Available online at www.sciencedirect.com
SCIENCE DIRECT
Neural Networks 18 (2005) 297–306
Neural Networks
www.elsevier.com/locate/neunet

Predictive neural networks for gene expression data analysis

Ah-Hwee Tan[a,*], Hong Pan[b]

[a]School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798, Singapore
[b]Genome Institute of Singapore, 60 Biopolis Street #02-01, Genome, Singapore 138672, Singapore

Received 27 July 2004; revised 17 January 2005; accepted 17 January 2005

Abstract

Gene expression data generated by DNA microarray experiments have provided a vast resource for medical diagnosis and disease understanding. Most prior work in analyzing gene expression data, however, focuses on predictive performance but not so much on deriving human understandable knowledge. This paper presents a systematic approach for learning and extracting rule-based knowledge from gene expression data. A class of predictive self-organizing networks known as Adaptive Resonance Associative Map (ARAM) is used for modelling gene expression data, whose learned knowledge can be transformed into a set of symbolic IF-THEN rules for interpretation. For dimensionality reduction, we illustrate how the system can work with a variety of feature selection methods. Benchmark experiments conducted on two gene expression data sets from acute leukemia and colon tumor patients show that the proposed system consistently produces predictive performance comparable, if not superior, to all previously published results. More importantly, very simple rules can be discovered that have extremely high diagnostic power. The proposed methodology, consisting of dimensionality reduction, predictive modelling, and rule extraction, provides a promising approach to gene expression analysis and disease understanding.

Keywords: Knowledge discovery; Gene expression analysis; Predictive modelling; Rule extraction; Feature selection

1. Introduction

Measurements of gene expression activities have provided a vast resource for medical diagnosis and disease understanding. Specifically, gene expression may provide the additional information needed to improve cancer classification and diagnosis (Slonim, Tamayo, Mesirov, Golub, & Lander, 2000). Many machine learning methods, such as Support Vector Machines (SVMs) (Furey et al., 2000), clustering (Alon et al., 1999), Self-Organizing Map (SOM), and a weighted correlation method (Golub et al., 1999), have been successfully applied to gene expression data. Although fairly high predictive performance accuracy has been obtained, most methods focus on diagnostic accuracy rather than extracting comprehensible knowledge. More recently, a method called *Emerging Patterns* has been proposed to identify gene groups characterizing specific disease classes from gene expression data (Li & Wong, 2002). To tackle the high feature dimensionality issue, a feature discretization algorithm based on entropy was used to identify the most discriminative genes before pattern discovery.

The main motivation of our work, similar to that of Li and Wong (2002), is to extract accurate as well as comprehensible knowledge from gene expression data. Specifically, we present a systematic and robust three-stage procedure for learning and extracting diagnostic knowledge from gene expression data (Fig. 1). First, feature selection is applied to the raw expression data so as to reduce the feature dimensionality to a manageable scale in accord with the number of samples available. Next, a predictive model of the gene data is learned based on the expression data in the reduced feature space. Finally, comprehensible knowledge in the form of rules are extracted from the predictive model for interpretation.

To build predictive models, we explore a class of self-organizing neural networks, known as predictive Adaptive Resonance Theory (predictive ART) networks (Carpenter, Grossberg, & Reynolds, 1991; Tan, 1995), for learning the linkages between gene expression data and diseases. Predictive ART networks are designed for fast and incremental learning of multidimensional pattern mappings. Members of predictive ART networks, such as fuzzy

* Corresponding author. Tel.: +65 6790 4326; fax: +65 6792 6559.
E-mail address: asahtan@ntu.edu.sg (A.-H. Tan).

0893-6080/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2005.01.003

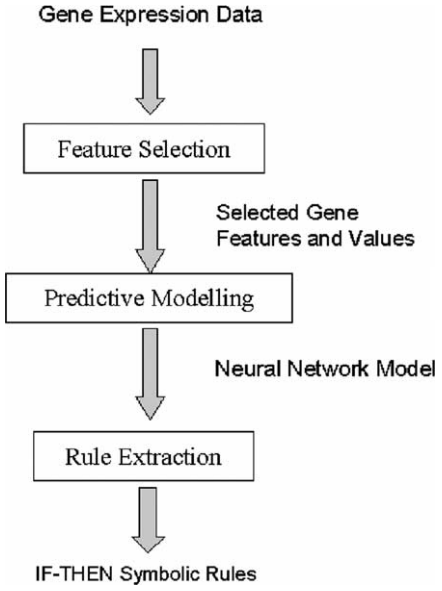

Fig. 1. The proposed methodology for gene expression analysis, consisting of feature selection, predictive modelling using neural networks, and rule extraction.

ARTMAP (Carpenter, Grossberg, Markuzon, Reynolds, & Rosen, 1992), ART-EMAP (Carpenter & Ross, 1993), and Gaussian ARTMAP (Williamson, 1996), have been successfully applied to a wide range of pattern analysis and recognition problems. However, to the best of our knowledge, there has been no attempt to date to use predictive ART networks for analyzing gene expression data.

