Quantitative Association Analysis Using Tree Hierarchies

Feng Pan [1], Lynda Yang [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
[1]{panfeng,lynda,mcmillan,weiwang}@cs.unc.edu, [2]{fernando, dwt}@med.unc.edu

Abstract

Association analysis arises in many important applications such as bioinformatics and business intelligence. Given a large collection of measurements over a set of samples, association analysis aims to find dependencies of target variables to subsets of measurements. Most previous algorithms adopt a two-stage approach; they first group samples based on the similarity in the subset of measurements, and then they examine the association between these groups and the specified target variables without considering the inter-group similarities or alternative groupings. This can lead to cases where the strength of association depends significantly on arbitrary clustering choices.

In this paper, we propose a tree-based method for quantitative association analysis. Tree hierarchies derived from sample similarities represent many possible sample groupings. They also provide a natural way to incorporate domain knowledge such as ontologies and to identify and remove outliers. Given a tree hierarchy, our association analysis evaluates all possible groupings and selects the one with strongest association to the target variable. We introduce an efficient algorithm, TreeQA, to systematically explore the search-space of all possible groupings in a set of input trees, with integrated permutation tests. Experimental results show that TreeQA is able to handle large-scale association analysis very efficiently and is more effective and robust in association analysis than previous methods.

1 Introduction

Association analysis aims to find the dependency of a target variable to subsets of features from a set of samples. The problem has many applications in bioinformatics. In genotype-phenotype association studies, biologists look for genetic variations that best explain and/or predict phenotypic values, such as blood pressure or susceptibility to disease. Traditional data mining tasks such as feature selection based on consistency driven filters [3, 7] can also be considered as variants of association analysis, where features that can predict the class label are detected.

Existing methods look for similarities among samples that are correlated with the target variable.

- In genetic association studies [6, 9], similarities between individuals are measured according to genetic differences in subsets of genes. If the phenotype values are similar in each group of similar individuals, the set of genes are correlated with the phenotype.
- In feature selection based on consistency driven filters, algorithms such as FOCUS [3] and LVF [7] divide objects by their identities/similarities in subsets of features. If most of the objects in a group have the same label, the features are correlated with the class label.

Even though these methods enumerate feature subsets in different ways, for a given feature subset, all of them take two steps: (1) grouping together similar samples and (2) examining associations between these derived groups and the target variable using statistical tests such as χ^2-test or ANOVA (ANalysis Of VAriance) test. In the first step, groupings are generated according to some similarity metric. Even with the same metric, groupings may vary due to the choice of clustering thresholds and/or cut-off parameters. In the second step, these derived groups are evaluated without further consideration of the original intergroup similarities or alternative clusterings. This can lead to cases where the strength of association depends significantly on arbitrary parameter choices.

In this paper, we propose a tree-based approach to replace the strict sample groupings in previous approaches for association analysis. Similarities between samples can be naturally represented using tree hierarchies, which are the input to our algorithm. Within this tree hierarchy, we assume that samples are leaves. Similar samples share common ancestors in the tree. Tree hierarchies are a rich yet compact representation of sample similarities. Each tree edge corresponds to a bisection of the samples. The removal of any subset of edges generates a distinct grouping of the samples. Thus a tree hierarchy represents many possible sample groupings ($2^{|E|}$). Given tree hierarchies for the feature subsets, our method evaluates the association of a forest of trees to a given target variable. For each tree hierarchy, our method considers all possible sample clusterings implied by the removal of tree edges and finds the strongest associations to the target variable. No grouping needs to be explicitly generated prior to association analysis.

A brute-force implementation consists of a double loop:

for every tree and for every grouping represented by the tree, we conduct a ANOVA test to measure its association to the target variable, and keep track of the best groupings and trees. This approach is inefficient and prone to multiple test errors [10]. Both the number of trees and number of groupings per tree can be very large. This large number of possible groupings requires many ANOVA tests, which is not only expensive computationally, but also gives rise to spurious associations[1]. Thus, permutation tests are necessary to ensure the statistical significance of the discovered associations, which further increase the computational burden.

