Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Theoretical Biology 228 (2004) 477–496
Journal of Theoretical Biology
www.elsevier.com/locate/jtbi

Tumor classification using phylogenetic methods on expression data

Richard Desper[a,*], Javed Khan[b], Alejandro A. Schäffer[a]

[a] *Computational Biology Branch, National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Department of Health and Human Services, Bldg. 38A, Room 8N805, 8600 Rockville Pike, Bethesda, MD 20894, USA*
[b] *Pediatric Oncology Branch, National Cancer Institute, National Institutes of Health, Department of Health and Human Services, Gaithersburg, MD, USA*

Received 19 March 2003; received in revised form 3 February 2004; accepted 20 February 2004

Abstract

Tumor classification is a well-studied problem in the field of bioinformatics. Developments in the field of DNA chip design have now made it possible to measure the expression levels of thousands of genes in sample tissue from healthy cell lines or tumors. A number of studies have examined the problems of tumor classification: *class discovery,* the problem of defining a number of classes of tumors using the data from a DNA chip, and *class prediction*, the problem of accurately classifying an unknown tumor, given expression data from the unknown tumor and from a learning set. The current work has applied phylogenetic methods to both problems. To solve the class discovery problem, we impose a metric on a set of tumors as a function of their gene expression levels, and impose a tree structure on this metric, using standard tree fitting methods borrowed from the field of phylogenetics. Phylogenetic methods provide a simple way of imposing a clear hierarchical relationship on the data, with branch lengths in the classification tree representing the degree of separation witnessed. We tested our method for class discovery on two data sets: a data set of 87 tissues, comprised mostly of small, round, blue-cell tumors (SRBCTs), and a data set of 22 breast tumors. We fit the 87 samples of the first set to a classification tree, which neatly separated into four major clusters corresponding exactly to the four groups of tumors, namely neuroblastomas, rhabdomyosarcomas, Burkitt's lymphomas, and the Ewing's family of tumors. The classification tree built using the breast cancer data separated tumors with BRCA1 mutations from those with BRCA2 mutations, with sporadic tumors separated from both groups and from each other. We also demonstrate the flexibility of the class discovery method with regard to standard resampling methodology such as jackknifing and noise perturbation. To solve the class prediction problem, we built a classification tree on the learning set, and then sought the optimal placement of each test sample within the classification tree. We tested this method on the SRBCT data set, and classified each tumor successfully.
© 2004 Elsevier Ltd. All rights reserved.

Keywords: Tumor classification; Gene expression data; Microarrays; Phylogenetic trees; Small round blue cell tumors; Breast cancer

1. Introduction

The past few years have seen the development of large expression data sets from oligonucleotide arrays and cDNA microarrays. These new technologies have yielded a wealth of data, the analysis of which offers the promise of new insights into the functions of genes as well as the development of new molecular taxonomies of cancers. The identification of genes with similar expression patterns may lead to a better understanding of the regulatory networks underlying both healthy and cancerous tissues. Such developments would have obvious implications for cancer diagnosis, prognosis, and guiding therapy, as well as for the identification of new targets for treatment.

Class discovery and *class prediction* are two of the major bioinformatics problems in the field of gene expression analysis of tumor samples. Expression data are collected, usually with a classifier variable that assigns each sample to one of a number of classes, for example the diagnosis or type of cancer. Class discovery is the process of recognizing or defining classes based solely on the expression data in the absence of classifier variables. For example, Alizadeh et al. (2000) discovered two previously unrecognized types of diffuse large B-cell lymphoma using only gene expression data. The related problem of class prediction presumes that one is given a fixed set of classes, perhaps explored with a training set, and that one seeks to assign unclassified tumors to this

*Corresponding author. Fax: +1-301-480-2288.
E-mail address: desper@ncbi.nlm.nih.gov (R. Desper).

0022-5193/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jtbi.2004.02.021

set of classes, where the classification is based on expression data. Both problems rely heavily upon the ability to discern informative genes and the removal of noise in the data sampling process.

A number of approaches have been developed to tackle the class discovery problem. Foremost among these is the method of *hierarchical clustering* (Eisen et al., 1998), which imposes a hierarchical structure on the set of genes (or, alternatively, tissue samples) using a distance function on the objects undergoing classification. Depending on the distance function and cluster diameter used, one could infer a variety of different clusterings from the hierarchy.

Another approach to clustering is the k-means approach (Herwig et al., 1999), which seeks an optimal clustering of the data around k centroids. A related approach known as self-organizing maps (SOMs), described by Tamayo et al. (1999) and Golub et al. (1999), assigns the gene expression data points to clusters chosen from among a small number of prescribed geometries, where each geometry has no more than k clusters. The former work clustered genes while the latter clustered tumors. Other approaches include linear discriminant analysis (e.g. von Heydebreck et al., 2001), the search for separating hyperplanes (e.g. Furey et al., 2000), and graph-theoretical methods such as the minimal spanning tree approach of Xu et al. (2002). A comparison of several methods is given by Dudoit et al. (2002).

The current work has two applications, one for the class prediction problem and one for the class discovery problem. The common thread to both applications is the construction of classification trees—trees whose leaves correspond to the items being classified. When some of the data are classified, we use the tree structure implied by classified nodes to classify unknown samples. When none of the input data are classified, we use features of the tree structure to define classes.

In the current work, we consider gene expression data measured by cDNA microarrays. We present a method of imposing a classification tree on a set of tumors, using the expression data. We present the program METRics on EXpression Data (METREX), which uses the expression data to calculate a variety of metrics on the tumors. We then use standard phylogenetic methods to fit trees to these metrics to define weighted classification trees.

A classification tree is a more detailed structure than a clustering: it contains not merely clusters and subclusters, but branch lengths in the classification tree can indicate the degree of separation between clusters. Longer branches tend to imply greater separation, and can be understood intuitively to represent more meaningful splits in the data. Our work is closely related to the minimum spanning tree method of Xu et al. (2002). Waddell and Kishino (2000) proposed using phylogenetic methods for clustering genes using microarray data. The motivation behind the phylogenetic methodology is manifold: we seek to avoid the geometric assumptions behind the separating hyperplane method, and the arbitrary selection of a number of clusters behind k-means and SOM approaches. Also, a hierarchical clustering lacks information contained in a classification tree, namely the degree to which varying clusters may be separated from each other.

As a proof of method test, we considered a data set on small, round blue-cell tumors (SRBCTs) presented in the work of Khan et al. (2001). This study included data from a 6567-element array, tested over a training set of 63 samples and 25 test samples. One of the test samples was from the same tumor as a training sample, so we deleted it from our consideration. Of the 6567 genes in the data array considered, 96 were selected by Khan et al. by a method using artificial neural networks to best distinguish the four groups in question. Thus, our consideration does not represent "class discovery" in a pure sense. Information about the tumor classes was used during the gene selection process. However, this information was not used during the process of building the classification tree from the set of selected genes. Thus, we have essentially chosen a relatively easy problem for our proof of method analysis. We feel this is acceptable, as we used the same data set for all the class discovery methods we examined. We compare the classification trees generated by this data set with the clusterings produced by other leading software packages.

To validate our methods, we applied them to a data set of 22 breast cancer samples with 3 classes: BRCA1-mutated, BRCA2-mutated, and sporadic (Hedenfalk et al., 2001).

We also present a method for using classification trees to solve the *class prediction* problem. We do this by building a classification tree for the learning set, and then finding the optimal insertion point for each tumor in the learning set. Presuming the initial classification is meaningful with regard to pre-defined group labels, the placement in the tree can suggest a labeling for any test data.

2. Expression data preprocessing

2.1. Notation

We shall use the following notational conventions. Most mathematical symbols shall be referred to using *italics*. Following a standard convention, we shall use boldface when referring to matrices or vectors, using capital letters for matrices and lowercase letters for vectors. The convention will be to use the same letter for the entries of a matrix as for the matrix itself, except that

the entries will be lowercase while the matrix will be boldfaced. For example, the matrix **B** will have entries $((b_{ij}))$. Also, we shall make reference to metrics and distance matrices. A distance matrix may have entries that correspond to a given metric. In such a case, the metric will be referred to, as a function would be, in italics, while the matrix will be boldfaced with the same letter. For example, we shall use **D** with entries $((d_{ij}))$ to represent the metric D, where $d_{ij} = D(i,j)$. Also, it is traditional to use the Greek letter Δ (with entries $((\delta_{ij}))$) to represent the input to a tree-fitting algorithm.

2.2. Data collection and gene selection

Let us assume that data is of the following form: for each of m tissue samples in the set $X = \{x_1, x_2, \ldots, x_m\}$, we are given the expression level for each of n genes in the set $G = \{g_1, g_2, \ldots, g_n\}$. This yields a $m \times n$ matrix $A = ((a_{ij}))$, where a_{ij} is the expression level in the sample x_i of the gene g_j. There may also, in some cases, be a discrete classifier variable, y_i, for $1 \leqslant i \leqslant m$, assigning each sample x_i to one of a small number of sets or *clusters*.

Traditionally, expression data studies have focused on the following two problems (Dudoit et al., 2002; Radmacher et al., 2002):

- The *tumor classification* problem: given **A**, and an integer $k > 0$, define a meaningful classification function $y : X \to \{1, 2, \ldots, k\}$. This area of research is known as *class identification* or *unsupervised learning*.
- The *class prediction* problem: also known as *supervised learning*, as the user is given not only the expression data **A**, but also a classification function y on a subset of the data, $X_0 \subset X$. The problem is to extend y to all of X in some meaningful manner.

Where previous studies have used a fixed k, or tried to maximize the fit over various values of k, phylogenetic software used on a METREX distance matrix produces a weighted tree. The edge weights could be used to classify the data for any value of k. In our discussions of the data sets examined, we will suppress discussion of the classification function y, instead making an explicit classification either by tissue type or by genetic composition.

2.3. Normalizing the input matrix

Our approach is aimed at fitting a tree metric to the expression data. But the input matrix **A** contains expression levels for a wide variety of genes. Since different genes activate at different expression levels, there is little sense in comparing the actual values of a_{ij}. Were we to do tree fitting on the raw data, the tree topology would be determined by those variables that have the greatest values. To avoid this problem, we normalize the input matrix by a linear transformation, to weigh each gene equally.

Consider the gene g_j, whose values are represented in the Jth column of **A**. Let μ_j denote the mean expression levels for g_j, and let σ_j denote the corresponding standard deviation. We define the normalized matrix **B** with entries $((b_{ij}))$ by

$$b_{ij} = \frac{a_{ij} - \mu_j}{\sigma_j}.$$

Normalizing the matrix allows us to compare the expression levels across columns in a meaningful manner that weighs each column equally. Also, to reduce any possible effects of outlier data, we trimmed any input data more than four standard deviations from the mean; i.e. we imposed a constraint that $|b_{ij}| \leqslant 4$ for all i and j. This constraint was imposed to diminish the deleterious effect outliers can have on distance-based tree reconstruction algorithms (Huson et al., 1999). This constraint was not a factor at all in the analysis of the two main data sets below, as all of the data points were less than four standard deviations from the mean, but it did play a very minor role in the resampling process.

3. classification methodology

In this section, we will define trees, metrics and tree metrics, and provide a simple method for imposing a meaningful tree structure on classification data.

3.1. Metrics and tree fitting

Formally, a *graph* is a pair $G=(V,E)$, where V is a finite set of objects and E is a set of pairs of objects from V. A *cycle* in a graph is a sequence $c = (v_0, e_1, v_1, e_2, \ldots, e_k, v_k)$, where $v_i \neq v_j$ for $1 \leqslant i < j \leqslant k$, but $v_0 = v_k$, and $e_i = (v_{i-1}, v_i) \in E$ for all i. Similarly, a *path* in a graph is a sequence $p = (v_0, e_1, v_1, e_2, \cdots, e_k, v_k)$ where $v_i \neq v_j$ for $0 \leqslant i < j \leqslant k$ and $e_i = (v_{i-1}, v_i) \in E$ for all i, in which case we say v_0 and v_k are connected by p. A graph is *connected* if, for all pairs $x, y \in V$, there is a path p_{xy} in G connecting x and y. A connected graph containing no cycles is called a *tree*.