In this paper, we adopt a simplified predictive ART architecture, known as fuzzy Adaptive Resonance Associative Map (fuzzy ARAM) (Tan, 1995), that produces classification performance equivalent to those of standard fuzzy ARTMAP. Fuzzy ARAM has been successfully applied to several machine learning tasks, including DNA promotor recognition (Tan, 1997), personal profiling (Tan & Soon, 2000), document classification (He, Tan, & Tan, 2003; Tan, 2001), and personalized content management (Tan, Ong, Pan, Ng, & Li, 2004). It has shown predictive performance comparable, if not superior, to those of many state-of-the-art learning-based systems, including C4.5 (Quinlan, 1993), Backpropagation Neural Network (Rumelhart, Hinton, & Williams, 1986), K Nearest Neighbour, and Support Vector Machines (Joachims, 1998). When performing classification tasks, fuzzy ARAM formulates recognition categories of input patterns and associates each category with a prediction. The knowledge that fuzzy ARAM discovers is compatible with IF-THEN rule-based representation. This enables the system architecture to be readily translated into a compact set of rules.

Two data sets, namely the ALL/AML data set (Golub et al., 1999) and the colon tumor data set (Alon et al., 1999), were used in our experiments. Identifying acute lymphoblastic leukemia (ALL) cases from acute myeloid leukemia (AML) cases is critical for the successful treatment of leukemia disease. Likewise, improvements in colon tumor classification have been central to advances in cancer treatment. One unique challenge of analyzing these gene expression data is the high feature dimensionality coupled with the small number of data samples. We illustrate fuzzy ARAM's predictive performance using features selected by two feature extraction methods, one statistical based (Furey et al., 2000) and the other entropy based (Fayyad & Irani, 1993). Our experiments show that fuzzy ARAM produces predictive performance comparable, if not superior, to those of all prior systems. More importantly, the rules extracted from fuzzy ARAM can be interpreted readily and used in disease understanding.

The rest of the paper is organized as follows. Section 2 presents two feature selection algorithms for reducing the dimensionality of the gene feature spaces. Section 3 presents the learning and prediction algorithms of the predictive model based on fuzzy ARAM. Section 4 illustrates how knowledge in the form of IF-THEN rules can be extracted from the predictive model. Section 5 reports our classification results and the knowledge extracted from the AML/ALL and the colon tumor data sets. The final section concludes and provides a discussion of our findings.

2. Dimensionality reduction

The first stage of our knowledge discovery process involves dimensionality reduction, in which the dimensionality of the gene expression data is reduced to a manageable number. We illustrate how predictive neural networks can work with two very distinct feature selection algorithms. The first algorithm, that computes a variant of the Fisher criterion scores, has been used by many statisticians and biologists. The Fisher method (Bishop, 1995) evaluates and selects each feature based on its own merits and preserves continuous gene expression values. The other algorithm, known as Entropy-based discretization (Fayyad & Irani, 1993), was proposed by computer scientists in the field of data mining. It employs a greedy method to select gene features, one at a time, which separate patterns into partitions with the minimum level of entropy. Both algorithms have been used in many prior experiments in extracting key features from gene expression data, including the two data sets that we investigate. Adopting the two algorithms enable us to compare the performance of the predictive neural networks in a more equal standing with those of alternative machine learning systems.

2.1. Fisher feature selection

The feature selection method based on a variant of the Fisher criterion (Furey et al., 2000; Golub et al., 1999) is summarized as follows. Consider a data set S with m expression vectors $\mathbf{x}^i = (x_1^i, \ldots, x_n^i)$, $1 \leq i \leq m$ where m is

the number of samples and n is the number of gene expression readings. Each sample is labelled with a class $Y \in \{+1, -1\}$ (eg. cancer vs. normal, AML vs. ALL). For each feature j, the Fisher score is calculated by

$$F(x_j) = \left| \frac{\mu_j^+ - \mu_j^-}{\rho_j^+ + \rho_j^-} \right| \quad (1)$$

where μ_j^+ and μ_j^- are the means and ρ_j^+ and ρ_j^- are the standard deviations of the feature values in the positive and negative classes respectively. The formula gives an advantage to genes with densely distributed and distinct expression levels on average in the two classes. The rationale is that such genes tend to have a higher discriminative power in classifying the samples into the two classes. The Fisher method is used merely as a criterion for selecting features. The gene feature values have to be normalized before presenting to our predictive modelling system for learning.

2.2. Entropy-based discretization

The entropy-based discretization method (Fayyad & Irani, 1993) couples an entropy based splitting criterion as used in the C4.5 decision tree (Quinlan, 1993) and a minimum description length stopping criterion. Working in a recursive manner, the method determines an optimal cutting point for each feature dimension to maximize the separation of the classes. Features that have no cutting points are deemed as not important and can be discarded. Suppose a cutting point T for a feature A partitions the set S of examples into two subsets S_1 and S_2 and there are k classes $C_1, \ldots, C_k$. The class entropy of a subset S_j, $j = 1,2$ is defined by

$$\mathrm{Ent}(S_j) = -\sum_{i=1}^{k} P(C_i, S_j) \log(P(C_i, S_j)) \quad (2)$$

where $P(C_i,S_j)$ is the proportion of examples in S_j that have class C_i. The class information entropy of the partition $E(A,T;S)$ is then given by

$$E(A, T; S) = \frac{|S_1|}{|S|} \mathrm{Ent}(S_1) + \frac{|S_2|}{|S|} \mathrm{Ent}(S_2). \quad (3)$$

A binary discretization for A is determined by selecting the cutting point T_A for which $E(A,T;S)$ is minimal amongst all the candidate cutting points. The same process can then be applied recursively to S_1 and S_2 until the stated stopping condition is satisfied. Whereas the Fisher method does not modify the feature values, the entropy-based method involves the identification of cutting points along each feature dimension and the conversion of continuous gene activity values into discrete features.

