, s_2, ..., s_n\}$ is the set of samples, and $M = \{m_1, m_2, ..., m_z\}$ is the SNP marker set. Each sample is represented by a binary vector, in which '0' represents the majority alleles and '1' represents the minority alleles. We use $f(s_i)$ to denote the phenotype value of a sample s_i and $F(S')$ to denote the phenotype values of samples in a subset S'. An example matrix H containing 10 samples and 10 SNP markers with phenotype is shown in Fig. 1(a).

	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	f(s)
s1	1	0	1	0	0	0	0	0	0	0	109
s2	1	0	0	0	1	0	0	0	1	0	97
s3	1	0	0	0	1	0	0	0	0	0	86
s4	0	0	0	0	0	0	0	0	0	0	108
s5	1	0	1	0	0	0	0	0	0	0	85
s6	0	0	0	0	0	0	0	0	0	0	56
s7	0	0	0	0	0	1	0	1	0	0	78
s8	0	1	0	1	0	1	1	0	1	1	79
s9	0	1	0	1	0	1	1	0	1	0	61
s10	0	0	0	0	0	1	0	1	0	0	54

(a) SNP data & phenotype



(b) Tree $T_{1,8}$

Figure 1. Example: a SNP dataset and a perfect phylogeny tree

Compatible region: A consecutive region of the genome is called a compatible region *iff* any pair of markers in that region are compatible by the 4-gamete test [2]. That is, among the 4 possible haplotypes formed by the two markers, at most three of them occur.

A compatible region is a genomic region exhibiting no evidence of historical recombination. In Fig. 1(a), the region from markers m_1 to m_8 is a compatible region. We use $C_{u,v}$ to denote a compatible region from markers m_u to m_v.

Maximal Compatible region: A compatible region is a maximal compatible region *iff* it can not be extended on either side to include more SNPs and remains

Pacific Symposium on Biocomputing 14:415-426 (2009)

compatible.

Perfect Phylogeny Tree: A phylogeny tree for a set of samples is *perfect* if the phylogeny avoids homoplasy. Every SNP is introduced by a mutation and is represented by an edge of the tree. Given a genomic region, a perfect phylogeny exists *iff* the region is a compatible region.

We use $T_{u,v}$ to denote the perfect phylogeny tree of compatible region $C_{u,v}$. Given $C_{1,8}$ in Fig. 1(a), its tree $T_{1,8}$ is shown in Fig. 1(b). All samples are at the leaf nodes. Samples having identical haplotypes in the region share the same leaf node in the tree, e.g., s_1 and s_5. Each internal node represents a hypothetical common ancestor of a subset of samples. Each edge uniquely corresponds to a SNP (or a historical mutation). Interested readers may refer to paper [3] for inferring perfect phylogenies from a set of SNPs.

Let $E(T_{u,v}) = \{e_1, e_2, ..., e_p\}$ denote the set of edges in $T_{u,v}$. The removal of each edge partitions the samples into two subsets denoted by $S^{(0)}(e_i)$ and $S^{(1)}(e_i)$. Given a tree $T_{u,v}$, we can generate $2|E(T_{u,v})|$ sample subsets by removing each edge separately. We denote this set of sample subsets by $S^{(E)}(T_{u,v})$, $S^{(E)}(T_{u,v}) = \{S^{(j)}(e_i)|j = \{0,1\}, e_i \in E(T_{u,v})\}$.

Definition 3.1. A grouping of a sample subset S', $G(S')$, is formed by a set of disjoint subsets of S', $G(S') = \{S'_1, S'_2, ..., S'_k\}, S'_i \subset S', S'_i \cap S'_j = \emptyset, \bigcup_{i=1}^{k} S'_i = S'$. Given a tree $T_{u,v}$, we say a grouping $G(S')$ *follows* $T_{u,v}$ *iff* $\forall S'_i \in G(S')$, $S'_i \in S^{(E)}(T_{u,v})$.

For example, grouping $G(S') = \{\{s_1, s_5, s_2, s_3\}, \{s_8, s_9, s_7, s_{10}\}\}$ follows the tree in Fig. 1(b), while grouping $G(S') = \{\{s_1, s_2\}, \{s_8, s_4\}\}$ does not.