Let L be a set of objects, and $\mathbb{R}$ the set of real numbers. Without loss of generality, $L = \{1, 2, \ldots n\}$. A metric on L is a function $D : L \times L \to \mathbb{R}$ satisfying the three properties:

1. $D(i,j) \geqslant 0$ for all $i, j \in L$, with $D(i,j) = 0$ if and only if $i = j$.
2. $D(i,j) = D(j,i)$ for all $x, y \in L$.
3. For all $i, j, k \in L$, $D(i,k) \leqslant D(i,j) + D(j,k)$

We can express D as a *distance matrix* $\mathbf{D}$, with entries $((d_{ij}))$, where $d_{ij} = D(i,j)$.

A tree metric is a specific kind of metric. Let T be a tree, $T = (V, E)$, with $L \subset V$, L the set of leaves of T. Let l be a function: $l : E \to \mathbb{R}^+$. For any pair of leaves $i, j \in L$, define $\mathscr{P}_{ij}$ to be the unique path in T from x to y. Let D^{T} be defined by

$$D^T(i,j) = \sum_{e \in \mathscr{P}_{ij}} l(e).$$

Suppose D is a metric on L. The tree fitting problem is: given D, find a tree T such that D^{T} is a good approximation for D. Tree fitting is one of a variety of methods that taxonomists use to construct phylogenetic trees to estimate evolutionary history. Leading tree fitting algorithms include the Neighbor Joining algorithm of Saitou and Nei (1987), the least-squares approach of Fitch and Margoliash (1967), and others. The advantages of using tree fitting include the ability to use well-refined, pre-existing software packages, whether commercial (PAUP) or in the public domain (PHYLIP), and a supporting body of literature which can guide us to which problems are computationally feasible and which are not.

Traditionally, tree fitting has been used in a setting where there is some reason to suppose that the input metric D can be well approximated by a tree metric. Most of the methods used in this paper are commonly used in phylogeny studies, and in classification problems in other settings. To our knowledge, the current work represents a novel step of using tree fitting for classification purposes on the metrics commonly used in expression data analysis.

3.2. Metrics on expression data

In the field of expression data analysis, it has been typical to define a metric on a set of genes, and to cluster the genes based on that metric. In the current work, we consider a variety of different metrics for our test data set. We use the Euclidean distance between expression profiles. Suppose we have m tumors sampled at each of n genes, and that $\mathbf{B}$ is the $m \times n$ normalized expression matrix with entries $((b_{ij}))$. We calculated each of the four metrics defined below on all pairs of row vectors of the matrix $\mathbf{B}$.

Suppose v and w are n-dimensional vectors (e.g. $v = (v_1, v_2, \ldots, v_n)$). For $p \geqslant 1$, the Minkowski p distance (also known as the L_p norm) between v and w is defined as

$$d_p(\mathbf{v}, \mathbf{w}) = \|\mathbf{v} - \mathbf{w}\|_p = \sqrt[p]{\sum_{i=1}^{n} |v_i - w_i|^p}. \tag{3.1}$$

For $p = 2$, this is the traditional Euclidean distance (or metric), while for $p = 1$ this is sometimes referred to as the "taxicab" metric. Similarly, suppose $\mathbf{M}$ and $\mathbf{N}$ are two m by n dimensional matrices. We define

$$d_p(\mathbf{M}, \mathbf{N}) = \|\mathbf{M} - \mathbf{N}\|_p = \sqrt[p]{\sum_{i=1}^{m} \sum_{j=1}^{n} |m_{ij} - n_{ij}|^p}. \tag{3.2}$$

This Euclidean distance is closely related to the "mean-square distance" of Chen et al. (1999):

$$d_{ms}(v, w) = \frac{1}{n} \sum_{i=1}^{n} (v_i - w_i)^2 = 2(1 - \sum_{i=1}^{n} v_i w_i). \tag{3.3}$$

There is zero mean-square distance between two variables that exhibit a positive linear relationship between the two variables (i.e. $\mathbf{v} = a\,\mathbf{w} + b$, with $a > 0$), and a distance of 4 between two variables that are negatively related to each other (i.e. $v = a\,w + b$, with $a < 0$), while a distance of 2 separates any pair of independent variables. The coefficient 2 in Eq. (3.3) is meaningless for classification purposes, and is dropped in many applications.

From an information theory perspective, negative correlation is just as important as positive correlation in establishing a relationship between two variables. To take advantage of this symmetry, we also use a version of mean-square distance where all correlations are treated the same:

$$d_{msa}(\mathbf{v}, \mathbf{w}) = 2(1 - \left|\sum_{i=1}^{n} v_i w_i\right|). \tag{3.4}$$

We refer to this quantity as the mean-square absolute distance.

Given an input data set consisting of an array $\mathbf{A}$ of expression data, the program METREX outputs a matrix of distances between the rows of the normalized matrix $\mathbf{B}$ corresponding to $\mathbf{A}$. METREX can calculate the $\mathbf{L}_1$ or $\mathbf{L}_2$ distances, as shown in Eq. (3.1), the mean-square distance (Eq. (3.3)), or the mean-square absolute distance (Eq. (3.4)).

Our approach to tree fitting, given an input matrix Δ with entries $((\delta_{ij}))$ representing one of the aforementioned metrics, is to use a number of different tree building programs to create trees whose corresponding tree metrics approximate Δ. We then compare the tree metrics to the original metric in order to select the tree that best fits the data. For the data we considered, we use the following programs to build the trees.

- We used the program `fitch` (Fitch and Margoliash, 1967) from the PHYLIP (Felsenstein, 1989) package to calculate the tree topology $T = T_W$ minimizing the weighted sum of squares (WLS).

  $$\sum_{i,j} \frac{\left(\delta_{ij} - d_{ij}^{\mathrm{T}}\right)^2}{\delta_{ij}^2}$$

  (Alternatively, one could use the heuristic search command *hsearch* of PAUP (Swofford, 1996) to

search across tree topologies, minimizing the WLS criterion.) We refer to this tree as the WLS tree. (To be more precise, neither `fitch` nor PAUP's *hsearch* guarantees a global minimum under the settings we used, but rather a local minimum. An exhaustive topology search is considered computationally prohibitive.)

- We used PAUP to calculate the neighbor-joining (Saitou and Nei, 1987) tree, T_{NJ}. (The PHYLIP program `neighbor` calculates the same tree.)
- We used the program `FastME`, the heuristic minimum evolution program of Desper and Gascuel (2002), to calculate three trees for each of the data sets. `FastME` can build an initial topology iteratively, or start topology searching from an input topology, using the minimum evolution criterion to select among topologies when each topology is assigned edge lengths according to the ordinary least-squares criterion. We used the former option to build one tree, and also built two trees resulting from topology searches using the fitch and NJ trees as starting topologies. We refer to the smallest of these three trees as the Minimum Evolution (ME) tree. (As with the WLS tree, the ME tree merely represents a local minimum under the search process, as exhaustive topology searching is computationally prohibitive. Simulations (Desper and Gascuel, 2002) have shown that the heuristic used is powerful in practice.)

Distance-based phylogenetic software such as PAUP and FASTME can build the trees very quickly, even when data sets grow to include hundreds of tumors. Given n tumors and m genes, the distance matrices can be computed with $O(n^2m)$ computations, and the NJ and ME trees can be calculated with $O(n^3)$ and $O(n^2 \log n)$ computations, respectively. Computational experiments in (Desper and Gascuel, 2002) showed that FASTME takes less than a minute for $n = 1000$.

3.3. Method of selecting tree from various choices from various algorithms

Given a variety of tree topologies, our next question was: which topology fits the data the best? Each of the various fitted trees induced a metric, which was then compared to the original metric. Each tree metric was compared to the original distance matrix by the L_1 and L_2 norms, defined by Eq. (3.2), and the L_∞ norm, defined as

$$\|\Delta - D^{\mathrm{T}}\|_\infty = \max_{i,j}\left|\delta_{ij} - d^{\mathrm{T}}_{ij}\right|.$$

Given a number of different output trees, we used an ad hoc method to decide which tree we preferred: given an input metric Δ, we sought the tree T whose corresponding tree metric D^{T} was closest to Δ according to the L_1, L_2 and L_∞ norms. If no tree metric was superior by all three norms, we selected one closest by a majority of the three norms, and, no tree was superior by at least two of the three norms, we looked at the two trees closest by the L_1 and L_2 norms, and used the L_∞ norm as a tiebreaker to decide between those two trees (discarding the tree that was optimal according to the L_∞ norm if it was not one of the two trees chosen already.) This ad hoc protocol was devised based on the belief that the L_∞ norm is a much poorer measure of convergence than either the L_1 or L_2 norms.

Each of these trees yields a metric that is only an approximation to the original metric. It is important to note how closely the metric of the preferred tree approximates the original metric, both in absolute terms, and relative to the other tree metrics. If two or more trees are approximately equally good in fitting the metric, then we should trust only the topological features shared by the trees.

4. Classification results

4.1. Classification of 87 samples: desired sub-units

We performed tree classification on the data set of Khan et al. (2001), consisting of expression levels of 96 genes from 87 tissue samples. This tissue samples include 63 samples from four types of small round blue-cell tumors (SRBCTs), including cell-line and tumor samples from neuroblastomas (NB), rhabdomyosarcomas (RMS), Burkitt lymphomas (BL), and the Ewing family of tumors (EWS). These 63 samples had been used by Khan et al. to calibrate and validate a number of artificial neural network models, which were then tested against a set of 24 blinded test samples including cell lines and tumors of the 4 aforementioned types, as well as 5 non-SRBCTs: 2 normal muscle tissues and 3 cell lines including an undifferentiated sarcoma, an osteosarcoma, and a prostate carcinoma. The 96 genes were selected by Khan et al. from a total data set of 6567 genes based on their ability to separate the four major classes. Whereas Khan et al. used the set of 63 samples as a training set, we blinded the entire data set from the start in order to test our ability to classify without any information about tissue types whatsoever.

Our primary goal was that each of the four large sets (NB, RMS, BL, and EWS) should be separated from the other three sets in our output tree. Also, we wished for a classification that would separate cell lines from tumor samples. We hoped that the five non-SRBCT samples would not be placed in the tree in a position that would suggest membership in one of the four major classes, and that the two skeletal muscle samples would group together as a subtree. Finally, we hoped that the classification tree might reveal structural elements

beyond those provided by the criteria for gene selection.

4.2. Comparison of ME, WLS, and NJ trees

4.2.1. Euclidean metric

We used METREX to define the metric Δ_E on the set of samples by setting $\Delta_E(x_i, x_j) = \|x_i - x_j\|_2$ for each pair of samples x_i and x_j. We then used `FastME`, `fitch`, and NJ on Δ_E to find the topologies of the trees T_E^{ME}, T_E^F, and T_E^{NJ}. Edge weights were assigned to the three topologies to minimize the OLS criterion. We used these trees to define the metrics Δ_E^{ME}, Δ_E^F, and Δ_E^{NJ}, respectively. Table 1 shows how each of these metrics compares to the original metric Δ_E. The fourth column, the residue, expresses the ratio of the L_1 distance between the input and output matrices to the size of the input matrix. This provides a meaningful way to compare the ability to fit trees to different metrics.