3. Predictive modelling

For building the predictive model, each training sample of a data set is first converted into a feature vector **A** and a class vector **B**. The feature vectors are derived based on the features selected by either the Fisher criterion or the entropy-based discretization method described in the previous section. For competitive learning systems, such as Adaptive Resonance Associative Map (ARAM), it is typically assumed that the feature values are bounded between 0 and 1. In other words, we have $0 \leq A_i \leq 1$ for each feature i.

Using the Fisher feature selection method, the real-valued expression reading of a feature i in a pattern sample x is normalized by

$$a_i = \frac{x_i - \min_p(x_i^p)}{\max_p(x_i^p) - \min_p(x_i^p)} \quad (4)$$

where $\min_p(x_i^p)$ and $\max_p(x_i^p)$ denote the minimum and maximum values of the feature i across all patterns p. To prevent the code proliferation problem (Carpenter et al., 1992), *complement coding* is applied to preserve the magnitude of the feature vectors. Specifically, the normalized feature vector **a** is augmented with a complement vector $\mathbf{a}^c$ to form the complement coded input vector $\mathbf{A} = (\mathbf{a}, \mathbf{a}^c)$, where $a_i^c = 1 - a_i$ for each feature i. Given a total of N features selected by the Fisher method, we derive an input vector **A** for each gene sample with a feature dimension of $2N$ and a norm ($|\mathbf{A}|$) of N.

Alternatively, if the entropy-based discretization method is used, the cutting point of each feature dimension creates a pair of binary features. For example, if the cutting point of the gene feature *Zyxin* is 994, a pair of binary features Zyxin $\geq$ 994 and Zyxin $<$ 994 will be included in the feature vector. Therefore, if a total of N gene features with cutting points are selected, we derive a binary input vector **A** for each gene sample with a feature dimension of $2N$. Complement coding is not needed here as the binary feature vectors already have a uniform norm of N.

The class vectors are typically binary (on-off) representation of the pattern classes or diagnostic categories of interests. Although the problems we investigate in this paper consist of only two classes, our predictive model, namely fuzzy ARAM, is capable of learning multidimensional mappings involving multiple pattern classes.

An ARAM system consists of an input field F_1^a, an output field F_1^b, and a category field F_2 (Fig. 2). Given a set of feature vectors presented at F_1^a with their corresponding class vectors presented at F_1^b, ARAM learns a predictive model (encoded by the recognition nodes in F_2) that associates combinations of key features to their respective classes. A simplified version of the fuzzy ARAM learning and prediction algorithms (Tan, 1995) is summarized as follows. The ARAM software is available at http://www.ntu.edu.sg/home/asahtan/downloads.

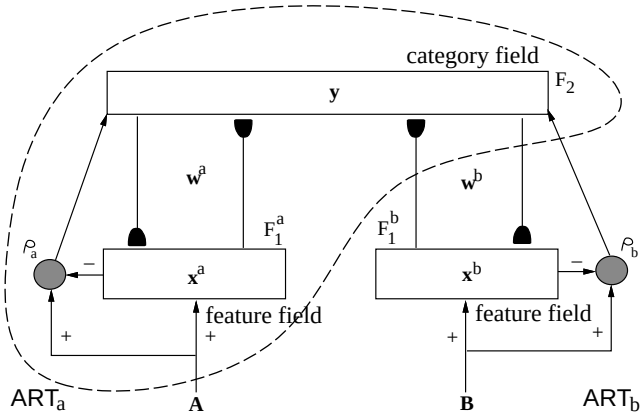

Fig. 2. The Adaptive Resonance Associative Map architecture.

Weight vectors. Each F_2 node j is associated with two adaptive weight templates $\mathbf{w}_j^a$ and $\mathbf{w}_j^b$. A F_2 node is said to be *uncommitted* if its weight templates have not encoded any input patterns. In fuzzy ARAM, the weight values of an uncommitted node are initialized to 1's. At the beginning of learning, there is no committed node and the F_2 field contains only one uncommitted node.

Parameters. ARAM dynamics are determined by the choice parameters $\alpha > 0$; the learning rates $\beta_a \in [0,1]$ and $\beta_b \in [0,1]$; and the vigilance parameters $\rho_a \in [0,1]$ and $\rho_b \in [0,1]$.

3.1. Learning

Learning in fuzzy ARAM consists of four key steps, namely bottom-up propagation, code competition and selection, top-down priming, and template learning, described as follows.