We introduce an efficient algorithm, TreeQA, for systematically exploring the search space of all possible groupings in a set of input trees, with integrated permutation tests. TreeQA exploits the following properties: (1) The groupings generated from the same tree obey a partial order, which allows the reuse of intermediate computations. (2) Different trees may lead to the same groupings and the duplicate computation should be identified and avoided. (3) The permutation tests are the most time consuming procedure. However, different variable permutations often share substantial common computations. As a result, TreeQA is able to handle large-scale association analysis very efficiently and is more effective and robust than previous methods.

2 Related Work

Association analysis is important in bioinformatic applications such as genome wide association mapping. Many algorithms have been developed in this area. Single-feature association mapping [2] considers the sample groupings induced independently by each single feature. Statistical tests such as χ^2 and F-tests are used to measure the association between the target variable and each grouping. More advanced methods which examine feature combinations have also been developed, such as HAM [9], QHPM [11] and HapMiner [6]. Many of these methods were originally designed for genotype-phenotype association studies and only considered genotypes in consecutive regions.

The utility of tree hierarchies in association mapping has been recently explored in Blossoc [8] and TreeDT [12]. Both methods use trees to represent sample similarities, but they assume simple categorical (binary) target variables. Their approach is to exhaustively examine all possible groupings without explicitly considering properties of the sample subspaces that are implied by the given tree structures.

In addition, feature selection algorithms based on consistency driven filters were developed to handle more general data and applications. FOCUS [3] and LVF [7] examine the feature subsets: whether they can divide objects into groups of (almost) pure classes.

3 Preliminaries

In this section, we introduce our notation (Table 1) and provide a formal problem definition.

Given a data matrix $D = S \times A$, let $S = \{s_1, s_2, ..., s_n\}$ represent the set of samples and $A = \{a_1, a_2, ...a_m\}$ represent the set of features. Each sample $s_i \in S$, has an associated target variable value that is denoted by $f(s_i)$, $f(s_i) \in R$. Given a subset of samples $S' \subseteq S$, the values of their target variable are denoted by $F(S')$, where $F(S') = \{f(s_i)|s_i \in S'\}$.

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

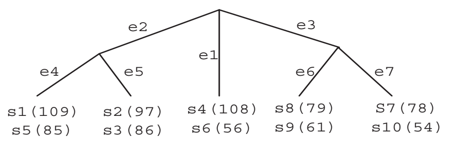


Figure 1. An example tree of 10 samples

Trees are used to represent similarities between samples. In a tree, all samples are at leaf nodes. It is possible that multiple samples are at the same leaf. For a tree T, let $E(T) = \{e_1, e_2, ..., e_p\}$ denote the set of edges in T. The removal of each edge partitions the samples into two subsets. Without loss of generality, we use $S^{(0)}(e_i)$ to denote the larger subset and $S^{(1)}(e_i)$ to denote the smaller one. $S^{(0)}(e_i) \cap S^{(1)}(e_i) = \emptyset$, $S^{(0)}(e_i) \cup S^{(1)}(e_i) = S$. Trees can be constructed on any feature subset using approaches such as hierarchical clustering [4] or phylogeny inference [1]. Details of tree construction are discussed in Section 4.

An example tree of 10 samples is shown in Figure 1. Each sample and edge is uniquely labelled. The target variable values are in the parentheses. For example, edge e_2 partitions the samples into two subsets, $S^{(1)}(e_2) = \{s_1, s_5, s_2, s_3\}$ and $S^{(0)}(e_2) = \{s_4, s_6, s_8, s_9, s_7, s_{10}\}$.

Given a tree T with $|E(T)|$ edges, we can generate $2|E(T)|$ sample subsets by removing each edge separately. We denote this set of sample subsets by $S^{(E)}(T)$,

$$S^{(E)}(T) = \{S^{(0)}(e_i), S^{(1)}(e_i)|e_i \in E(T)\}.$$

A set of disjoint sample subsets form a *grouping*.

Definition 3.1 *A grouping of a sample subset $S' \subseteq S$ contains a set of non-overlapping subsets of S' whose union equals to S'. Let $G(S')$ represent a grouping of S',*

$$G(S') = \{S'_1, S'_2, ..., S'_k\}, S'_i \subset S', S'_i \cap S'_j = \emptyset, \bigcup_{i=1}^{k} S'_i = S'.$$

Definition 3.2 *Given a tree T_i, we say a grouping $G(S')$ follows T if and only if $\forall S'_i \in G(S')$, $S'_i \in S^{(E)}(T)$.*

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 Figure 1, while grouping $G(S') = \{\{s_1, s_2\}, \{s_8, s_4\}\}$ does not.