Definition 3.2. Given a sample subset S', $G_1(S')$ is called a *parent-grouping* of $G_2(S')$ ($G_2(S')$ called a *child-grouping* of $G_1(S')$) *iff* $\forall S'_i \in G_1(S')$

$$\exists S'_j \in G_2(S'), s.t. S'_i = S'_j.\ OR\ \exists \{S'_{j_q} | S'_{j_q} \in G_2(S'), q = 1, ..., u\}, s.t. S'_i = \bigcup_{q=1}^{u} S'_{j_q}$$

A child-grouping represents a finer partition of its parent-grouping on the same set of samples. For example, grouping $\{\{s_1, s_5, s_2, s_3\}, \{s_4, s_6\}\}$ is the parent-grouping of $\{\{s_1, s_5\}, \{s_2, s_3\}, \{s_4, s_6\}\}$. We summarize the notations in Table 1.

Association between a Compatible Region and a Phenotype

We use the one-way ANOVA test with permutations to measure the association between a grouping of samples and a quantitative phenotype. To accelerate the execution, we re-derive the formula of the ANOVA test.

Given a grouping $G(S') = \{S'_1, ..., S'_k\}$, for every $S'_i \in G(S')$, we calculate

$$SQ(S'_i) = \sum_{s_j \in S'_i} f(s_j)^2, \quad SM(S'_i) = \sum_{s_j \in S'_i} f(s_j) \tag{1}$$

Table 1. Summary of Notations

S, s_i, S'_i	the sample set, a sample, a subset of samples
M, m_i	the marker set, a marker
H	a binary matrix representing the data
$C_{u,v}$	a compatible interval of H
$f(s_i)$	phenotype value of sample s_i
$F(S'_i)$	the set of phenotype values of the samples in S'_i
$G_i(S')$	a grouping of a sample subsets S'
$T_{u,v}$	the perfect phylogeny tree of $C_{u,v}$
$E(T_{u,v})$	the edge set of $T_{u,v}$
$S^{(E)}(T_{u,v})$	the set of sample subsets implied in tree $T_{u,v}$ (leaf-sets)

$$SSE_i = SQ(S'_i) - SM(S'_i)^2/|S'_i|, \quad SSB_i = SM(S'_i)^2/|S'_i| \tag{2}$$

Combining all subsets together, we have $MM = \frac{1}{|S'|}\sum_{i=1}^{k} SM(S'_i)$ and

$$MSE = \frac{1}{|S'|-k}\sum_{i=1}^{k} SSE_i, \quad MSB = \frac{1}{k-1}(\sum_{i=1}^{k} SSB_i - |S'|\cdot MM^2) \tag{3}$$

We obtain a base score for grouping $G(S')$

$$F_0(G(S')) = \frac{MSB}{MSE} \tag{4}$$

A higher score indicates a stronger association between the grouping and the phenotype. Given the tree and the data in Fig. 1 and the following two groupings: $G(S'_1) = \{\{s_2, s_3\}, \{s_4, s_6\}, \{s_8, s_9\}\}$, $G(S'_2) = \{\{s_2, s_3\}, \{s_8, s_9\}\}$, the scores are $F_0(G(S'_1)) = 0.44$, $F_0(G(S'_2)) = 4.16$. Thus, grouping $G(S'_2)$ has a stronger association with the phenotype than grouping $G(S'_1)$.

To correct the multiple test errors, we apply a permutation test on $G(S')$ to calculate a significance score. To permute the phenotype, the phenotype values in $F(S')$ are randomly re-assigned to samples in S'. Then we calculate an F-score using the permuted phenotype following Eqs. 1 to 4.

Assume that we conduct *nPerm* random permutations in total, for each permutation, we get score $F_j (j = 1...nPerm)$. Among the $nPerm$ F-scores, let p be the number of scores which are greater than or equal to the base score $F_0(G(S'))$, i.e., $p = |\{F_j | F_j \geq F_0(G(S')), j \in 1...nPerm\}|$. Then the significant score (P score) of $G(S')$ is

$$P(G(S')) = \log_{10}\left(\frac{nPerm}{p}\right) \tag{5}$$

A higher P score indicates that the association between grouping $G(S')$ and the phenotype is more significant.

Definition 3.3. The association between a compatible region and a phenotype: For a compatible region $C_{u,v}$, the highest P score achieved by any grouping following $T_{u,v}$ is regarded as the P score of $C_{u,v}$. The P score represents the

Pacific Symposium on Biocomputing 14:415-426 (2009)

association between the compatible region and the phenotype,

$$P(C_{u,v}) = max\{P(G_j(S'))|\forall G_j(S') \ follows \ T_{u,v}, S' \subseteq S\}. \quad (6)$$

Problem Definition: Given a SNP data and a quantitative phenotype, calculate the P-score of every maximal compatible region and report the most significant ones.