Bottom-up propagation. A bottom-up propagation process first takes place in which the activities (known as choice function values) of the nodes in the F_2 field are computed (Fig. 3). Specifically, given a feature vector $\mathbf{A}$, for each F_2 node j, ARAM computes a *choice function*

$$T_j = \frac{|\mathbf{A} \wedge \mathbf{w}_j^a|}{\alpha + |\mathbf{w}_j^a|} \tag{5}$$

where the fuzzy AND operation $\wedge$ is defined by

$$(\mathbf{p} \wedge \mathbf{q})_i \equiv \min(p_i, q_i), \tag{6}$$

and the norm $|\cdot|$ is defined by

$$|\mathbf{p}| \equiv \sum_i p_i \tag{7}$$

for vectors $\mathbf{p}$ and $\mathbf{q}$. In essence, the choice function T_j computes the match of the feature vector $\mathbf{A}$ with the ART_a weight vector $\mathbf{w}_j^a$ of the F_2 node j with respect to the norm of the weight vector.

Code competition and selection. A code competition process follows under which the F_2 node with the highest choice function value is identified. The process thus identifies the F_2 node that encodes an ART_a weight template $\mathbf{w}_j^a$ closest to the feature vector $\mathbf{A}$. The system is said to make a choice when at most one F_2 node can become active after the code competition process. The winner is indexed at J where

$$T_J^c = \max\{T_j^c : \text{for all } F_2^c \text{ node } j\}. \tag{8}$$

Top-down priming. Before node J can be used for learning, a template matching process checks that its weight templates are sufficiently close to their respective feature and class vectors (Fig. 4). Specifically, *resonance* occurs if the *match functions* (m_J^a and m_J^b) meet the vigilance criteria in their respective modules:

$$m_J^a = \frac{|\mathbf{A} \wedge \mathbf{w}_J^a|}{|\mathbf{A}|} \geq \rho_a \text{ and } m_J^b = \frac{|\mathbf{B} \wedge \mathbf{w}_J^b|}{|\mathbf{B}|} \geq \rho_b. \tag{9}$$

Whereas the choice function computes the similarity between the input and weight template vectors with respect to the norm of the weight template vectors, the match function computes the similarity with respect to the norm of the input

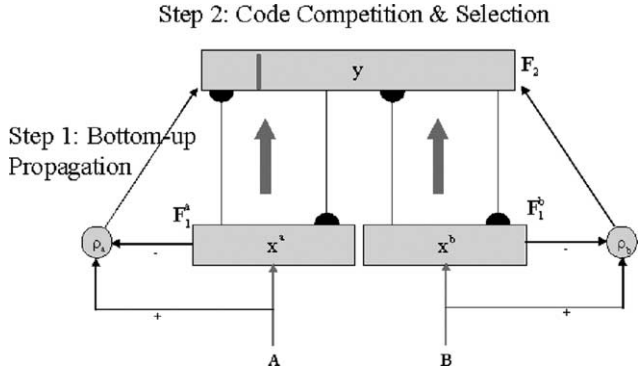

Fig. 3. During bottom-up propagation, a choice function value is computed for each F_2 node. The F_2 node with the highest choice function value is then selected.

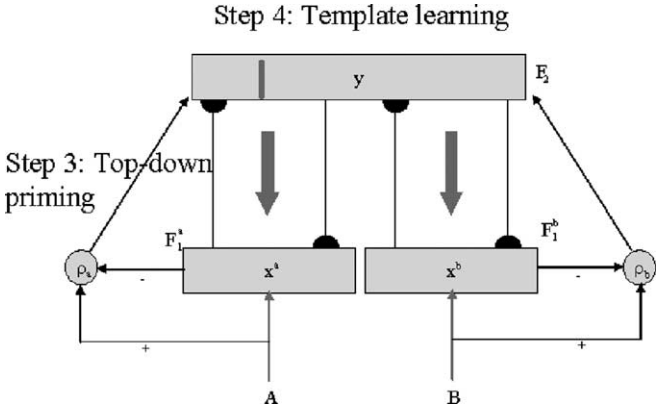

Fig. 4. During top-down priming, the match function values of the selected F_2 node are evaluated. If each match value satisfies the match criterion in the respective module, resonance occurs and template learning follows under which the selected node learns to encode the feature and class vectors.

feature and class vectors. In conjunction, the choice and match functions work cooperatively to achieve stable coding and maximize code compression.

Once *resonance* occurs, learning ensues, as defined below. If any of the vigilance constraints is violated, *mismatch reset* occurs in which the value of the choice function T_J^c is set to 0 for the duration of the input presentation. With a *match tracking* process, at the beginning of each input presentation, the vigilance parameter ρ_a equals a baseline vigilance $\bar{\rho}_a$. If a mismatch reset occurs, ρ_a is increased until it is slightly larger than the match function m_J^a. The search process then selects another F_2 node J under the revised vigilance criterion until a resonance is achieved. This search and test process is guaranteed to end as ARAM will either find a committed node that satisfies the vigilance criterion or activate an uncommitted node which would definitely satisfy the criterion due to its initial weight values of 1' s.