Definition 3.3 *Given a sample subset S', $G_1(S')$ is called the parent-grouping of $G_2(S')$ (and $G_2(S')$ called the child-grouping of $G_1(S')$) if and only if $\forall S'_i \in G_1(S')$*

$$\exists S'_j \in G_2(S'), s.t. S'_i = S'_j$$

or

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

Intuitively, a child-grouping represents a refined sample partition from its parent-grouping. For example, in Figure 1, grouping $\{\{s_1, s_5, s_2, s_3\}, \{s_4, s_6\}\}$ is a parent-grouping of $\{\{s_1, s_5\}, \{s_2, s_3\}, \{s_4, s_6\}\}$.

Table 1. Notation Summary

D	data matrix $D = S \times A$
S, s_i	the sample set, a sample
S',S'_i	a subset of samples
A, a_i	the feature set, a feature
$f(s_i)$	target variable value of sample s_i
$F(S')$	target variable values of the samples in S'
$G(S')$	a grouping of sample subset S'
T,T_i	a tree connecting the samples
$E(T)$	the edge set of T
$S^{(0)}(e_i)$, $S^{(1)}(e_i)$	two sample subsets separated by e_i
$S^{(E)}(T)$	all the sample subsets separated by edges in T

Association Computation

We use the one-way ANOVA test with permutations to measure the association between a grouping of samples and the target variable. To accelerate the execution, we re-derive the formula of ANOVA test to maximize the reuse of computations. Consider grouping $G(S') = \{S'_1, S'_2, ..., S'_k\}$, for each sample subsets S'_i, we calculate

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

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

Combining all sample subsets together, we have

$$MM = \frac{1}{|S'|}\sum_{i=1}^{k} SM(S'_i),\ MSE = \frac{1}{|S'|-k}\sum_{i=1}^{k} SSE_i \quad (3)$$

$$MSB = \frac{1}{k-1}(\sum_{i=1}^{k} SSB_i - |S'| \cdot MM^2) \quad (4)$$

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

$$\Gamma_0(G(S')) = \frac{MSB}{MSE} \quad (5)$$

A higher base score indicates a stronger association between the grouping and the target variable. For example, given the tree and the target variable values in Figure 1 and two groupings, $G(S'_1) = \{\{s_2, s_3\}, \{s_4, s_6\}, \{s_8, s_9\}\}$ and $G(S'_2) = \{\{s_2, s_3\}, \{s_8, s_9\}\}$. Their Γ_0 scores are $\Gamma_0(G(S'_1)) = 0.44$ and $\Gamma_0(G(S'_2)) = 4.16$. Thus, grouping $G(S'_2)$ has a stronger association with the target variable than grouping $G(S'_1)$.

To overcome the multiple test errors, we apply a permutation test on the grouping $G(S')$ to calculate a significance score. To generate a random permutation, the target variable values in $F(S')$ are randomly re-assigned to samples in S'. Then we calculate a Γ-score using the new assignment of target variable values following Equations 1 to 5.

Assume that we conduct *nPerm* random permutations in total, for each permutation, we get score $\Gamma_j (j = 1...nPerm)$. Among the $nPerm$ Γ-scores, let p be the number of scores which are greater than or equal to the base score $\Gamma_0(G(S'))$, $p = |\{\Gamma_j | \Gamma_j \geq \Gamma_0(G(S')), j \in 1...nPerm\}|$. Then the significant score (P score) is

$$P(G(S')) = log_{10}(\frac{nPerm}{p}) \quad (6)$$

A high P score indicates a significant association between grouping $G(S')$ and the target variable.

Definition 3.4 ***The association between a tree and the target variable****: For a tree T, the highest P score achieved by any grouping based on the tree is considered the P score of the tree which represents the association between the tree and the target variable,*

$$P(T) = max\{P(G_j(S'))|S' \subseteq S\}, G_j(S')\ follows\ T. \quad (7)$$

Problem Definition: Given a set of samples, their quantitative target variable values and a set of trees representing similarities between samples, we want the top K trees with the highest P scores.

4 TreeQA Algorithm

In this section, we present our TreeQA algorithm. TreeQA takes as input a quantitative target variable and a set of trees defined over a common set of samples. It explores the search space of all possible groupings and returns the K best trees with highest P scores according to Equation 7.