From Table 1, we see that the WLS tree fits the metric Δ_E slightly better than the ME tree does, and that both do a slightly better job at fitting than the NJ tree does. All of the output metrics vary by approximately 6% from the input metric. The WLS tree is shown in Fig. 1. In this tree, the edge lengths of the tree are proportional to the horizontal distances of the drawing, while the vertical distances in the drawing have no meaning other than to create space for the labels. The leftmost node of the tree is the *root* of the tree; we have rooted the tree in Fig. 1 such that one of the tree subtrees incident to the root is the subtree comprised of the two skeletal muscle samples. We chose this node as a root solely because the skeletal muscle subtree was present in all the trees we considered. All of the tree drawings in this paper were made using TREEVIEW (Page, 1996).

4.2.2. Taxicab metric

Given two vectors, u and v, the taxicab metric between them is defined to be

$$\|u - v\|_1 = \sum_i |u_i - v_i|.$$

We defined the taxicab metric Δ_T on the set of samples by setting $\Delta_T(x_i, x_j) = |x_i - x_j|_1$ for each pair of samples x_i and x_j. As with the Euclidean metric, we used `FastME`, `fitch`, and NJ on Δ_T to form the trees T_E^{ME}, T_E^F, and T_E^{NJ}. We used these trees to define the metrics Δ_E^{ME}, Δ_E^F, and Δ_E^{NJ}, respectively. Table 2 shows how each of these metrics compares to the original metric Δ_E. Again, the WLS tree is slightly better than the ME tree, and both are better than the NJ tree. The WLS tree is shown in Fig. 2.

Table 1
Fit of tree metrics to Euclidean metric

	L_1 distance	L_2 distance	L_∞ distance	Residue
ME tree	6026.5779	89.4440	4.9379	0.059301
WLS tree	5932.7527	87.8451	5.9943	0.058377
NJ tree	6079.7852	91.0193	5.7338	0.059824

Recall that the trees in Figs. 1 and 2 were built using blinded data, with the labels affixed to taxa only after the trees were created. The Ewing tumors are labeled EWS, the rhabdomyosarcoma RMS, the neuroblastoma NB, and the non-Hodgkin lymphomas either by Lymph or BL. Additionally, the data included a training set and 25 test samples, labeled with a TEST prefix in each tree.

Each of trees in Figs. 1 and 2 contains the desired subtrees: the RMS subtree, which itself splits into a tumor subtree and a cell line subtree; two analogous subtrees that together constitute a EWS subtree; a non-Hodgkin lymphoma subtree; and a neuroblastoma subtree. The lymphoma subtree was particularly set off from the other three trees; among the other three major subtrees, the neuroblastoma had the greatest separation from the other two. The test tissue samples were also placed in the corresponding subtrees, except for the five samples deemed to be outside the four main categories and two EWS samples. The two samples from skeletal muscle tissue were placed within the RMS subtree. The Test20-EWS-T sample, which not been confidently placed in (Khan et al., 2001), was placed close to the RMS subtree topologically, but it would be more accurate to say that it was by itself. The Test2-EWS-C sample, as well as the sarcoma, osteosarcoma, and prostate cancer samples all were placed roughly in the middle of the tree, with long pendant edges indicating small relationships relative to the rest of the tree.

4.2.3. Mean-square distance

We defined the metric Δ_{ms} to be the mean-square distance on the data, as defined in Section 3. Recall that this metric is defined by

$$\Delta_{ms}(x_i, x_j) = E[(X_i - X_j)^2] = 2(1 - E[X_i X_j]),$$

where X_i and X_j are the values in the corresponding rows of the normalized input matrix. As with the Euclidean and taxicab metrics, we compared the output trees from NJ, `fitch`, and `FastME`. Results are in Table 3. By all three measures, the WLS tree is the best of the three. We can see the tree in Fig. 3. We see the Test2-EWC-C sample nests inside the EWS tree when fitting to this metric, but the Test21-EWS-T sample now clusters with the previously problematic Test20-EWS-T sample in an isolated subtree.

4.2.4. Mean-square absolute distance

Recall that the mean-squared absolute distance is defined by

$$\Delta_{msa}(x_i, x_j) = 2(1 - \|E[X_i X_j]\|),$$

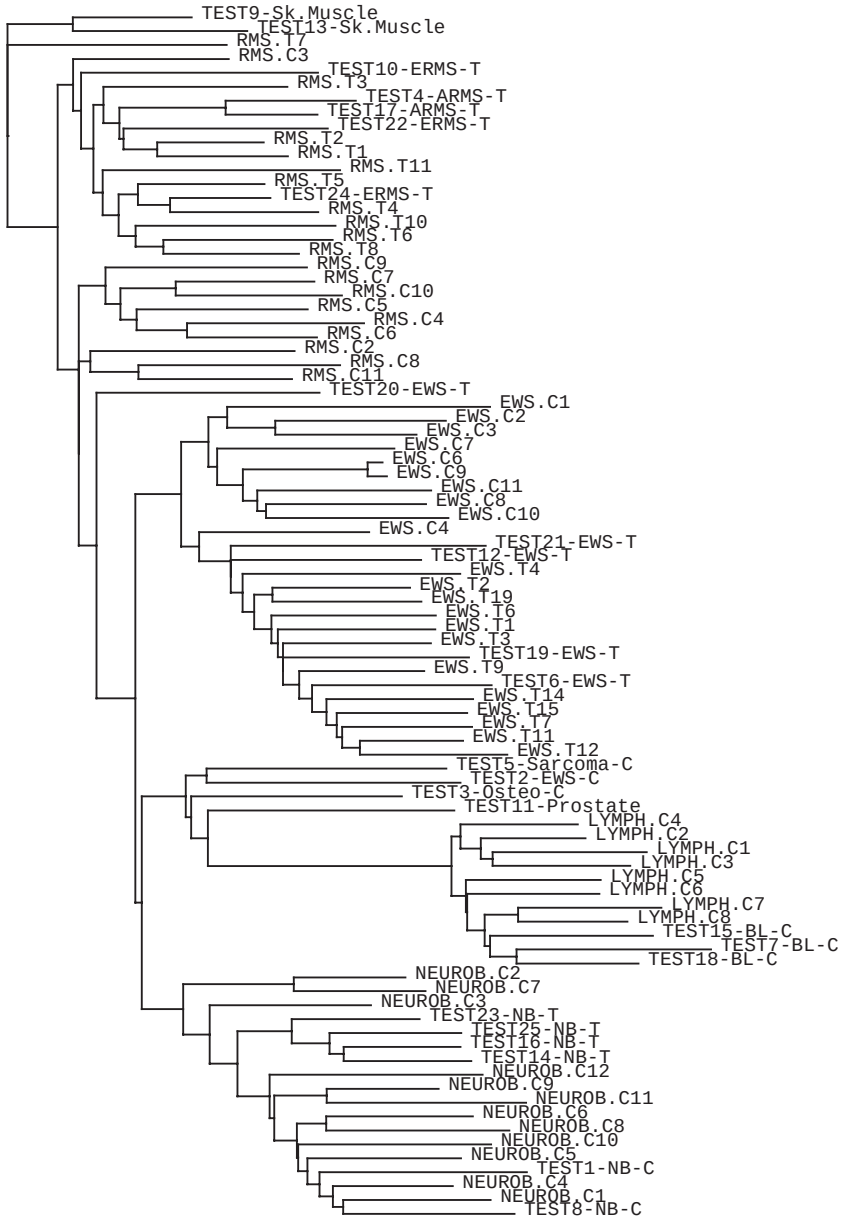

Fig. 1. Fitch/WLS tree for Euclidean metric.

Table 2
Fit of tree metrics to taxicab metric

	L_1 distance	L_2 distance	L_∞ distance	Residue
ME tree	57907.9441	851.4441	43.9973	0.071037
WLS tree	56844.5873	832.0697	50.6529	0.069733
NJ tree	62476.7315	930.1912	45.4722	0.076642

in contrast to the mean-squared distance. As with the other three metrics, we compared output trees from NJ, `fitch`, and `FastME`. Results are in Table 4.

As with the other metrics, the WLS topology yields the best fit of the three topologies examined. The residue for the WLS tree using this method was better than using the mean-square distance, but worse than for the other two methods examined. The WLS tree is in Fig. 4. This tree was the best of all the trees we saw, over all the methods, at separating the four major groups into separate subtrees in a clear manner. Even Test20, Test21, and Test2, the three samples that had fallen outside the EWS cluster in many tests, were all placed in the EWS subtree.

The only meaningful way to compare the fit of the various metrics and the various resulting trees simultaneously is to compare the residues. The trees built from the Euclidean metric had the smallest residues—of these, the best was the WLS tree shown in Fig. 1. This tree has most of the characteristics we seek: four major subtrees

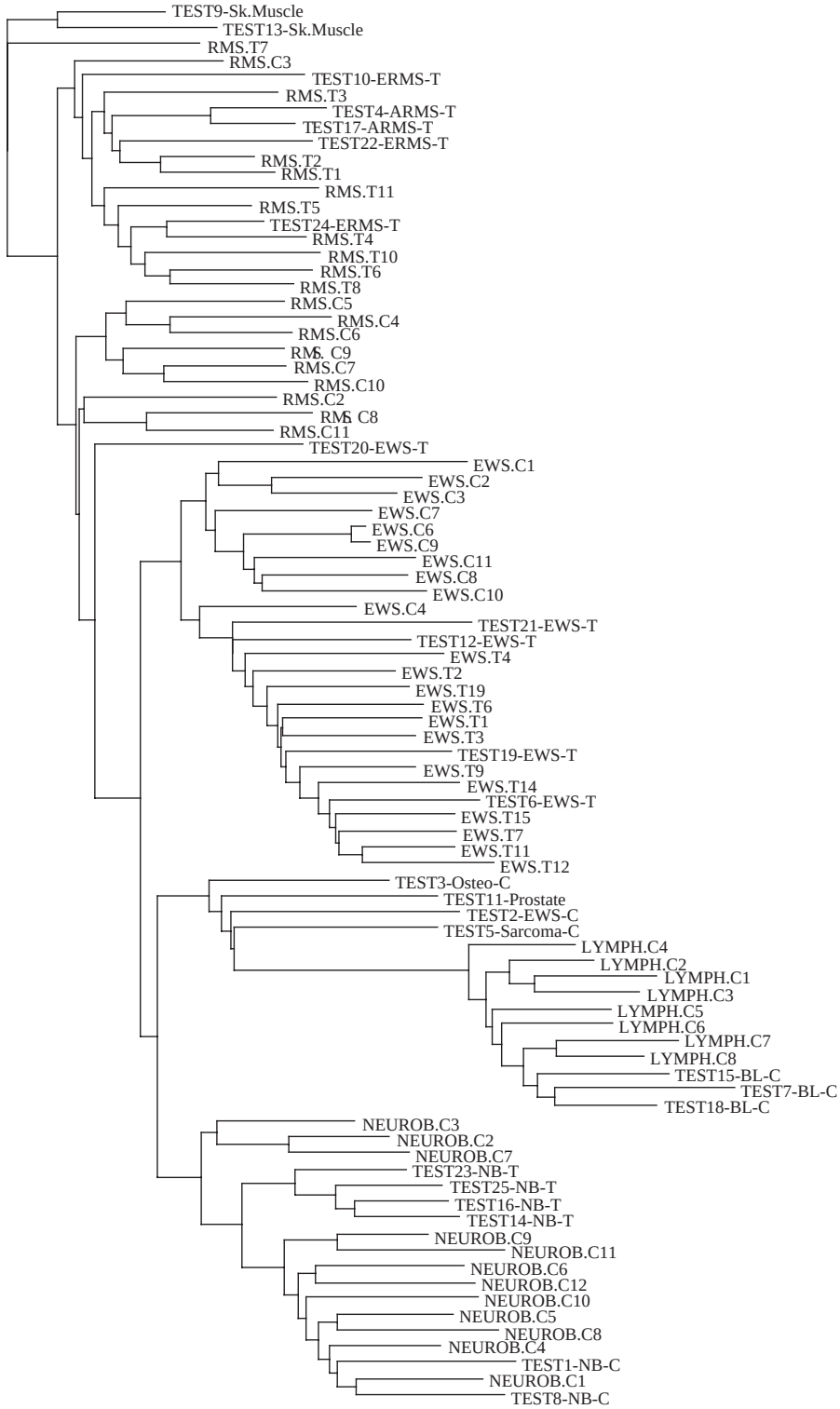

Fig. 2. Fitch/WLS tree for taxicab metric.

corresponding to the four cancer sub-types, and also tumor subtrees within the EWS, RMS, and NB subtrees. As the 96 genes were not selected for their ability to discriminate between tumors and non-tumors, the presence of these tumor subtrees represents true class discovery.