Template learning. Once the search ends, the weight vectors $\mathbf{w}_J^a$ and $\mathbf{w}_J^b$ of the chosen node J are updated according to

$$\mathbf{w}_J^{a(\text{new})} = (1-\beta_a)\mathbf{w}_J^{a(\text{old})} + \beta_a(\mathbf{A} \wedge \mathbf{w}_J^{a(\text{old})}) \tag{10}$$

and

$$\mathbf{w}_J^{b(\text{new})} = (1-\beta_b)\mathbf{w}_J^{b(\text{old})} + \beta_b(\mathbf{B} \wedge \mathbf{w}_J^{b(\text{old})}) \tag{11}$$

respectively. The learning rule adjusts the weight vectors towards the fuzzy AND of their original weight vectors and the respective input feature and class vectors. The rationale is to learn by encoding the common attribute values of the input vectors and the weight vectors. For an *uncommitted* node J, the learning rates β_a and β_b are typically set to 1. For *committed* nodes, the learning rates can remain as 1 for fast learning or below 1 for slow learning in noisy environment. When an uncommitted F_2 node is selected for learning a pattern, it becomes *committed* immediately and a new *uncommitted* node is added to the F_2 field. The network thus creates a dynamic number of F_2 nodes in response to the incoming patterns. Quick commitment is a key characteristic of predictive self-organizing neural networks as part of the real time online learning. Despite that the learning is instantaneous, it is also stable due to the top down priming mechanism.

3.2. Prediction

During prediction, only the feature vector is presented. The system is supposed to predict the class vector based on its learned knowledge. In ARAM systems with category choice, only the F_2 node J that receives maximal $F_1^a \to F_2$ input T_j predicts ART_b output. Typically, the activity value of a F_2 node j is given by

$$y_j = \begin{cases} 1 & \text{if } j = J \text{ where } T_J > T_k \text{ for all } k \neq J \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

To cater for tasks where probabilistic likelihood prediction scores are desired, a new variant of choice is proposed here to preserve the activation value of the chosen F_2 node J after code competition. In other words,

$$y_j = \begin{cases} T_J & \text{if } j = J \text{ where } T_J > T_k \text{ for all } k \neq J \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

The output class vector $\mathbf{C}$ is then computed by

$$\mathbf{C} = \mathbf{w}_J^b y_J \tag{14}$$

where C_i indicates the estimated likelihood of the input feature vector belonging to class i.

4. Rule extraction

In an ARAM network, each node in the F_2 field learns to encode a group of input patterns and associate them with an output prediction. Learned weight vectors, one for each F_2 node, constitute a set of rules that link antecedents to consequences. Specifically, given a committed F_2 node j with the weight template vectors $\mathbf{w}_j^a$ and $\mathbf{w}_j^b$, we derive an IF-THEN rule of the form

$$\mathcal{C} : -\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \tag{15}$$

where $\mathcal{C}$ is the class indicated by the (typically only one) non-zero attribute value in $\mathbf{w}_j^b$ and $\mathcal{A}_1,\mathcal{A}_2,\ldots,\mathcal{A}_n$ are the antecedents or conditions corresponding to the non-zero feature values in $\mathbf{w}_j^a$. For analyzing gene expression data, $\mathcal{C}$ typically corresponds to an outcome or a class of the diagnosis (such as tumor or normal cells), whereas $\mathcal{A}_1,\mathcal{A}_2,\ldots,\mathcal{A}_n$ denote the conditions of the expression levels at the gene sites $g_1,g_2,\ldots,g_n$ respectively.

Using the Fisher feature selection method, a pair of complement coded weight values $(w_{ji}^a, \bar{w}_{ji}^a)$ for a feature i translates into a value range of $[w_{ji}^a, 1-\bar{w}_{ji}^a]$. For example, a pair of weight values (0.7, 0.0) for feature i indicates a value range of [0.7,1.0], i.e. the normalized feature value $a_i \geq 0.7$. For more details, please refer to Carpenter et al. (1992) for a discussion of complement coded weight values. The value ranges obtained may subsequently be mapped back to the original scale of the expression values for human interpretation. For example, the normalized feature value range of $a_i \geq 0.7$ may correspond to the gene expression range of $x_i \geq 334$ in absolute terms.

Using the entropy-based discretization method, a pair of weight values for a feature i indicates the truth values of the conditions $x_i \geq c_i$ and $x_i < c_i$ respectively, where c_i is the cutting point for the feature i. A weight value pair of (1,0) indicates that the condition $x_i \geq c_i$ is true. A weight value pair of (0,1) indicates that the condition $x_i < c_i$ is true. For example, a weight value pair of (0,1) for the feature Zyxin with a cutting point at 994 translates into the condition of $Zyxin < 994$. A weight value pair of (0,0) indicates that both conditions are not relevant and can be omitted from the rule.

An ARAM rule can be interpreted individually but most often functions as a member of a rule ensemble. During prediction, ARAM rules compete in accordance with the category choice process (Eq. (8)). Similar to typical conjunctive rules, an ARAM rule is activated when all of its conditions are satisfied. However, to maximize generalization, ARAM rules typically operate in a fuzzy and nearest match manner. In other words, a rule can be activated as long as a sufficient number of its conditions are satisfied and it is the closest match for the given input. Given two rules with all of their antecedents satisfied, the choice function gives an advantage to the rule with a larger number of antecedents. For example, consider a rule set consisting of the two following rules,

$$\text{(Rule 1) } \mathcal{C}_1 : -\mathcal{A}_1 \tag{16}$$

and

$$\text{(Rule 2) } \mathcal{C}_1 : -\mathcal{A}_1, \mathcal{A}_2. \tag{17}$$

If both $\mathcal{A}_1$ and $\mathcal{A}_2$ are satisfied, *Rule 2* will have a choice function value of $2/(\alpha+2)$ which is higher than $1/(\alpha+1)$ of Rule 1. It follows that Rule 2 will be chosen over Rule 1. In fact, *Rule 2* can be viewed as an exception rule for *Rule 1*. When $\mathcal{A}_2$ is not present, *Rule 1* is used to predict C_1 given $\mathcal{A}_1$. When both $\mathcal{A}_1$ and $\mathcal{A}_2$ are present, *Rule 2* kicks in to make the decision.