4.1 Pre-processing: Tree Construction

The tree construction approach and the selection of feature subsets depend on the data and application. Due to space limitation, we give two brief examples of the tree construction pre-process in different applications.

Example 1: NBA dataset. In the NBA statistic dataset, each row represents a NBA player and each column represents a feature such as 'number of rebounds'. We select all feature subsets of size up to r (r may be 3,4,..) and construct a tree hierarchy for each feature subset using a **hierarchical clustering algorithm** [4]. Each player is represented by a leaf node. Euclidian distance is used to measure the similarities between players.

Example 2: Mouse dataset. The mouse dataset contains a set of mouse samples whose features are the genetic variations on a set of positions in their DNA sequences. We want to find a set of positions in the DNA whose genetic variation can explain the target variables. Trees are constructed by a **phylogeny inference algorithm** [1] in genomic regions exhibiting no evidence of historical recombination, as indicated by the 4-gamete test [5]. All samples are at the leaf nodes. Each internal node represents a hypothetical common ancestor of a subset of samples. The similarities between samples are measured by their relationships in the evolution.

The tree construction in Example 1 is general and can be applied to any numeric data matrix and target variables.

4.2 TreeQA

TreeQA explores all groupings following a given tree. According to Definition 3.2, a grouping $G(S')$ follows T if and only if $G(S')$ can be created using non-overlapping subsets in $S^{(E)}(T)$. By imposing an order on sample subsets in $S^{(E)}(T)$, TreeQA is able to enumerate and evaluate all combinations of non-overlapping subsets systematically. The pseudocode code of TreeQA is in Figure 2.

Main Routine
Input:

- Sample set S.
- A list of trees connecting the samples, $\{T_i\}$.
- $F(S)$, quantitative target variable values.
- *nPerm*, number of permutations.
- K, number of (top-K) trees to report.

Output: Top-K trees with $P(T_i)$.
Method:

```
1.  for each $T_i$ in the set of trees
2.      $SE = \{se_i | se_i \in S^{(E)}(T_i)\}$
3.      $P(T_i) = 0$.
4.      for j=1 to $|SE|$
5.          add $se_j$ to curlist.
6.          add $se_l$ to remlist if $l > j, se_j \cap se_l = \emptyset$
7.          Enumerate(curlist,remlist,nPerm,$P(T_i)$)
8.  report the $K$ trees having the highest $P(T_i)$ scores.
```

Subroutine: Enumerate(*curlist*,*remlist*,*nPerm*,$P(T_i)$)
Method:

```
1.  for each remaining subset $se_l$ in remlist
2.      create grouping $G(S')$=curlist$\cup\{se_l\}$.
            $S' = \bigcup se_q, se_q \in curlist \cup \{se_l\}$
3.      calculate base score $\Gamma_0(G(S'))$.
4.      $p = 0$.
5.      for j=1 to nPerm
6.          random permutation and re-calculate $\Gamma$.
7.          if($\Gamma \geq \Gamma_0(G(S'))$), $p = p + 1$.
8.      $p = log_{10}(\frac{nPerm}{p})$.
9.      $P(T_i) = max(p, P(T_i))$.
10.     remlist=remlist-$\{se_l\}$
11.     curlist'=curlist$\cup\{se_l\}$. remlist'=remlist.
12.     remove subsets in remlist' that overlap with $se_l$.
13.     Enumerate(curlist',remlist',nPerm,$P(T_i)$)
```

Figure 2. The TreeQA Algorithm

TreeQA enumerates all groupings via a depth-first recursive procedure. Two lists, $curlist$ and $remlist$, are used to hold the current grouping and the list of other sample subsets that do not overlap with current grouping, respectively. TreeQA extends the current grouping by adding each sample subset in $remlist$ to $curlist$ one at a time. The association of each new grouping to the target variable via a permutation test is computed. The P score of the tree is updated accordingly. The enumeration continues recursively for each newly extended grouping.

4.2.1 Reuse of Statistical Calculations

In Section 3, we re-derive the formula of ANOVA test to maximize the reuse of intermediate computations. For any sample subset S', $SQ(S')$ and $SM(S')$ calculated using the original target variable values (with no permutation) may be reused in any groupings (in any trees) that contain S'.