4. TreeQA Algorithm

TreeQA takes two major steps: 1) identify maximal compatible regions in the genome and construct the perfect phylogenies of the regions; 2) compute the association between each compatible region and the phenotype.

4.1. *Maximal Compatible Region and Phylogeny Construction*

TreeQA scans the markers in a left to right order. In order to find the maximal compatible regions, it continuously extends the current region by adding the next marker until the new marker is incompatible with some markers in the region. And it maximizes the overlap between two consecutive regions. Assume that the current compatible region is $C_{u,v}$, and marker m_{v+1} is incompatible with markers $m_{i_1}, ..., m_{i_k}$, $u \leq i_1 < ... < i_k \leq v$, then TreeQA starts the next compatible region at marker m_{i_k+1}. For each maximal compatible region, TreeQA utilizes the inferring algorithm [3] to construct the local perfect phylogeny.

4.2. *Association Computing*

In the second step, TreeQA takes as input a quantitative phenotype and a set of local perfect phylogenies. It considers all possible groupings following the phylogenies and systematically explores the search space of these groupings in a carefully designed order such that intermediate computations can be maximally reused.

According to Definition 3.1, any grouping of a sample subset[c] that follows a tree $T_{u,v}$ can be created from non-overlapping subsets in $S^{(E)}(T_{u,v})$. By utilizing the lexicographical order[d] of subsets in $S^{(E)}(T_{u,v})$, TreeQA can enumerate and evaluate all combinations of non-overlapping subsets systematically.

TreeQA enumerates all groupings via a depth-first recursive procedure. TreeQA extends the current grouping by including a new sample subset which does not overlap with any subsets in the current grouping. The association of each new grouping to the phenotype via a permutation test is computed. The P score of the corresponding maximal compatible region is updated accordingly. The enumeration continues recursively for each newly extended grouping.

Consider the tree in Figure 1. There are 14 sample subsets in $S^{(E)}(T_{1,8})$. Assume that the subsets have the following order,

[c] Considering groupings of a sample subset allows TreeQA to exclude potential outliers from the ANOVA test.

[d] Any other ways of defining a total order of the subsets would also work.

Pacific Symposium on Biocomputing 14:415-426 (2009)

$se_1 = \{s_1, s_5\}, se_2 = S - se_1, se_3 = \{s_2, s_3\}, se_4 = S - se_3, se_5 = \{s_4, s_6\}$
$se_6 = S - se_5, se_7 = \{s_8, s_9\}, se_8 = S - se_7, se_9 = \{s_7, s_{10}\}, se_{10} = S - se_9$
$se_{11} = \{s_1, s_5, s_2, s_3\}, se_{12} = S - se_{11}, se_{13} = \{s_8, s_9, s_7, s_{10}\}, se_{14} = S - se_{13}$

TreeQA first generates a grouping containing se_1 only. Among the remaining sample subsets, $\{se_2, se_3, se_5, se_7, se_9, se_{12}, se_{13}\}$ do not overlap with se_1. In the next step, a grouping $\{se_1, se_2\}$ is formed by adding se_2 into the current grouping and its P score is calculated. $P(C_{1,8})$ is updated accordingly. Since all other sample subsets overlap with se_1 or se_2. Thus, no new grouping can be extended from $\{se_1, se_2\}$. Then, TreeQA examines the next grouping extended from $\{se_1\}$, $\{se_1, se_3\}$, and all groupings extended from it. After examining all groupings containing se_1, TreeQA will start from the grouping $\{se_2\}$ and extend it recursively to generate all groupings containing se_2 but not se_1. This process continues until all distinct groupings are enumerated.

4.3. *Effective Permutation*

We found that more than 90% of the execution time of TreeQA is spent in permutation tests. Given a grouping $G(S')$, a permutation test is conducted in two steps: 1) randomly re-assigning the phenotype values in $F(S')$ to samples in S'; 2) calculating the corresponding F score by Eq. 4.

Given a subset S', both steps take $O(|S'|)$ time. TreeQA exploits maximal reusability of intermediate computation shared by permutation through the following two optimizations:

1) ***inTree***: Common computation units shared by permutation tests of parent/child-groupings in a tree.