To reduce the complexity of ARAM rules, a rule pruning procedure (Carpenter & Tan, 1995) is used here to select a concise set of rules from trained ARAM networks based on their confidence factors. For large data sets, the rule pruning algorithm derives a confidence factor for each F_2 node in terms of its usage frequency in a *training* set and its predictive accuracy on a *predicting* set. For small data sets, we compute confidence factors solely based on *usage* in the training set. The confidence factor identifies good rules with nodes that are frequently and/or correctly used.

Specifically, the pruning algorithm evaluates each F_2 node j in terms of a confidence factor CF_j:

$$CF_j = \gamma \text{Usage}_j + (1-\gamma)\text{Accuracy}_j, \tag{18}$$

where Usage_j is the usage of node j, Accuracy_j is its accuracy, and $\gamma \in [0,1]$ is a weighting factor.

For a F_2 node j that predicts class c, its usage equals the fraction of the training set patterns of class c encoded by the node j (S_j), divided by the maximum fraction of training patterns of class c encoded by any node J (S_J):

$$\text{Usage}_j = S_j / \max\{S_J : \text{node } J \text{ predicts class } c\}. \tag{19}$$

As usage is normalized across nodes with the same class, for each class c, there is at least one node predicting class c with a usage value of 1.

For a F_2 category j that predicts class c, its accuracy equals the percent of the predicting set patterns predicted correctly by node j (P_j), divided by the maximum percent of patterns predicted correctly by any node J (P_J) that predicts class c:

$$\text{Accuracy}_j = P_j / \max\{P_J : \text{node } J \text{ predicts class } c\}. \tag{20}$$

As accuracy is also normalized across nodes predicting the same class, for each class c, there is always at least one F_2 node (or rule) with an accuracy of 1.

After confidence factors are determined, F_2 nodes can be pruned from the network using one of following strategies:

Threshold pruning. This is the simplest type of pruning where the F_2 nodes with confidence factors below a given threshold τ are removed from the network. A typical setting for τ is 0.1 for small data sets. This method is fast and provides an initial elimination of unwanted nodes. To avoid over-pruning, it is sometimes useful to specify a minimum number of recognition categories to be preserved in the system.

Local pruning. Local pruning removes recognition categories one at a time from an ARAM network. The baseline system performance on the training and the predicting sets is first determined. Then the algorithm deletes the recognition category with the lowest confidence factor. The category is replaced, however, if its removal degrades system performance on the training and predicting sets.

A variant of the local pruning strategy updates baseline performance each time a category is removed. This option, called *hill-climbing*, gives slightly larger rule sets but better predictive accuracy. A hybrid strategy first prunes the ARAM systems using threshold pruning and then applies local pruning on the remaining smaller set of rules.

5. Experiments

5.1. The AML/ALL data set

The ALL/AML data set (Golub et al., 1999), available at http://www-genome.wi.mit.edu/cgi-bin/cancer, is provided for the classification of acute leukemia cases into those arising from lymphoid precursors (acute lymphoblastic leukemia, ALL) or from myeloid precursors (acute myeloid leukemia, AML). The training set consists of 38 bone marrow samples (27 ALL and 11 AML cases) over 7129 probes from 6817 human genes. In addition, 34 testing samples are provided, with 20 ALL and 14 AML cases. In order to compare with previously published results, we used this original data partition for our experiments.

To determine an appropriate feature set, we performed leave-one-out cross validation on the training set based on a varying number of features selected based on the Fisher selection criterion. Each experiment was repeated for 10 times for statistical stability. In all experiments, ARAM model used a standard set of parameter values: choice

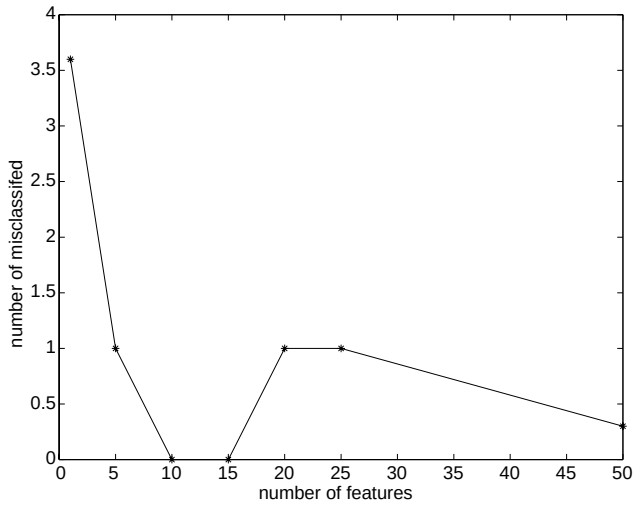

Fig. 5. The cross validation performance of fuzzy ARAM on the AML/ALL training data using a varying number of features.

parameter $\alpha = 1.0$; learning rates $\beta_a = \beta_b = 1.0$ for fast learning; and baseline vigilance parameter $\bar{\rho}_a = 0.0$. Soft category choice is used in the F_2 layer to provide probabilistic prediction scores.