We employ a global prefix tree structure $Tree_{subset}$ to keep track of all sample subsets in any groupings examined thus far. It is shared by all trees. When a sample subset se_l is examined in **Enumerate**(), TreeQA first searches in $Tree_{subset}$ to see if it was examined before. If so, the corresponding SQ and SM values are retrieved and reused in the calculation. If it is the first time se_l is examined, SQ and SM are computed and a new entry is inserted in $Tree_{subset}$.

4.2.2 Effective Permutation

Given a grouping $G(S')$, a permutation test is conducted in the following two steps: (1) Randomly re-assigning the target variable values in $F(S')$ to samples in S'; (2) Calculating the corresponding Γ score using Equation 5. Both steps of the permutation test take $O(nPerm \cdot |S'|)$ time. Considering the large number of groupings in total, TreeQA uses a global prefix tree structure $Tree_{grouping}$ to organize groupings examined thus far to exploit maximal reusability of partial computation shared by permutation tests. In particular, we investigate two optimizations: *inTree* and *amgTree*.

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 Figure 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 Γ_0 scores are: $\Gamma_0(G_1(S')) = 9.79$ and $\Gamma_0(G_2(S')) = 4.32$. 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 Γ scores for both groupings: $\Gamma(G_1(S')) = 0.12$ and $\Gamma(G_2(S')) = 0.7$. 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 the 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 Γ 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.

amgTree: Effective permutation among trees

In TreeQA, the trees are processed sequentially. Among these trees, the same grouping occurs repeatedly. 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. For subsequent occurrences, we only need to retrieve its P score from $Tree_{grouping}$.

Based on our experiments on real data, using *amgTree* alone can reduce $40\% - 50\%$ of the execution time. When using *inTree* and *amgTree* together, we can reduce $70\% - 80\%$ of the execution time.

5 Experiments

We compared TreeQA with other quantitative association analysis methods on real datasets and tested the scalability of TreeQA. All experiments were conducted on a PC with CPU P4 3GHz, 1G RAM and 80G HDD.

5.1 Genome-wide association mapping

The goal of genome-wide association (GWA) mapping is to identify genes or regions in the genome that have significant statistical associations to a given phenotype.

Samples: We use a mouse dataset[2] from the Jackson Laboratory, consisting of 51 samples (mouse strains) and 7 million genetic variations among these 51 samples in the form of single nucleotide polymorphisms (SNP) in their DNA sequences. Each SNP is treated as a feature.

Trees: Evolutionary (phylogeny) trees can be constructed based on sequence similarities between samples. An example is shown in Figure 3(c). A total of 600K evolutionary trees were derived. Details of tree construction have been discussed in Section 4.

Target variables: We used high density lipoprotein cholesterol (HDL-C) levels in blood as the test phenotype (target variable). These phenotype data can be downloaded from the Mouse Phenome Database[3]. Several sets of HDL cholesterol levels are included, each of which was collected under different conditions, and was treated as a separate target variable. Three sets of HDL-C levels are used in our experiments in this section, which are listed in Table 2.

Table 2. Three Target Variables

target variable	set of HDL cholesterol levels
1	Paigen1, % of total plasma cholesterol for male
2	Paigen4, amount of HDL cholesterol [mg/dL] for male
3	Paigen2, change of cholesterol after diet for male

Algorithms: We compare TreeQA with two previous algorithms designed specifically for genomic quantitative association analysis.

- SMA: We implemented the Single Marker Association (SMA) algorithm as the representative for single feature association analysis algorithms [2]. SMA tests the association between each single feature (i.e., SNP) and a target variable using the F-statistics, validated by random permutation tests.

- HAM: We implemented the Haplotype Association Mapping (HAM) [9] algorithm as the representative for multi-feature quantitative association analysis algorithms. It slides a 3-SNP window through the genome and performs ANOVA on sample groups having the same 3-SNP sequence and random permutation tests are used to calculate each association's significance.

QHPM [11] is not used in our comparison because of its inability to handle a large number of features. Blossoc[8] and TreeDT [12] are not used because they require categorical target variables.