The three non-SRBCT cancer samples were placed in an unresolved position in the center of the tree, as were,

Table 3
Fit of tree metrics to mean-square metric

	L_1 distance	L_2 distance	L_∞ distance	Residue
ME tree	1302.81414	19.398149	0.88210	0.109282
WLS tree	1254.32242	18.518600	0.78351	0.105214
NJ tree	1310.57464	19.555791	0.93178	0.109933

Table 4
Fit of tree metrics to mean-square absolute distance

	L_1 distance	L_2 distance	L_∞ distance	Residue
ME tree	1069.11540	16.048909	0.88510	0.095066
WLS tree	1026.39922	15.203374	0.76618	0.091268
NJ tree	1071.83410	16.139465	0.93669	0.095308

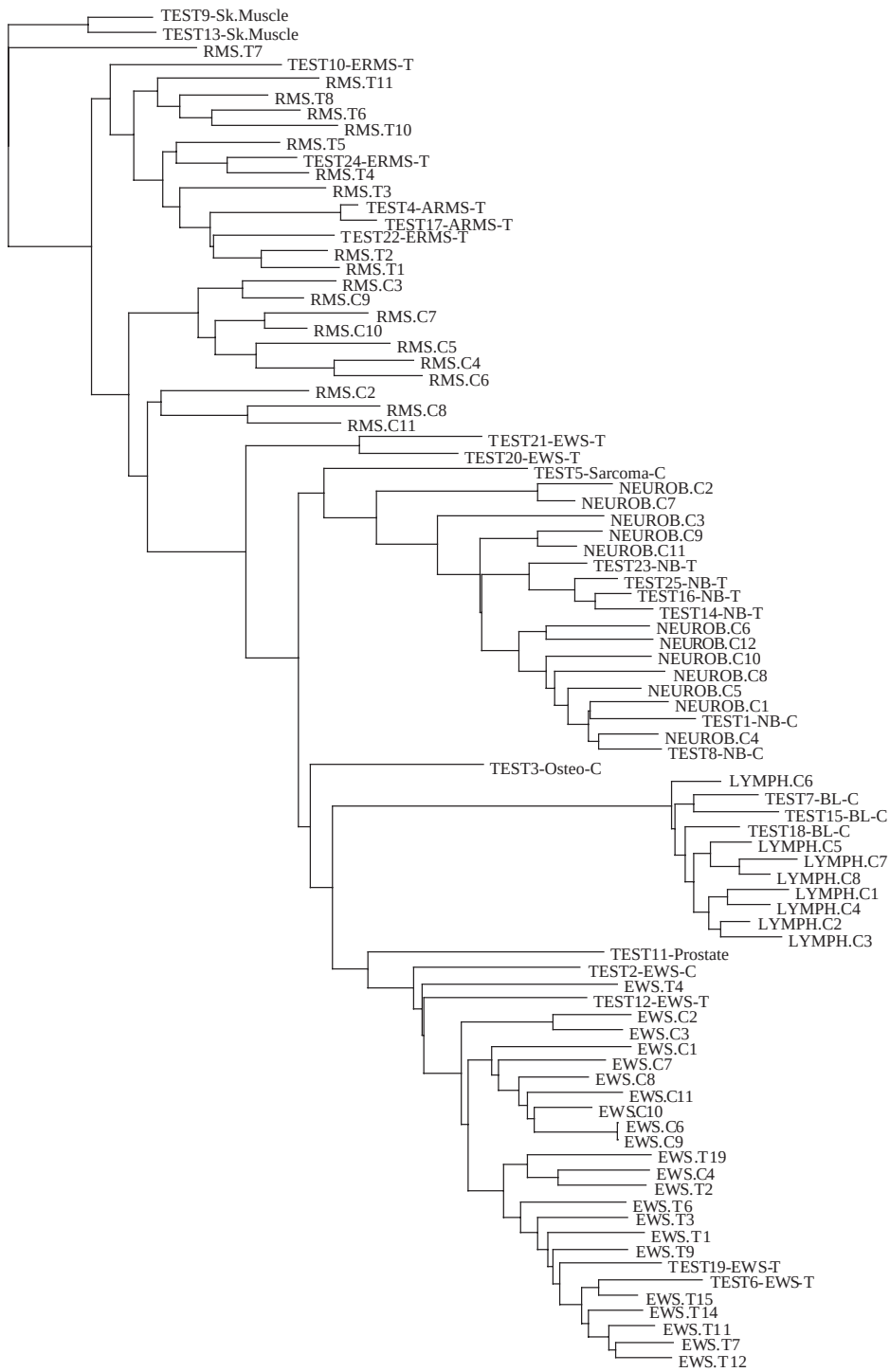

Fig. 3. Fitch/WLS tree for mean-squared metric.

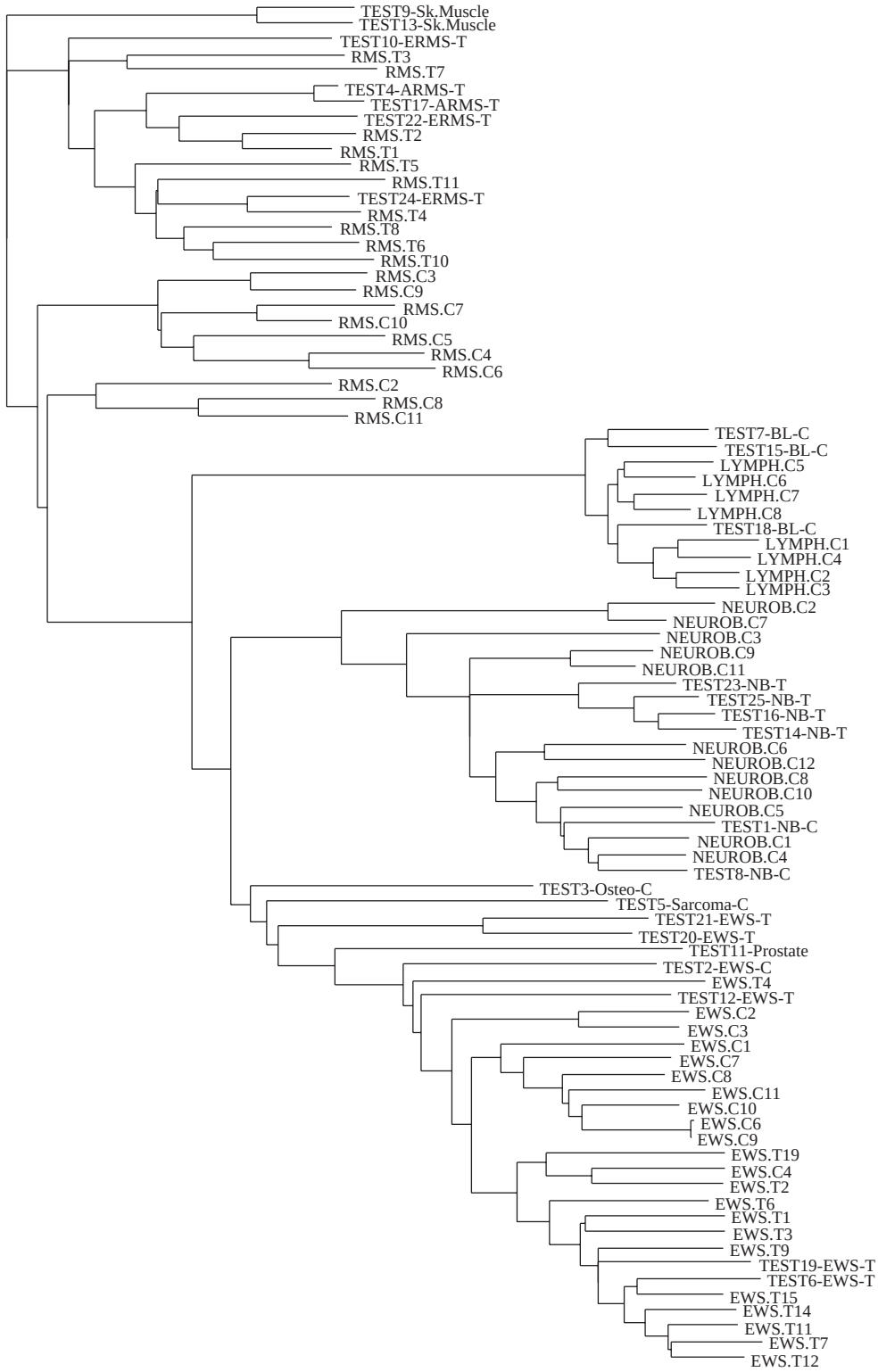

Fig. 4. Fitch/WLS tree for mean-squared absolute distance.

unfortunately, two EWS test samples. Also, the two skeletal muscle samples grouped together in the RMS subtree; this placement was consistent with the observation in Khan et al. that the expression profiles of these two samples were closer to the profiles of the RMS group than to any of the other groups.

5. Robustness

5.1. Jackknife test

To test the stability of the classification tree method, we performed two different tests. First, we performed a jackknife test. The jackknife method (Efron, 1982) consists of taking one data point from the data and omitting it, and then performing the analysis on the remainder of the data. Traditional leave-one-out analysis removes one data point and creates a clustering on the remaining (n-1) data points. It then inserts the nth sample into the fixed pre-existing clustering. With our methods, the clustering is not fixed on the subset of (n-1) data points, so we did not pursue such an analysis.

Instead of deleting one tissue sample at a time, we tested the robustness of the tree structure by jackknifing the genes. We considered each of the 96 data sets generated by omitting one gene, and each resulting tree. From this set of 96 trees, we formed a consensus tree, using a standard tree consensus program `consense` from the PHYLIP package. This program forms a consensus tree including all edges included in more than half of the 96 input trees, and some other edges also, to form a bifurcating tree. Each edge in the tree has a jackknife value: the number of input trees that contained an edge corresponding to the same split of leaves (tissue samples). The idea to use `consense` in this manner was suggested by (Belbin et al., 2002).

We performed such a jackknife test for each of the three algorithms, for each of the three metrics under consideration. The consensus WLS tree for the Euclidean metric was of the same topology as the WLS tree on the entire data set. The consensus WLS tree for the taxicab metric is shown in Fig. 5. Each internal node in the tree is labeled with a value between 1 and 96. This number is called the jackknife value: it counts the number of trees constructed from the partial data sets which include the split defined by the subtree to the right of said node. For example, all 96 jackknife trees include the split defined by the four neuroblastoma tumor samples, but only 81 jackknife trees contain the split {NEUROB.C5, NEUROB.C8}. Note: in contrast to the edge lengths in the earlier tree figures output by the distance algorithms, the horizontal distances in the consensus trees have no meaning.