2) ***amgTree***: Common computation units shared by permutation tests on groupings following multiple trees.

We use two global prefix-tree structures [21], $Tree_{grouping}$ and $Tree_{subset}$ to organize groupings and sample subsets examined thus far respectively to enable effective permutation tests.

4.3.1. *inTree: Effective permutation tests within a tree*

A pair of parent/child-groupings always involve the same set of samples. Let S' denote a set of samples. For the permutation tests of the parent/child groupings of S', instead of re-assigning the phenotype values in $F(S')$ independently for each grouping, they can share the same set of random permutations of $F(S')$.

For example, given the example in Fig. 1 and a pair of parent/child-groupings, $G_1(S') = \{\{s_1, s_5, s_2, s_3\}, \{s_8, s_9, s_7, s_{10}\}\}$ and $G_2(S') = \{\{s_1, s_5\}, \{s_2, s_3\}, \{s_8, s_9, s_7, s_{10}\}\}$, their F_0 scores are: $F_0(G_1(S')) = 9.79$ and $F_0(G_2(S')) = 4.32$. Assume that after a random permutation, the new phenotype values for the samples are: $f(s_1) = 85$, $f(s_2) = 79$, $f(s_3) = 109$, $f(s_5) = 61$, $f(s_7) = 86$, $f(s_8) = 97$, $f(s_9) = 78$, $f(s_{10}) = 54$. Using this new assignment, we can calculate the new F scores for both groupings: $F(G_1(S')) = 0.12$ and $F(G_2(S')) = 0.7$. By reusing the phenotype permutation

between $G_1(S')$ and $G_2(S')$, we save $O(|S'|)$ runtime in each permutation.

A child-grouping represents a finer partition of sample subsets in its parent-grouping. We say a grouping is at the finest level if it does not have any child-groupings. We use a global prefix-tree $Tree_{grouping}$ to index all groupings and maintain the parent/child relationship through auxiliary links (from a child-grouping to its parent-groupings). For each permutation of the phenotype, the F scores of a finest grouping and all of its parent-groupings are calculated together. We examine the finest grouping immediately followed by the examination of its parent groupings for maximum computation reuse. If a finest child-grouping has n parent-groupings, we save $O(n|S'|)$ time in each permutation.

4.3.2. *amgTree: Effective permutation among trees*

The same grouping occurs repeatedly in different trees. We only need to compute its P score at its first occurrence. We use $Tree_{grouping}$ to store and retrieve the P score of all examined groupings. If the grouping formed by TreeQA can be found in $Tree_{grouping}$, its P score is directly used. Otherwise, its P score is calculated and stored in $Tree_{grouping}$.

4.4. *Reuse of Intermediate Computation of Statistical Tests*

For any sample subset S', $SQ(S')$ and $SM(S')$ calculated using the original phenotype values (with no permutation) may be reused in any grouping containing S' and all its parent-groupings. We denote them by $SQ_0(S')$ and $SM_0(S')$ respectively in the following discussion.

We employ a global prefix-tree $Tree_{subset}$ to keep track of all sample subsets in any groupings examined thus far. Three values are stored at the leaf node corresponding to the subset S': (subset ID, $SQ_0(S')$, $SM_0(S')$).

For example, given the 10 samples and their phenotype values in Fig. 1(a), we calculate the base score F_0 of grouping $G_1(S') = \{\{s_1, s_5\}, \{s_2, s_3\}, \{s_7, s_{10}\}\}$.

$SQ_0(S'_{1_1}) = 19106, SQ_0(S'_{1_2}) = 16805, SQ_0(S'_{1_3}) = 9000.$
$SM_0(S'_{1_1}) = 194, SM_0(S'_{1_2}) = 183, SM_0(S'_{1_3}) = 132.$
$F_0(G_1(S')) = 547.17/212.17 = 2.58.$

The SQ_0 and SM_0 values of the three subsets are then stored in $Tree_{subset}$. Given a parent-grouping of $G_1(S')$, $G_2(S') = \{\{s_1, s_5, s_2, s_3\}, \{s_7, s_{10}\}\}$, we can retrieve the values of SQ_0 and SM_0 and use them to calculate $F_0(G_2(S'))$,

$SQ_0(S'_{2_1}) = SQ_0(S'_{1_1}) + SQ_0(S'_{1_2}) = 35911, SQ_0(S'_{2_2}) = SQ_0(S'_{1_3}).$
$SM_0(S'_{2_1}) = SM_0(S'_{1_1}) + SM_0(S'_{1_2}) = 377, SM_0(S'_{2_2}) = SM_0(S'_{1_3}).$
$F_0(G_2(S')) = 1064.08/166.69 = 6.38.$

The reuse of $SQ_0(S')$ and $SM_0(S')$ between parent/child groupings may work in conjunction with the *inTree* effective permutation. Besides, $SQ_0(S')$ and $SM_0(S')$ can also be reused by any groupings that contain the subset S'.