[2]http://cgd.jax.org/ImputedSNPData/imputedSNPs.htm

[3]http://phenome.jax.org/pub-cgi/phenome/mpdcgi?rtn=meas/catlister/req=Cblood+lipids

Results

Some candidate genes (*ppara*, *srb1*, *tnfrsf6*) that may play a role in regulating HDL-C levels are reported in [13]. We compared the effectiveness of TreeQA with other algorithms based on how accurately they can localize these genes. We ran the three algorithms, TreeQA, SMA and HAM on the mouse data with 10000 random permutations and compare the top 100 regions/locations reported by these algorithms.

Only TreeQA reports the locations of all three genes, while HAM and SMA locate only one. Due to space limitation, we only show the results on genes *ppara* and *srb1*.

Gene: *ppara*. The first candidate gene *ppara* spans from 85563542(base) to 85630585(base) on chromosome 15. We use target variable 1 in Table 2. Figure 3(a) plots the top 100 findings of each algorithm. The vertical line indicates the position of *ppara*. Only TreeQA reports significant associations at the gene's location with the highest P score 4. Both SMA and HAM miss it.

The tree structure having the strongest association is shown in Figure 3(c). The target variable value of each sample is given in the parentheses. Most samples in the left subtree have high values while most samples in the right subtree have low values. The samples in the middle have mixed values. SMA and HAM fail to identify it because they only examine sample groupings that can be generated from single SNPs or 3-SNP combinations. Some samples in middle branch of the tree may be outliers and hence are excluded by TreeQA. Both SMA and HAM do not allow for this option.

Gene: *srb1*. The second candidate gene *srb1* spans from 125566069(base) to 125630053(base) on chromosome 5. We use target variable 2 in Table 2. The top 100 locations reported by the three algorithms are plotted in Figure 3(b). TreeQA is able to find the gene. The nearest significant location found by HAM is 766k (bases) away from *srb1*. The locations reported by SMA are even further away.

5.2 Association analysis in the NBA data

To demonstrate the versatility of TreeQA, we applied it to an NBA dataset. Both HAM and SMA were designed specifically for genome association studies, and cannot easily handle other datatypes. Thus, we do not furnish a comparison.

Samples: The data was downloaded from the ESPN web page[4]. The dataset contains 28 statistics of 200 NBA players, such as "number of blocks", "number of points", etc..

Trees: An hierarchical-clustering algorithm [4] is used to construct a tree in every 2 and 3-feature subspace. Details of tree construction have been discussed in Section 4.

Target variables: A subset of features are selected to be the target variables, such as "minutes played in each game" and "total number of rebounds".

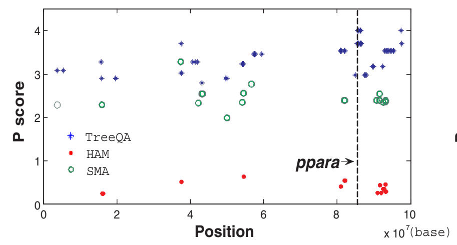
Results

TreeQA finds many obvious associations such as

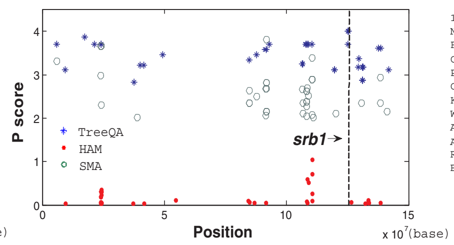
- {"number of assists", "number of turnovers"} $\rightarrow$"assistant/turnover".

as well as some non-trivial ones including

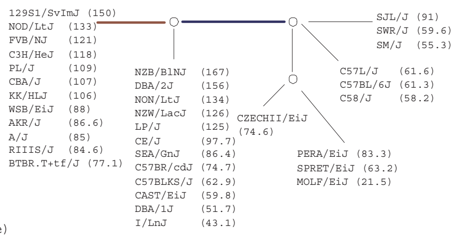
[4]http://sports.espn.go.com/nba/teams/stats?team=Bos year=2007



(a) Top 100 locations found by TreeQA, HAM and SMA on chromosome 15



(b) Top 100 locations found by TreeQA, HAM and SMA on chromosome 5



(c) The evolutionary tree at the location of gene ppara

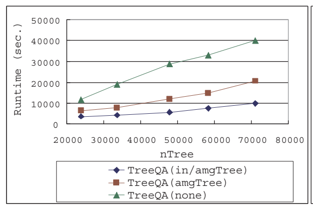
Figure 3. Experiments of genome-wide association mapping

- {"number of defensive rebound", "fields goals made"} →"minutes played in each game". Indicating that players who are better at defensive rebounding and making field goals tend to get more playing time.
- {"number of steals", "number of turnovers"} →"number of assists". Suggesting that players who have a high number of assists also tend to have high number of steals as well as turnovers.