Most of the splits corresponding to our predefined clusters have very high jackknife values in the consensus tree: the BL and NB subtrees have jackknife values of 96, the maximum possible, and the major EWS subtree has a jackknife value of 89. Some ambiguity is caused by the difficulty our algorithms had in placing the sample labeled TEST20-EWS-T. In fact, referring back to the original analysis (Khan et al., 2001) of this data point, we see that the initial classification of this sample as EWS was tenuous: only 40% of the ANN models placed this sample within the EWS cluster, while 30% placed it within the RMS cluster.

Figs. 6–9 summarize the relevant jackknife values for the various trees with regard to the Euclidean metric, the taxicab metric, the mean-squared metric, and the absolute mean-squared metric. The four major groups were preserved in all the consensus trees, as were the tumor subgroups of RMS, NB, and EWS. In some cases, the cell line subgroup was split into two subtrees in each of these groups. In those cases, the jackknife values given below reflect the larger respective subtrees.

We see that the consensus trees for all four metrics have full support for the BL, NB, NB.T and NB.C subtrees. The Neighbor-Joining algorithm yields much lower jackknife values, however, for the other subtrees. The Fitch/WLS approach is less sensitive (i.e. more consistent with higher jackknife values) for the mean-squared metric, while the Minimum Evolution tree is slightly superior to the WLS tree for the Euclidean metric and the absolute mean-squared metric. Of the four metrics, it is clear that the taxicab metric is the most sensitive to small changes in the data set (and hence the least trustworthy).

5.2. Noise test

When using a hierarchical clustering algorithm, one must consider how it performs in the presence of noise. To test the robustness of the classification tree method, we performed the following test 100 times:

- For $k = 1$–100, we created a new input matrix $A_{(k)}$ from A by multiplying each entry a_{ij} by $(1 + \delta_{ij})$, where the values (δ_{ij}) are chosen independently from a normal distribution with mean $\mu = 0$ and standard deviation $\sigma = 1$.
- We then performed the steps in Section 4 on each of these modified data sets, to produce trees $T_1, T_2, \ldots, T_{100}$.
- We then again used the program consense from the PHYLIP package to form a consensus tree on the data.

An example of such a tree is in Fig. 10. As with the jackknife tree, the numbers assigned to the internal nodes reflect how many of the 100 noisy samples produced a `FastME` tree with the corresponding split. Also, we note with interest that this consensus tree correctly classifies the TEST20 sample, which was difficult for the original algorithm to classify.

Since the amount of noise chosen was arbitrary, the consensus tree resulting after noisy perturbations is not being presented with a statistical justification for the original topology. Further analysis would be required to make such an argument. The topology would become increasingly unstable if the amount of noise added were

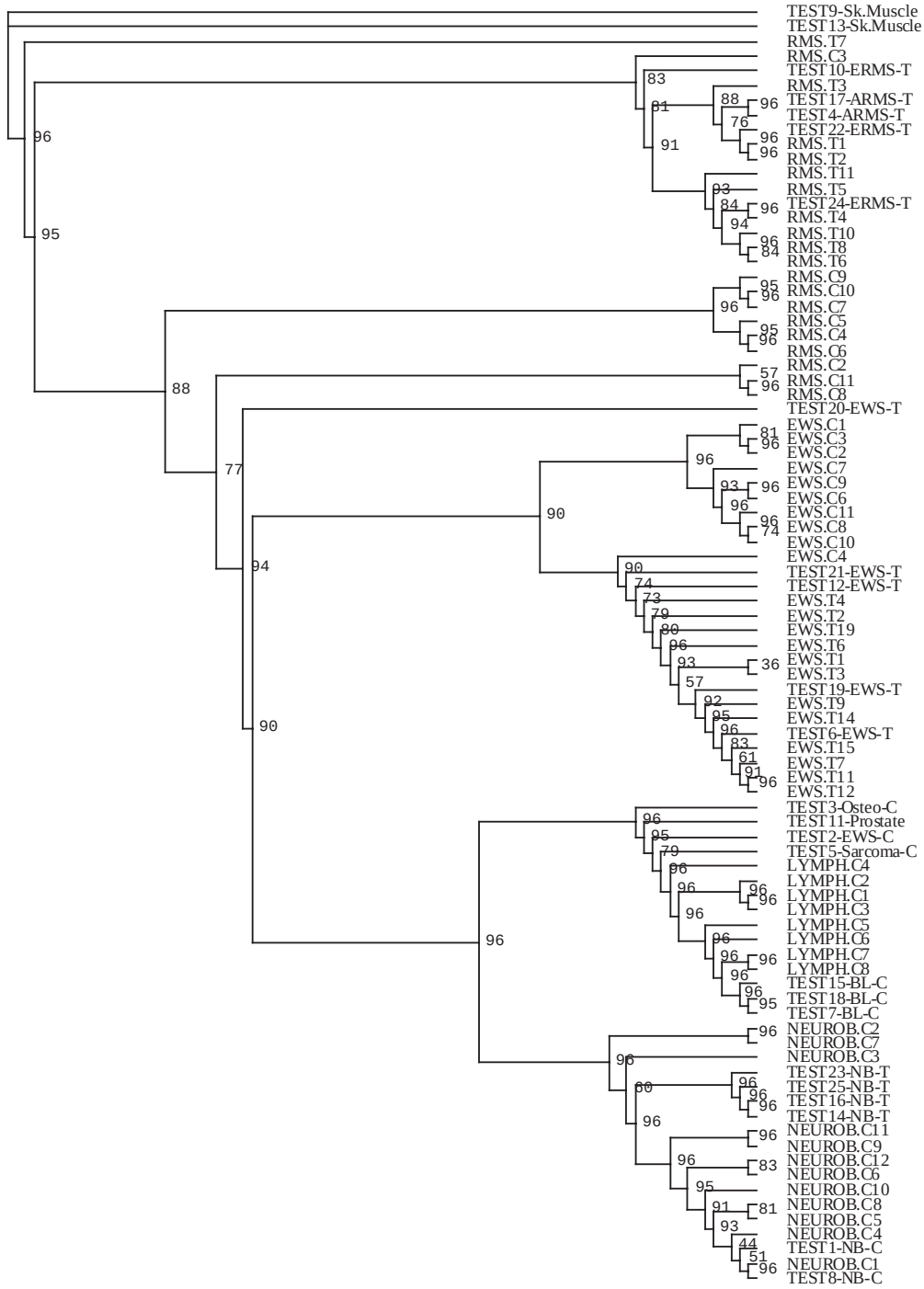

Fig. 5. Jackknife Fitch tree for taxicab metric.

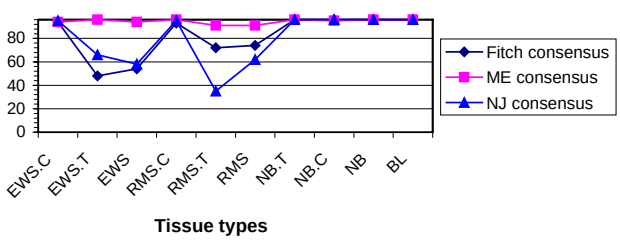

Fig. 6. Jackknife values—Euclidean metric.

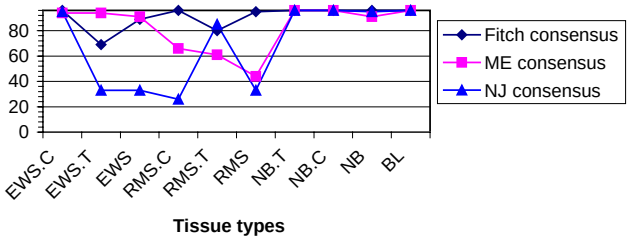

Fig. 7. Jackknife values—Taxicab metric.

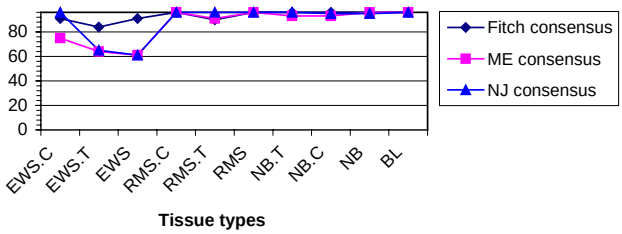

Fig. 8. Jackknife values—Mean-square metric.

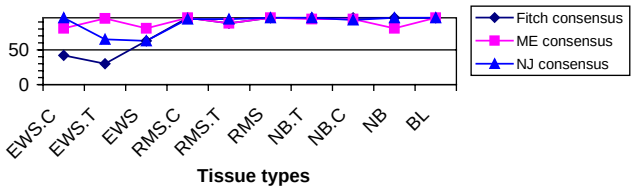

Fig. 9. Jackknife values—Absolute m.s. metric.

to increase; however, the degree to which one should expect this to happen is beyond the scope of the current work. Nevertheless, in a manner similar to the jackknife test, the noise sensitivity test provides some information as to the relative confidence of various branches within one tree. Also, both jackknifing and noise sensitivity testing give us a good idea as to the relative merit of the various metrics and various tree-fitting algorithms for our purposes.

The results of the noise resampling tests are summarized in Figs. 11–14. The noise sensitivity tests neatly parallel the jackknife tests. Again, the taxicab metric is seen to be the most sensitive, and least appropriate, of the four metrics. However, the consensus NJ tree on the perturbed Euclidean metrics could not accurately reconstruct the RMS and EWS subgroups. The `FastME`

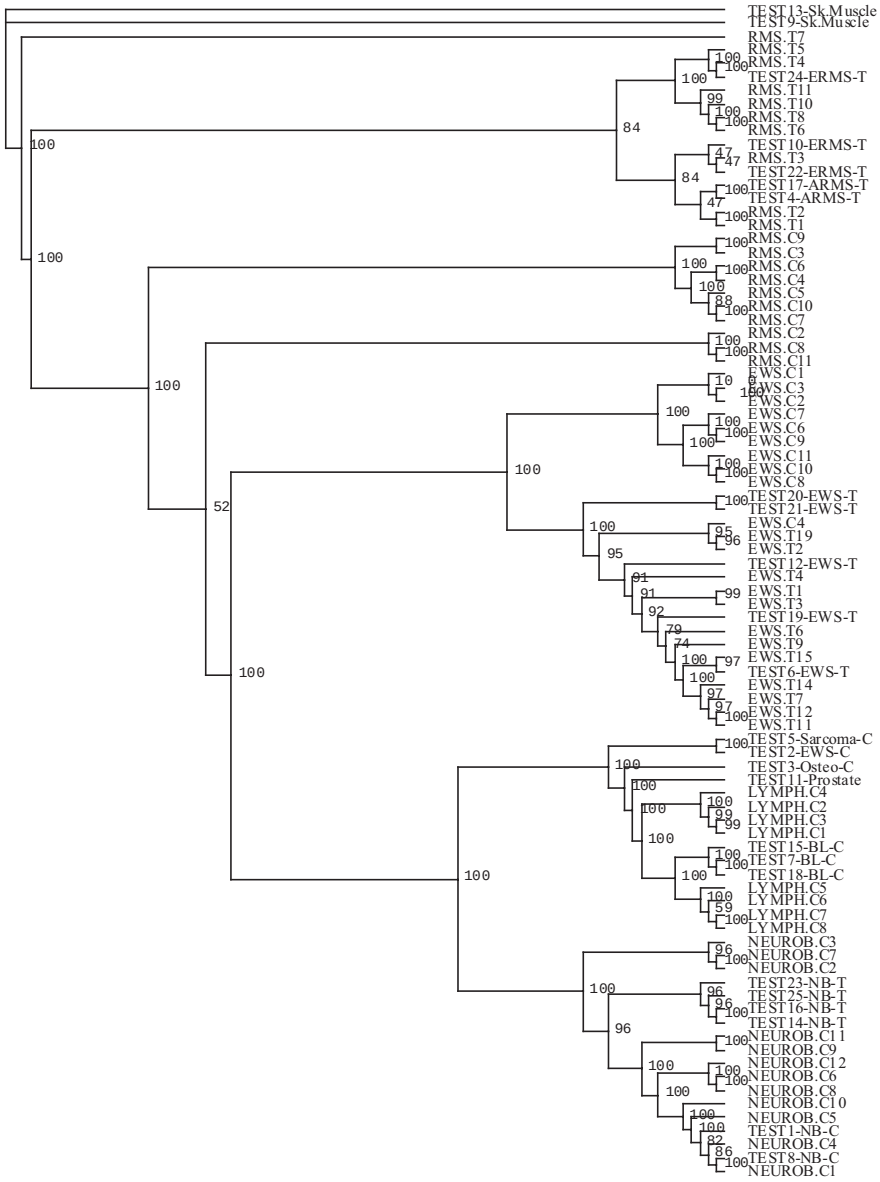

Fig. 10. ME consensus tree for noise resampling tests using Euclidean metric.