Pacific Symposium on Biocomputing 14:415-426 (2009)

5. Results

We compare TreeQA with the following algorithms: 1) **SMA**, our implementation of the Single Marker Association algorithm [17,5]; 2) **HAM**, our implementation of the Haplotype Association Mapping algorithm [15] that slides a 3-SNP window through the genome; 3) **HapMiner** [10], downloaded from the website[e]; and 4) **TreeLD** [13], downloaded from the website[f]. Both **SMA** and **HAM** use the one-way ANOVA test for fair comparison.

QHPM [8] is not used for comparison because it is not scalable to large data sets. Blossoc[14] and TreeDT [18] are not used because they require categorical phenotypes.

5.1. *Experiments on Simulated Data*

We use Coasim [11] to simulate 1000 sequences with scaled recombination rate $\rho = 400$ that corresponds roughly to 10 cM. 10,000 SNP markers are placed uniformly at random over the sequences.

SNP markers on the sequences are randomly selected as causative loci with one, two and three causative mutations. The first SNP is always selected randomly from all SNPs. In the cases of two and three mutations, the second and third causative SNPs are selected from compatible SNPs that are located less than 10 SNPs away from the first SNP. Phenotype values are sampled from four Gaussian distributions: $N_1(140, 35)$, $N_2(90, 35)$, $N_3(50, 40)$, and $N_4(10, 35)$. The one-mutation case uses N_1 and N_3. The two-mutation case uses N_1, N_2 and N_3. The three-mutation case uses all four Gaussian distributions. After assigning the phenotype values, all causative SNPs are removed from the data and we randomly select 100 sequences for our experiments.

SMA, HAM and HapMiner output the top one scoring locus as a point estimation of the causative locus, while TreeQA outputs the top one compatible region. We compare the effectiveness of the algorithms by measuring the distance (in cM) from the top one scoring locus or the center of the top one region to the causative SNP (or the average distance to every causative SNP). We call the distance the **Prediction error**.

Since HapMiner can not finish processing 10,000 SNP markers in a reasonable time, we only use the first 1,000 markers of each sequence when applying HapMiner on the simulated data.

The comparison of SMA, HAM, HapMiner and TreeQA is shown in Figure 2. The x-axis represents the prediction error (distance) to the causative locus and the y-axis represents the percentage of causative loci which are found in distance less than x. In all three cases, the estimated loci by TreeQA are closer to the causative loci than those by SMA, HAM and HapMiner.

The TreeLD algorithm uses local phylogenies and analyzes quantitative phenotypes. However, TreeLD can only process a very small amount of data in rea-

[e] http://vorlon.case.edu/ jxl175/HapMiner.html
[f] http://pritch.bsd.uchicago.edu/treeld.html