Our experiment results indicated that the Fisher feature selection method was effective in deriving small gene sets with good prediction accuracy. Specifically, perfect cross validation result on the training set was achieved by using the top 10 and top 15 features (Fig. 5). Based on the top 10 genes, we trained ARAM model using the training set and evaluated its performance on the test data. Among the 34 test samples, an average of 3.4 samples (typically belonging to the 57, 60 and 66th patients) were misclassified over 10 experiments using different input presentation orders. The three misclassified samples were among the five common misclassifications reported previously (Gloub et al., 1999).

We repeated the experiments using the entropy-based feature selection and discretization method. Only 866 of the 7129 genes in the training data were partitioned into two or three intervals, while there were no cutting points for the rest of the features. We examined the 866 genes and sorted them by increasing order of entropy values. Applying leave-one-out cross validation using the same ARAM parameter setting, perfect predication on training set was readily achieved by using the top one, top five, and top 10 genes. Based on just the top one gene, only three samples in the test set were misclassified. This result was similar to that obtained using Emerging Patterns (Li & Wong, 2002).

Among the top 10 genes (Table 1), we found that *Zyxin*, *FAH Fumarylacetoacetate*, and *CST3 Cystatin C* were among the biologically instructive genes identified earlier (Golub et al., 1999). Specifically, *Zyxin* was reported to encode a LIM domain protein important in cell adhesion in fibroblasts. Note that only three out of the 10 genes were among the top 10 genes (Table 2) picked up based on the Fisher scores. Whereas the Fisher criterion evaluates each gene individually, the entropy-based method selects the features one after another. This strategy helps to create compact gene combinations in which the features complement each other. Based on the top 10 genes selected by entropy-based discretization, the number of misclassified test samples decreased to two (typically, the 66 and 67th patients). The predictive performance of ARAM, compared with those obtained by SVM (Furey et al., 2000), Weighted Voting (Golub et al., 1999), and Emerging Patterns (Li & Wong, 2002), are summarized in Table 3.

Ben-Dor et al. have also conducted experiments on the AML/ALL data set using a myriad of methods, including Nearest Neighbor, Support Vector Machines, and AdaBoost algorithm. Their results however, were based on a leave-one-out benchmark paradigm on the entire set of

Table 1
The top 10 genes selected for the AML/ALL data set through entropy-based discretization

Gene	Entropy	Cutting point	Description
X95735	0.0000	994.0	Zyxin
M55150	0.0393	1346.0	FAH Fumarylacetoacetate
M31166	0.0493	83.5	PTX3 Pentaxin-related gene, rapidly induced by IL-1 beta (PTX3)
M27891	0.0493	1419.5	CST3 Cystatin C (amyloid angiopathy and cerebral hemorrhage)
X70297	0.0638	339.0	CHRNA7 Cholinergic receptor, nicotinic, alpha polypeptide 7
P31483	0.0638	80.5	Nucleolysin TIA-1
L09209	0.0638	992.5	APLP2 Amyloid beta (A4) precursor-like protein 2
U46499	0.0638	156.5	Glutathione S-transferase, microsomal
M16038	0.0831	651.5	LYN V-yes-1 Yamaguchi sarcoma viral related oncogene homolog
M92287	0.0831	1869.5	CCND3 Cyclin D3

Table 2
The top 10 genes selected for the AML/ALL data set based on Fisher scores

Gene	Fisher score	Description
M55150	1.518	FAH Fumarylacetoacetate
U50136	1.479	Leukotriene C4 synthase (LTC4S) gene
X95735	1.465	Zyxin
U22376	1.371	C-myb gene extracted from Human (c-myb) gene, complete primary cds, and five complete alternatively spliced cds
M16038	1.254	LYN V-yes-1 Yamaguchi sarcoma viral related oncogene Homolog
M23197	1.248	CD33 CD33 antigen (differentiation antigen)
M84526	1.247	DF D component of complement (adipsin)
P48357	1.232	LEPR Leptin receptor
P31269	1.205	GB DEF, Homeodomain protein HoxA9 mRNA
D49950	1.183	Liver mRNA for interferon-gamma inducing factor(IGIF)

Table 3
The classification performance of ARAM on the AML/ALL data set based on the 34 test samples, compared with SVM, Weighted Voting, and Emerging Patterns (EP)

Method	Number of features	Number of misclassifications
SVM	25–1000	2–4
Weighted Voting	50	5
EP (Entropy)	1	3
vARAM (Fisher)	10	3.4
vARAM (Entropy)	10	2.0

Table 4
A sample set of two ARAM rules based on features derived by entropy-based discretization that correctly classifies all 72 samples in the ALL/AML data set

AML	Glutathione s-transferase, Microsomal $>$ 156.5
ALL	Syxin $<$ 994.0 and CST $<$ 383.5

the 72 samples. Also using a feature selection method, the best result of 1 classification error was obtained by SVM with a quadratic kernel. To compare with their results, we repeated ARAM evaluation using the leave-one-out cross validation and found only 1 error out of the 72 experiments. This is equivalent to the best results reported by Ben-Dor et al. To put this level of performance into perspectives, the recent leave-one-out experiments conducted by Zhang et al. on the AML/ALL data set, that involved growing decision trees by recursive partitioning and combining them into forests (Zhang, Yu, & Burton, 2003), produced 3–4 errors using deterministic forests and 9–10 errors using single trees.