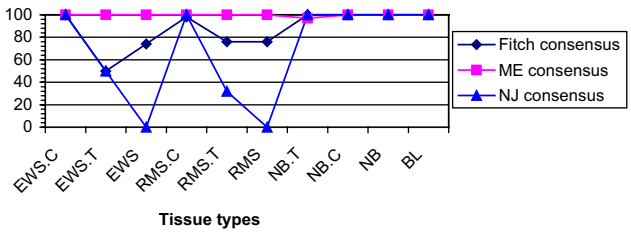

Fig. 11. Noise resistance—Euclidean metric.

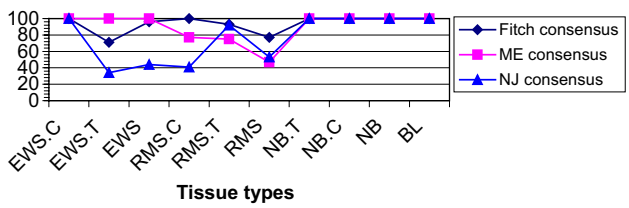

Fig. 12. Noise resistance—taxicab metric.

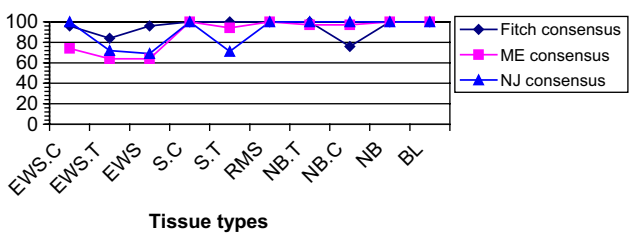

Fig. 13. Noise resistance—mean-square metric.

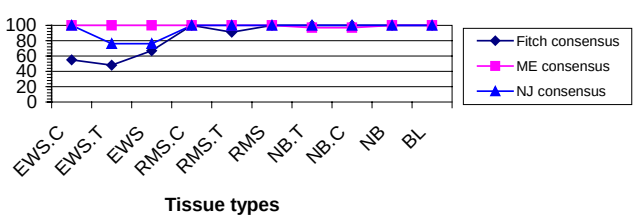

Fig. 14. Noise resistance—absolute m.s. metric.

tree was extremely consistent again with the Euclidean metric and the absolute mean-squared metric, and the Fitch/WLS tree was again the best for the mean-squared metric. Based on the above results, we would conclude that the taxicab metric is inappropriate, and would recommend using `FastME` for the Euclidean metric or absolute mean-squared metric, and Fitch/WLS for the mean-squared metric.

6. Classification of breast cancer tumors

Having established a general protocol in our examination of the SRBCT data set, we tested this protocol against a second data set. We considered the data set from Hedenfalk et al. (2001), consisting of 3225 genes measured from 22 breast cancer tumors. The tumors divided into three groups: seven tumors with BRCA1 mutations, eight tumors from seven patients with BRCA2 mutations, and seven sporadic tumors, one of which contained a hypermethylated BRCA1 promoter region. Hedenfalk et al. considered only two binary classification questions, namely, whether each tumor carried a BRCA1 or BRCA2 mutation, respectively. Our attention focused on 51 genes, selected by Hedenfalk et al. using a method based on an F-test to be those genes whose variation in expression best differentiated among the three types of cancers.

We considered two metrics on this data set: the Euclidean metric and the mean-squared metric. (We dismissed the mean-squared absolute metric after noting that it was identical to the mean-squared metric for this data set.) For each metric, we considered the Minimum Evolution tree and the Fitch tree, yielding four trees altogether. All four trees shared the following features:

- All seven of the BRCA1 tumors clustered together in one subtree.
- All eight of the BRCA2 tumors clustered together in another subtree.
- The seven sporadic tumors lay between the BRCA1 and BRCA2 subtrees, mostly as leaves off a main path between the two main clusters.
- The sporadic tumor with a methylated BRCA1 promoter region was the sporadic tumor closest to the BRCA1 cluster.

The trees resulting from the correlation metric were of particular interest, as the edges connecting the sporadic tumors to the tree were much longer, on average, than the edges pendant to the BRCA1 or BRCA2 tumors. In fact, the six longest edges in the tree are pendant edges connecting six of the seven sporadic tumors to the backbone of the tree. Fig. 15 shows the fitch tree for the correlation metric, which displays all of the properties listed above. This tree is shown in radial form to give a better picture of relative distances.

We also created jackknife trees for each of the two metrics, and each of the two tree-building algorithms. In each of the four resulting jackknife trees, the majority of the jackknife values were perfect scores of 51, and nearly all of the jackknife values were well above 40. The Euclidean metric was slightly less sensitive to jackknifing; the `fitch` jackknife tree for this metric is shown in Fig. 16.

7. A class prediction algorithm using phylogenetic methods

Class discovery is just one of the possible applications of phylogenetic methods to the field of expression data

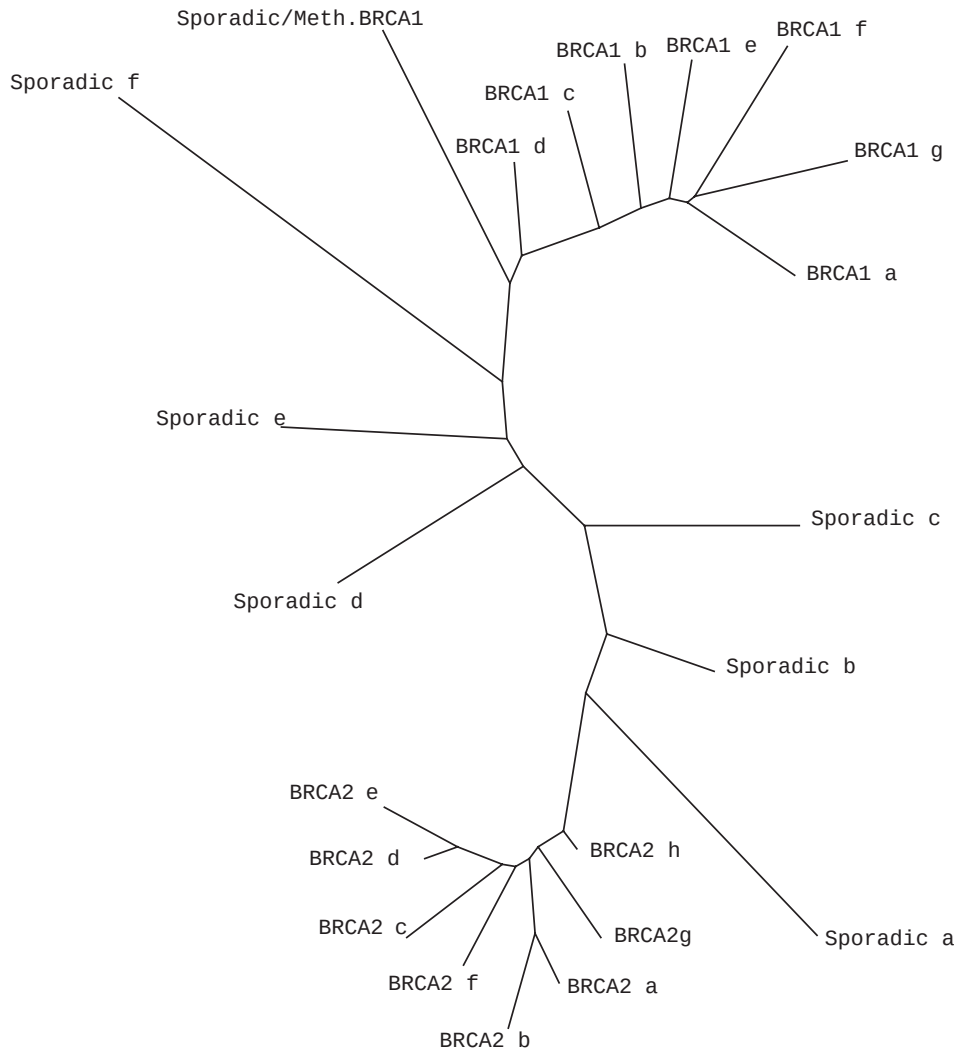

Fig. 15. Breast cancer tumors, classified using Fitch algorithm on mean-square metric.

analysis. In this section, we examine a method for class prediction using taxonomic distance methods.

7.1. The class prediction algorithm

Recall that, in the field of tumor classification, *class prediction* is the process of assigning labels to a set of unknown tumors using information gained from a set of labeled tumors. Our approach will be to use a classification tree generated from tumors that have been previously classified. Suppose we use $i=1\ldots t$ to label the different types of tumors, and we are given a learning set L, $L=\bigcup_{i=1}^{t} L_i$, such that all of the tumors in L_i are of type i. Given a test set X, the general method is as follows:

- Apply the methods of Section 3.2 to define a metric Δ on L.
- Use ME, NJ or a least-squares method to define a classification tree T on L.
- Check that the topology of T agrees with the partition $L=\bigcup_{i=1}^{t} L_i$ i.e., each subset L_i induces a subtree of T. If not, T is not suitable as a guide to future classification. (Note: this rule need not be strict. A tree that comes very close to perfectly partitioning the data might be usable if a small number of offending tumors were trimmed from the tree. This difficulty did not arise with the SRBCT data set.)
- For each $x \in X$, define the metric Δ_x on the set $L \cup \{x\}$.
- Use an optimization method (least squares or minimum evolution) and the metric Δ_x to find the optimal insertion point for x in T to create the tree T_x.
  - For each edge $e=(u,v)$ in T, let T_e be the tree formed by removing e, creating a