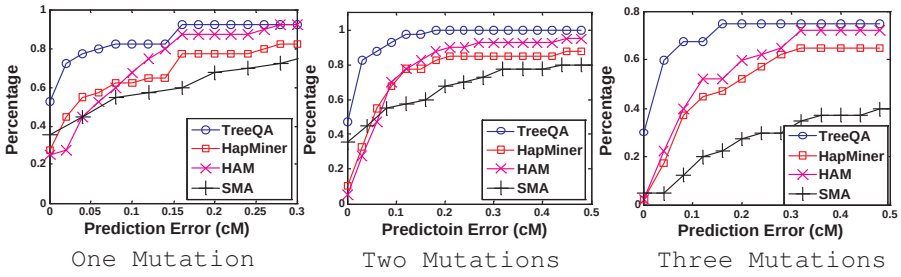


Figure 2. Comparison of SMA, HAM, HapMiner and TreeQA on the simulated data

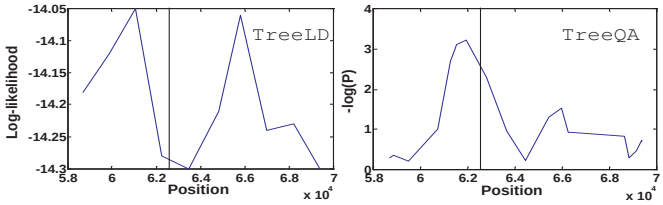


Figure 3. Comparison of TreeLD and TreeQA on the simulated data

sonable time. Therefore, we select 36 samples and 20 SNP markers from the simulated data for performance comparison. A one-mutation causative locus is selected from the 20 SNPs. For TreeQA, instead of generating maximal compatible regions as discussed in Sec. 4, a compatible region is generated around each SNP and contains up to five SNPs. TreeLD takes about two hours to analyze this small data while TreeQA finishes in seconds. Figure 3 plots the results from TreeLD and TreeQA. The x-axis represents the simulated positions in the genome and the y-axis represents the scores of the SNPs. The vertical line demonstrates the causative locus. TreeQA detects a peak near the causative locus while TreeLD identifies two spurious peaks.

5.2. *Experiments on Mouse Genotype Data*

We used a set of mouse genotypes that combines experimental and imputed data[g] [20] from the Jackson Laboratory, consisting of 74 samples. The dataset contains over 7 million SNP markers distributed over all 20 chromosomes. We removed wild derived mouse inbred strains since they are quantitatively and qualitatively different than other laboratory inbred strains and we only used in our experiments the remaining 55 samples that have a share set of common ancestral relationships[19].

We used high density lipoprotein cholesterol (HDL-C) levels in blood as the test phenotype, downloaded from the Mouse Phenome Database[h]. Several HDL-C datasets are available, each of which was collected under different conditions,

[g] http://cgd.jax.org/ImputedSNPData/imputedSNPs.htm
[h] http://phenome.jax.org/pub-cgi/phenome/mpdcgi?rtn=meas/catlister/req=Cblood+lipids

and are thus treated as separate phenotypes. Some candidate genes that may play a role in regulating HDL-C levels are reported in [9].

We apply SMA, HAM and TreeQA on the data and examine how close they can identify the top peak near the locus of those candidate genes.

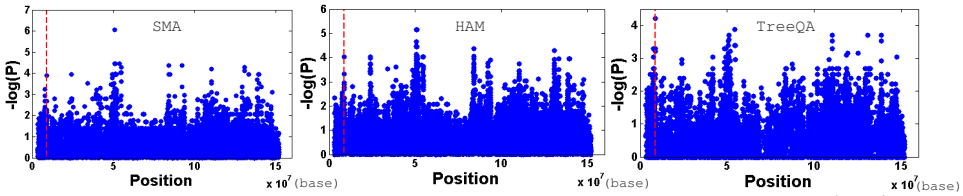


Figure 4. Compare SMA, HAM and TreeQA on the mouse genotype data

TreeQA detects top peaks near the locations for over 10 of the candidate genes[9], including *Ppara*, *Abcb4* and *Rxrb*. The top peaks reported by SMA and HAM are often far from the locations of these genes. Due to space limitation, we only show the results for one of them, *Abcb4*, in Figure 4.

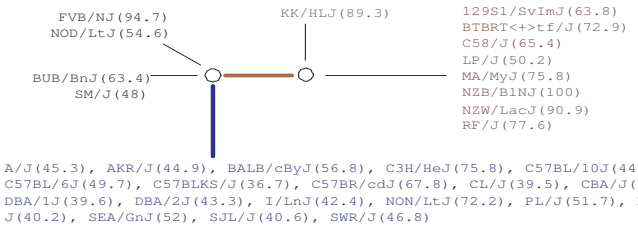


Figure 5. The perfect phylogeny at the peak point found by TreeQA in Figure 4

The perfect phylogeny corresponding to the peak point (compatible region from 8799298 to 8801558 (base)) found by TreeQA is plotted in Fig. 5. The phenotype values of the samples are in parentheses. Samples with unknown phenotype values are omitted from the tree. The subtree on the right contains samples having high phenotype values while the subtree at the bottom contains samples having low values. Other subtrees are considered as outliers and are excluded from the grouping. SMA and HAM fail to identify the locus because they only examine sample groupings that can be generated from single SNPs or 3-SNP windows, which are a small subset of the groupings examined by TreeQA.