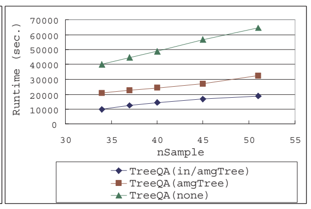
5.3 Scalability Analysis

We test the scalability of TreeQA with respect to two parameters: (1) *nTree*, number of trees (default value, 7.1×10^4) and (2) *nSample*, number of samples (default value, 34). When we vary one parameter, the other uses its default value. We use 10000 permutations in each permutation test.

Figure 4 plots the runtime performance of: ***TreeQA(none)***, TreeQA without efficient permutation; ***TreeQA(*amgTree*)***, with *amgTree* only; and ***TreeQA(*in/amgTree*)***, with *inTree* and *amgTree*.



(a)



(b)

Figure 4. Runtime of TreeQA(s)

The runtime of TreeQA increases linearly as *nPerm*, *nTree* and *nSample* increases. In fact, the runtime of TreeQA is also linear with respect to the number of permutations in the permutation tests. Using the *amgTree* method alone reduces the runtime of TreeQA(none) by approximately $40-50\%$, while using *amgTree* and *inTree* together can reduce the runtime of TreeQA(none) by $70-80\%$. As the number of trees grows, *amgTree* becomes more effective at reducing the computational time, i.e., from 44% reduction for 2.4×10^4 trees to 48% reduction for 7.1×10^4 trees.

6 Conclusion

Association analysis aims to find the dependency of a target variable to subsets of measurements from a set of samples. Most previous algorithms take a two-stage approach in which the groupings of samples are determined prior to the association assessment. In this case, the strength of association may be influenced by arbitrary choices in clustering methods and/or cut-off parameters. In this paper, we propose a tree-based method for quantitative association analysis which does not require a priori sample groupings. Instead, trees are used as the representation of similarities between samples in different feature subspaces or under different metrics. We propose an algorithm, TreeQA, which explores the space of all possible groupings in a carefully designed order so that intermediate computation can be maximally reused. The K trees with strongest associations with the target variable are returned. Our experimental results show that TreeQA is able to handle large-scale association analysis very efficiently and effectively.

References

[1] R. Agarwala, D. Fernandez-Baca, and G. Slutzki. Fast algorithms for inferring evolutionary trees. *Journal of Computational Biology*, 2(3):397–408, 1995.
[2] 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.
[3] H. Almuallim and T. G. Dietterich. Learning with many irrelevant features. *Proceedings of the 9th National Conference on Artificial Intelligence,*, 1991.
[4] E. B. Fowlkes and C. L. Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383):553–569, Sep. 1983.
[5] 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.
[6] J. Li and T. Jiang. Haplotype-based linkage disequilibrium mapping via direct data mining. *Bioinformatics*, 21(24):4384–4393, 2005.
[7] H. Liu and R. Setiono. A probabilistic approach to feature selection : A filter solution. *Proceedings of the 13th International Conference on Machine Learning*, 1996.
[8] T. Mailund, S. Besenbacher, and M. H. Schierup. Whole genome association mapping by incompatibilities and local perfect phylogenies. *BMC Bioinformatics*, 7:454, 2006.
[9] P. McClurg and et al. Comparative analysis of haplotype association mapping algorithms. *BMC Bioinformatics*, 2006.
[10] R. G. Miller. *Simultaneous Statistical Inference (2nd Ed)*. Springer Verlag New York, 1981.
[11] P. Onkamo and et al. Association analysis for quantitative traits by data mining: Qhpm. *Ann. Hum. Genet.*, 2002.
[12] P. Sevon, H. Toivonen, and V. Ollikainen. Treedt: Tree pattern mining for gene mapping. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3, 2006.
[13] X. Wang and B. Paigen. Quantitative trait loci and candidate genes regulating hdl cholesterol. *Arteriosclerosis, Thrombosis, and Vascular Biology*, 22:1390, 2002.