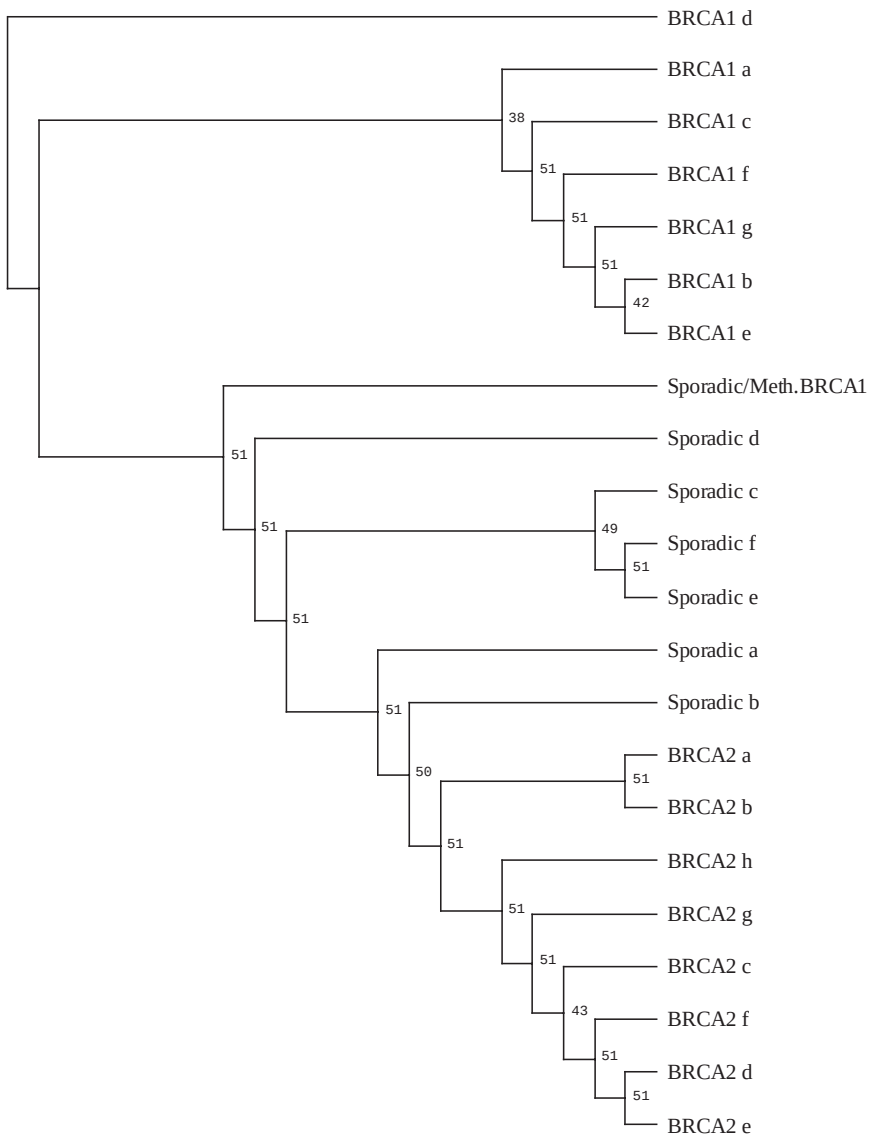

Fig. 16. Fitch jackknife tree on breast cancer tumors using Euclidean metric.

new node w, and adding the edges (u, w) (w, v) and (w, x).

- ○ Let $f(e)$ be the value of T_e under the given optimization criterion (e.g. if using the minimum evolution criterion, $f(e)$ would be the sum of the edge lengths of T_e).

- Choose $T_x = T_e$ such that $f(e)$ is optimal (minimal for our measures of optimality).
- If the placement of x is within one of the trees T_i generated by the set L_i, then x is predicted to be of class i. If x falls outside of all the subtrees $T_1, \ldots T_t$ we leave x unclassified.

7.2. Results

We tested the method of Section 7.1 on the SRBCT data set. This data set was divided by Khan et al. into a set of 63 labeled samples and a set of test samples including 20 SRBCTs, 3 non-SRBCT tumor samples, and 2 samples from healthy tissue (skeletal muscle). One of the SRBCT test samples was from a patient that had also provided a learning sample. The two samples in question had virtually identical expression profiles, so we discarded the second (test) sample from the patient in question, as its inclusion would be uninteresting.

We decided to use the Minimum Evolution method on the Euclidean metric, as this combination had proven to be highly consistent in the jackknife and noisy sampling tests. This tree neatly divides into four subtrees corresponding to the four types of SRBCT.

All of the 19 SRBCT test samples were placed in the correct subtrees by this method, and the other two non-SRBCT samples were optimally placed in the

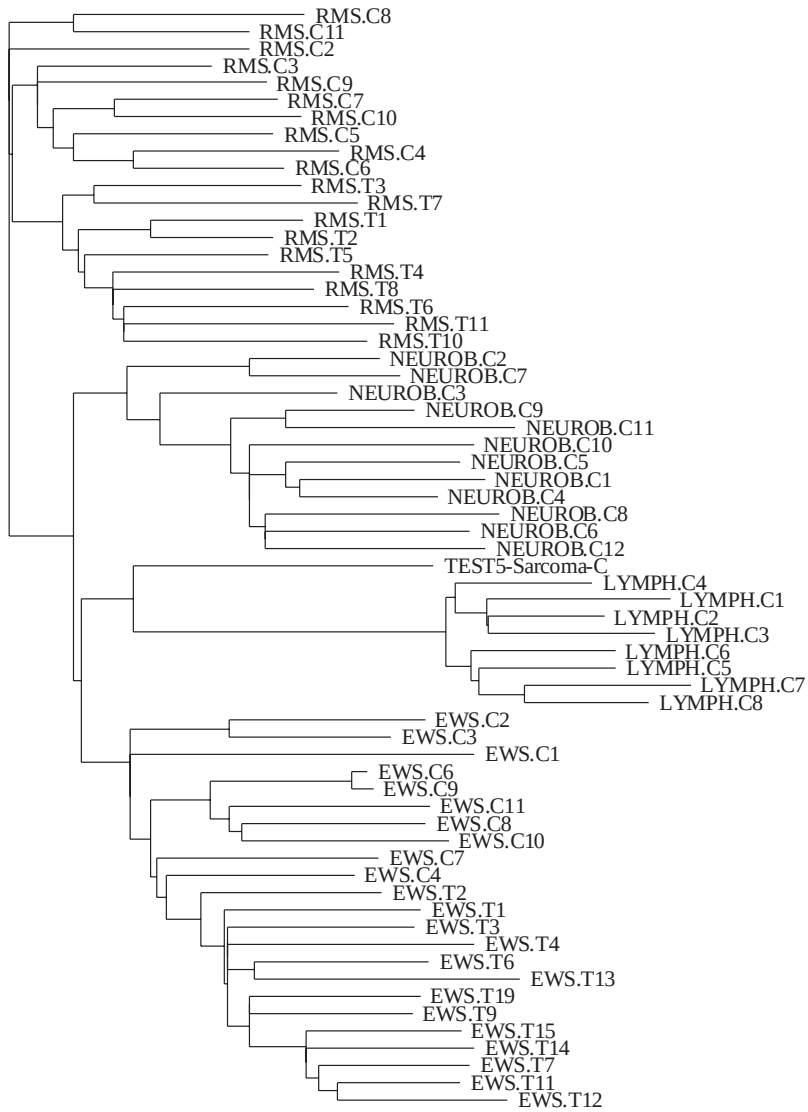

Fig. 17. Minimum evolution tree on SRBCT learning set and sarcoma test sample.

center of the tree. Fig. 17 shows the optimal insertion point for the sarcoma test sample, in the interior of the tree, separated from the four major subtrees. Fig. 18 shows the optimal insertion point for TEST19, one of the EWS tumor samples. It is optimally placed in the EWS-T subtree. The two skeletal muscle samples were placed in the RMS subtree, which is consistent with their placement in the class discovery trees, and with their placement by the ANN methods of Khan et al.

8. Comparison to other methods and discussion

We have presented a new method of clustering of microarray data from tumors. The phylogenetic method has the advantage that the clusters are not fixed, allowing reasonable placement of samples that do not belong to any of the tumor categories. The geometric representation of the output gives some idea of how well the samples cluster. By using a jackknife test we can get a concrete measure of the confidence in each split in the clustering. By constructing the consensus tree from the trees created by repeatedly adding noise to the data, we can get a numerical assessment of the confidence in the clustering. We have demonstrated our method on the SRBCT data set of (Khan et al., 2001) with good results.

The same SRBCT data set was used Culhane et al. (2002) to test a method of supervised clustering called "Between Group Analysis". Their method also gave good results, but had difficulty placing one of the neuroblastoma test samples, which did not cause difficulty for any other methods. Grate et al. (2002) also used SRBCT data set as a test of a two-category supervised classification method, but they distinguished between cell line samples vs. tumor

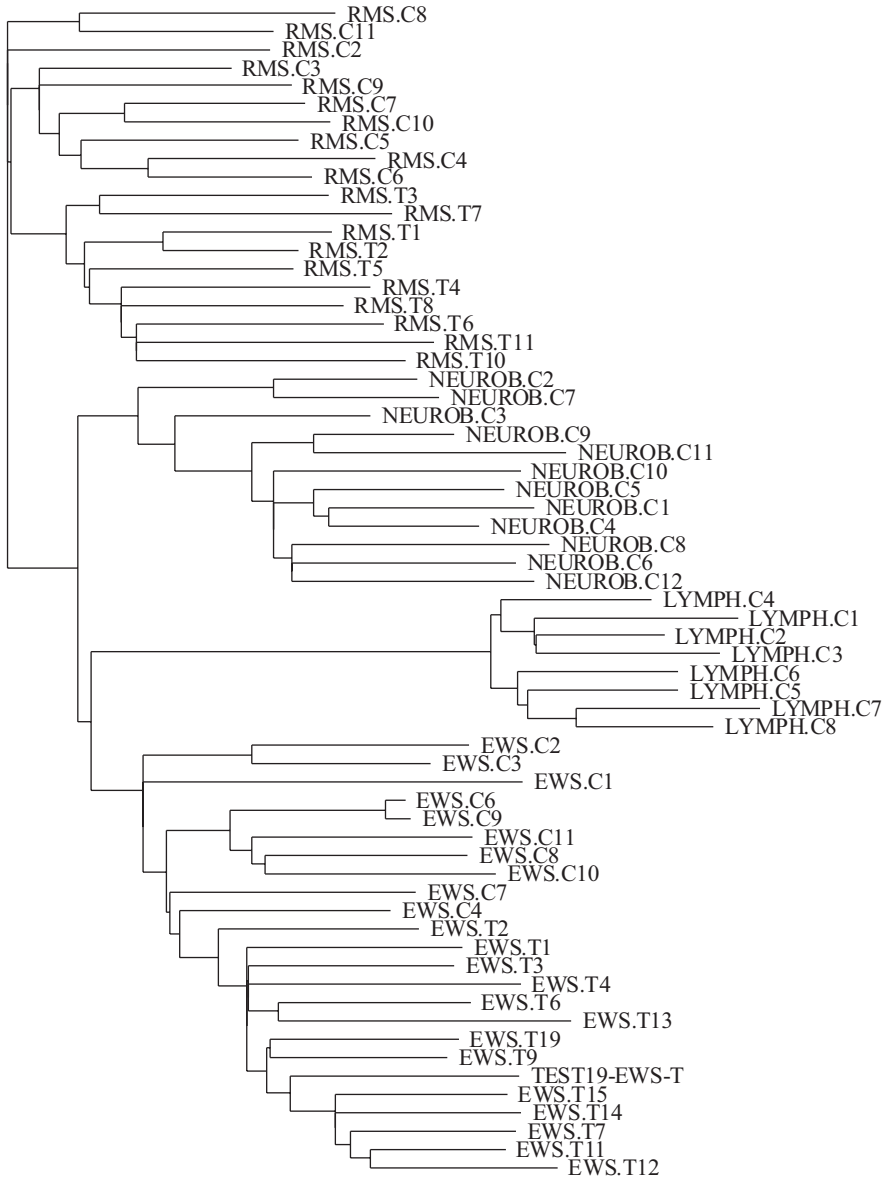

Fig. 18. Minimum Evolution tree with EWS test sample inserted.

samples, instead of attempting to distinguish the four categories of tumors.

In this section, we compare the phylogenetic method to other unsupervised class discovery methods. Note that we use the word "unsupervised" even though the process of gene selection was itself supervised. It would be desirable to develop a method to select the most important genes without using a training set, but such a task is beyond a scope of the current paper. None of the algorithms we consider actually use the tumor labels while building clusterings, thus they still qualify as unsupervised algorithms even when the data selection process is supervised. This usage of "unsupervised" is consistent with (Langley, 1996, Section 8.4).