TreeQA takes about 10 minutes to analyze each chromosome which contains around 40000 SNPs on average. SMA and HAM take slightly less time than TreeQA. Both HapMiner and TreeLD are unable to finish in reasonable time.

6. Conclusion

In this paper, we present a tree-based quantitative GWA mapping algorithm, TreeQA. TreeQA utilizes local perfect phylogenies in detecting associations. Per-

fect phylogenies provide sensible groupings of samples at multiple resolutions. TreeQA explores the space of all possible groupings implied by the perfect phylogenies in a carefully designed order so that intermediate computations can be maximally reused. Our experimental results on both simulated and real data show that TreeQA can efficiently conduct quantitative GWA analysis and is more effective than the previous methods.

References

1. R. G. Miller. *Simultaneous Statistical Inference*. Springer Verlag New York, 1981.
2. R. R. Hudson and N. L. Kaplan. Statistical properties of the number of recombination events in the history of a sample of dna sequences. *Genetics*, 111(1):147C164, 1985.
3. R. Agarwala, D. Fernandez-Baca, and G. Slutzki. Fast algorithms for inferring evolutionary trees. *Journal of Computational Biology*, 2(3):397–408, 1995.
4. H. Toivonen, P. Onkamo, K. Vasko, and et al. Data mining applied to linkage disequilibrium mapping. *Am J Hum Genet*, 67:133–145, 2000.
5. J. Akey, L. Jin, and M. Xiong. Haplotypes vs single marker linkage disequilibrium tests: what do we gain? *Eur J. Hum Genet.*, 9(4):291–300, 2001.
6. F. Larribea, S. Lessarda, and N. J. Schork. Gene mapping via the ancestral recombination graph. *Theoretical Population Biology*, 62(2):215–229, 2002.
7. A. P. Morris, J. C. Whittaker, and D. J. Balding. Fine-scale mapping of disease loci via shattered coalescent modeling of genealogies. *Am J Hum Genet*, 70(3), 2002.
8. P. Onkamo, V. Ollikainen, P. Sevon, and et al. Association analysis for quantitative traits by data mining: Qhpm. *Ann. Hum. Genet.*, 66:419–429, 2002.
9. X. Wang and B. Paigen. Quantitative trait loci and candidate genes regulating hdl cholesterol. *Arteriosclerosis, Thrombosis, and Vascular Biology*, 22:1390, 2002.
10. J. Li and T. Jiang. Haplotype-based linkage disequilibrium mapping via direct data mining. *Bioinformatics*, 21(24):4384–4393, 2005.
11. T. Mailund, M. H. Schierup, and et al. Coasim: A flexible environment for simulating genetic data under coalescent model. *BMC Bioinformatics*, 6(252), 2005.
12. E. Waldron, J. Whittaker, and D. Balding. Fine mapping of disease genes via haplotype clustering. *Genetic Epidemiology*, 30:170–179, 2005.
13. S. Zöllner and J. K. Pritchard. Coalescent-based association mapping and fine mapping of complex trait loci. *Genetics*, 169(2):1071–92, 2005.
14. T. Mailund, S. Besenbacher, and M. H. Schierup. Whole genome association mapping by incompatibilities and local perfect phylogenies. *BMC Bioinformatics*, 7:454, 2006.
15. P. McClurg, M. T. Pletcher, T. Wiltshire, and A. I. Su. Comparative analysis of haplotype association mapping algorithms. *BMC Bioinformatics*, 7:61, 2006.
16. M. J. Minichiello, and R. Durbin. Mapping Trait Loci by Use of Inferred Ancestral Recombination Graphs. *Am J Hum Genet*, 79(5):910–922, 2006.
17. I. Pe'er and et al. Evaluating and improving power in whole-genome association studies using fixed marker sets. *Nature Genetics*, 38:663–667, 2006.
18. P. Sevon, H. Toivonen, and V. Ollikainen. Treedt: Tree pattern mining for gene mapping. *IEEE Transactions on Computational Biology and Bioinformatics*, 3(2), 2006.
19. H. Yang, T. A. Bell, G. A. Churchill, and F. P.-M. de Villena. On the subspecific origin of the laboratory mouse. *Nature Genetics*, 39, 2007.
20. J. P. Szatkiewicz, G. L. Beane, Y. Ding, L. Hutchins, F. P.-M. de Villena, and G. A. Churchill. An imputed genotype resource for the laboratory mouse. *Mammalian Genome*, 19(3), 2008.
21. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Introduction to algorithms.