We tested the widely used GENECLUSTER software (Tamayo et al., 1999, version 2.1) and the EXCAVATOR software (Xu et al., 2002, version 1.0) because they seemed close in spirit to our approach. We also tested the widely used hierarchical clustering program CLUSTER (Eisen et al., 1998, version 2.13.1). All three packages were tested using default settings on the same 87 SRBCT set with 96 genes. We used the normalization described above, not the normalization options in these three software packages. GENECLUSTER and EXCAVATOR are both easy to install and use, and the outputs are easy to interpret.

We tried GENECLUSTER using 2–6 clusters, and we summarize here the results using 4 clusters, which unsurprisingly gave the best results. For 4 clusters the output geometry can be either 1×4 or 2×2, and these gave slightly different results because TEST5-Sarcoma-C was placed in the NB cluster in the 2×2 geometry and

in the EWS cluster in the 1×4 geometry. In either geometry, 19/20 test samples that are SRBCTs were placed correctly, and the TEST20-EWS-T sample was placed incorrectly in the RMS cluster. TEST20-EWS-T and 4 of the 5 non-SRBCTs ranked 2nd–6th highest in "distance to centroid" of their clusters, giving some indication that they are outliers. However, this criterion is not perfect; the highest distance was for TEST21-EWS-T, while TEST13-Sk.Muscle ranked 14th or 15th highest in "distance to centroid". The principal disadvantage of the GENECLUSTER method for unsupervised learning is that it does not identify the preferred number of clusters.

The EXCAVATOR software uses an information theoretic criterion and correctly predicted that there are 4 clusters in the SRBCT data. All samples, except for TEST20-EWS-T were correctly classified. The TEST3-Osteo-C, TEST5-Sarcoma-C, and TEST11-Prostate samples all come out as leaves in the EWS cluster. The two skeletal muscle samples form a pendant "cherry" (two-leaf) subtree in the RMS cluster. The edges connecting TEST20-EWS-T and the 5 non-SRBCTs to their clusters are 5 of the 9 longest edges in the output, again giving an imperfect indication that these classifications are suspect. The principal disadvantages of EXCAVATOR are that it does not come with a visualization tool and it does not provide any assessment of confidence in the predictions.

We tested Eisen's clustering software CLUSTER on the SRBCT data set. The resulting clustering was inferior to the classification tree produced by the phylogenetic method. CLUSTER placed the sample labeled NEUROB.C3 in the rhabdomyosarcoma cluster. Also, two Ewing's sarcoma samples, EWS.T13 and EWS.T4 were also falsely placed in the in RMS cluster; an alternative explanation is that these two EWS samples and the sample labeled RMS.T7 fall outside any large cluster. CLUSTER's weaknesses motivated the development of GENECLUSTER, EXCAVATOR, Between Group Analysis, etc.

The Fitch trees were the best at fitting the various metrics we considered. The Fitch tree for the mean-square metric accurately classified all the SRBCTs into four major clusters, including the TEST20 sample, which had proved troublesome for other classification methods. Unlike simple hierarchical methods, fitted trees contain information reflecting the relative confidence in various clusters and subclusters, namely the length of the respective edges. They also lend themselves to quick resampling analysis.

Jackknifing is a quick way to test the confidence of each cluster, and noise perturbation demonstrates the robustness of the solution. The resulting consensus trees demonstrated which subtrees were more reliable than others. Furthermore, the consensus trees produced superior topologies than simple tree fitting done alone.

We also tested the phylogenetic method on the breast cancer data set of (Hedenfalk et al., 2001). The tree-fitting methodology neatly created classification trees that separated tumors with BRCA1 mutations from tumors with BRCA2 mutations. Sporadic tumors appear in the tree as topological leaves relatively distant from the path between the two main clusters.

In Section 7, we demonstrated the utility of phylogenetic methods for the problem of class prediction. The classification tree generated by the learning set neatly separated all of the cancer subtypes in question. Using a simple minimum evolution criterion, we found optimal insertion points for all of the data in the test set. All of the SRBCT samples were properly classified, and the three non-SRBCT cancers were placed outside of any of the four groups. Although most of the class discovery methods could easily be similarly modified to approach the question of class prediction, we did not pursue this idea, as it would likely require modifying software for which source code is not distributed.

We have considered the applicability of standard tree fitting methods from the field of phylogeny to the problems of class discovery and class prediction from tumor expression array data analysis. This avenue of research is similar to the CLUSTER package that produces hierarchical clusterings using Pearson correlation, as well as the program EXCAVATOR of Xu et al. that finds minimal spanning trees. All three approaches result in tree structures on the set of tumors. But where both hierarchical clustering and minimal spanning trees are iteratively formed based entirely on the iterative consideration of pairwise data, the trees resulting from phylogenetic software work to minimize global functions that implicitly force the simultaneous consideration of all first-order correlations. Also, in contrast to the dendograms produced by the simple hierarchical method, fitted trees contain information reflecting the relative confidence in various clusters and subclusters, namely the lengths of the respective edges.

In an unsupervised setting, the phylogenetic approach does not require the user to preset a fixed number of clusters—this is an advantage it shares with EXCAVATOR over the geometric algorithms such as k-means and the SOM approach of Tamayo et al. Also, the tree structure lends itself for easy resampling analysis using noise perturbation or jackknifing (or bootstrapping for larger data sets). The tree structure can lead to a cluster interpretation in some cases. The obvious method for selecting a cluster would be to search for large subtrees connected by long edges, such as the Lymphoma subtree in Fig. 1. Also, one could use resampling techniques and select subtrees with high jackknife or bootstrap values. But the transition from a phylogeny to a clustering dendogram would entail discarding information and thus we feel that the weighted tree should be considered the final product of this approach.

In conclusion, tree fitting is a useful tool for classifying tumors based on expression data. In the present work, we have demonstrated its suitability for the problem of class discovery, with the caveat that we used supervised gene selection—a problem that needs to be addressed in future work. We have also demonstrated suitability for the problem of class prediction, with no such caveat. For both problems, the Euclidean metric was very useful in conjunction with a minimum evolution approach, as was the mean-square metric in conjunction with a weighted least-squares approach. We would recommend using both combinations for classifying data, and would also recommend using resampling techniques to verify the various topological features of any classification tree. Software to transform microarray output files into input files for tree-fitting programs, and to compare various output tree metrics, is available by e-mail from the authors.

Acknowledgements

Our thanks go to John Powell for his assistance installing CLUSTER. Also, we thank three anonymous referees for many helpful suggestions that led to substantial improvements in the manuscript.

References

Alizadeh, A.A., Eisen, M.B., Davis, R.E., Ma, C., Lossos, I.S., Rosenwald, A., Boldrick, J.C., Sabet, H., Tran, T., Yu, X., Powell, J.I., Yang, L., Marti, G.E., Moore, T., Hudson, J., Lu, L., Lewis, D.B., Tibshirani, R., Sherlock, G., Chan, W.C., Greiner, T.G., Weisenburger, D.D., Armitage, J.O., Warnke, R., Levy, R., Wilson, W., Grever, M.R., Byrd, J.C., Botstein, D., Brown, P.O., Staudt, L.M., 2000. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature 403, 503–511.

Belbin, T.J., Singh, B., Barber, I., Socci, N., Wenig, B., Smith, R., Prystowsky, M.B., Childs, G., 2002. Molecular classification of head and neck squamous cell carcinoma using cDNA microarrays. Cancer Res. 62, 1184–1190.

Chen, Y., Bittner, M.L., Dougherty, E.R., 1999. Issues associated with microarray data analysis and integration (Supplement to Bittner, M., Trent, J., Meltzer, P., 1999. Data analysis and integration: of steps and arrows). Nature Genet. 22, 213–215.

Culhane, A.C., Perrière, G., Considine, E.C., Cotter, T.G., Higgins, D.G., 2002. Between-group analysis of micorarray data. Bioinformatics 18, 1600–1608.

Desper, R., Gascuel, O., 2002. Fast and accurate phylogeny reconstruction algorithms based on the minimum evolution principle. J. Comput. Biol. 9, 687–705.

Dudoit, S., Fridlyand, J., Speed, T.P., 2002. Comparison of discrimination methods for the classification of tumors using gene expression data. J. Am. Stat. Assoc. 97, 77–87.

Efron, B., 1982. *The Jackknife, the Bootstrap, and Other Resampling Plans*, CBMS-NSF Regional Conferenc Series in Applied Mathematics, Vol. 8, SIAM, Philadelphia.

Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D., 1998. Cluster analysis and display of genome-wide expression patterns. Proc. Natl Acad. Sci. USA 95, 14863–14868.

Felsenstein, J., 1989. Phylip—phylogeny inference package (ver. 3.2). Cladistics 5, 164–166.

Fitch, W.M., Margoliash, E., 1967. Construction of phylogenetic trees. Science 155, 279–284.

Furey, T.S., Cristianini, N., Duffy, N., Bednarski, D.W., Schummer, M., Haussler, D., 2000. Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics 16, 906–914.

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P, Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., Lander, E.S., 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286, 531–537.

Grate, L.R., Bhattacharyya, C., Jordan, M.I., Mian, I.S., 2002. Simultaneous relevant feature identification and classification in high-dimensional spaces. In: Guigó, R., Gusfield, D. (Eds.), *Algorithms in Bioinfirmatics, Proceedings of the Second International Workshop, WABI 2002*. Lecture Notes in Computer Science, vol. 2452. Springer, Berlin, pp. 1–9.

Hedenfalk, I., Duggan, D., Chen, Y., Radmacher, M., Bittner, M., Simon, R., Meltzer, P., Gusterson, B., Esteller, M., Kallioniemi, O.-P., Wilfond, B., Borg, Å., Trent, J., 2001. Gene-expression profiles in hereditary breast cancer. N. Engl. J. Med. 344 (8), 539–548.

Herwig, R., Poustka, A.J., Müller, C., Bull, C., Lehrach, H., O'Brien, J., 1999. Large-scale clustering of cDNA-fingerprinting data. Genome Res. 9, 1093–1105.

Huson, D.H., Smith, K.A., Warnow, T.J., 1999. Estimating large distances in phylogenetic reconstruction. In: Vitter, J.S., Zaroliagis, C.D. (Eds.), *Algorithm Engineering, Proceedings of the Third International Workshop, WAE '99*. Lecture Notes in Computer Science, 1668. Springer, Berlin, pp. 271–285.

Khan, J., Wei, J.S., Ringnér, M., Saal, L.H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C.R., Peterson, C., Meltzer, P.S., 2001. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. Nat. Med. 7, 673–679.

Langley, Pat. 1996. *Elements of Machine Learning*. Morgan Kaufmann Publishers, San Francisco.

Page, R.D.M., 1996. Treview: an application to display phylogenetic trees on personal computers. Comput. Appl. Biosci. 12, 357–358.

Radmacher, M., McSchane, L., Simon, R., 2002. A paradigm for class prediction using gene expression profiles. J. Comp. Biol. 9, 505–511.

Saitou, N., Nei, M., 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol. Biol. Evol. 4, 406–424.

Swofford, D., 1996. PAUP — Phylogenetic Analysis Using Parsimony (and other methods), Version 4.0.

Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E.S., Golub, T.R., 1999. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. Proc. Natl. Acad. Sci. USA 96, 2907–2912.

von Heydebreck, A., Huber, W., Poustka, A., Vingron, M., 2001. Identifying splits with clear separation: a new class discovery method for gene expression data. Bioinformatics 17 (1), S107–S114.

Waddell, P.J., Kishino, H., 2000. Cluster inference methods and graphical models evaluated on NC160 microarray gene expression data. Genome Inform. 11, 129–140.

Xu, Y., Olman, V., Xu, D., 2002. Clustering gene expression data using a graph-theoretic approach: an application of minimum spanning trees. Bioinformatics 18, 536–545.