Although Ben-Dor et al. and other prior studies have also made use of feature selection, their systems can only identify individual 'informative' genes that have high predictive power for the various cancer classes. In our experiments, we were able to go further to derive 'informative combinations' of genes with an *AND* relationship. Table 4 illustrates a sample set of two ARAM rules extracted that classifies correctly all 72 cases of the AML/ALL data set. In this case, the combination of *Zyxin* and *CST3 Cystatin C* has proven to be a very reliable predictor for ALL cases.

5.2. The colon tumor data set

The colon tumor data set (Alon et al., 1999) contains 40 tumor and 22 normal colon tissue samples. The data are processed with an Affymetrix oligonucleotide array complementary to more than 6500 human genes. Of these genes, the 2000 with the highest minimal intensity across the tissues are selected for classification purpose. These scores are publicly available via http://microarray.princeton.edu/oncology/affydata.

Based on the Fisher criterion, the top 1,5,10,…,500 genes with the highest $F(x_j)$ score were chosen from the data set. To enable comparison with prior results, we performed leave-one-out cross validation directly on the entire set of

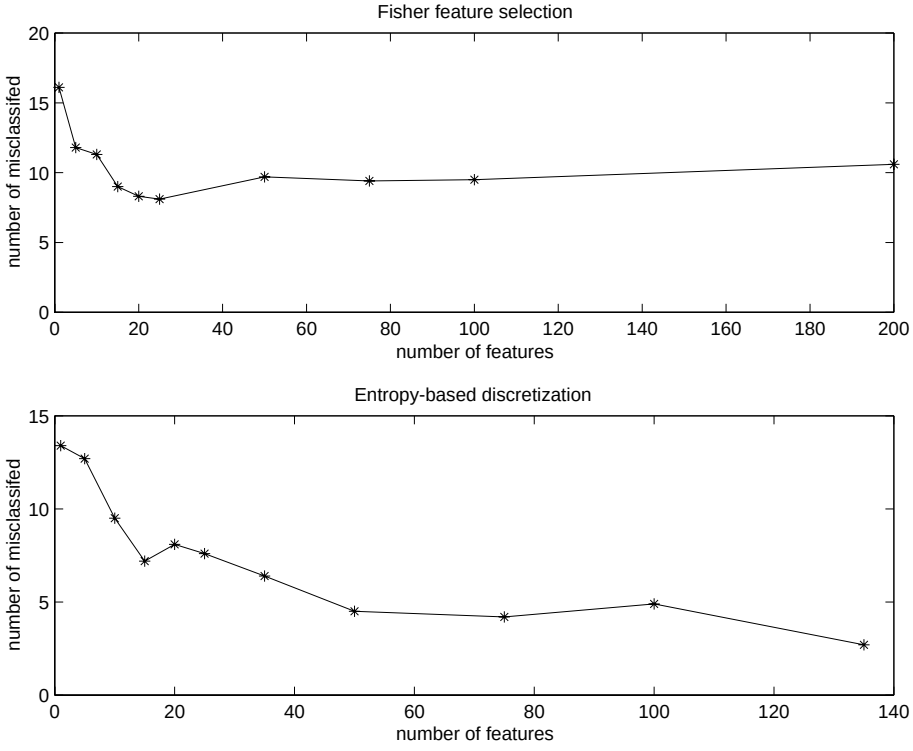

Fig. 6. The classification performance of fuzzy ARAM on the colon tumor data set using a varying number of features selected by Fisher (top) and entropy-based discretization (bottom).

Table 5
A sample set of two ARAM rules that correctly classifies 60 out of 62 samples in the colon tumor data set

Normal	X61118 < 189.2
tumor	M26383 ≥ 59.8 and M76378 < 842.3 and D14812 ≥ 155.5 and K03460 ≥ 123.6

Table 6
The classification performance of fuzzy ARAM on the colon tumor data set compared with SVM, clustering, and Emerging Patterns (EP)

Method	Number of features	Number of misclassifications
SVM	2000	6
Clustering	2000	8
EP	35	5
vARAM (Fisher)	25	8.1
vARAM (Entropy)	135	2.4

62 samples available. As shown in Fig. 6 (top), at least seven samples were misclassified no matter how many genes were used. This performance was similar to those reported using SVM and clustering. Using 25 features, the system produced an average of 8.1 misclassifications across 10 runs of leave-one-out cross validation.

Based on the discretization method, only 135 of the 2000 genes were partitioned into two intervals. We sorted the 135 genes according to increasing entropy values and conducted leave-one-out cross validation using a varying number of features from 1 to 135. As shown in Fig. 6 (bottom), the number of misclassifications decreased to around four with 50 or more features. The best result, an average of 2.4 misclassifications, was obtained using all 135 features over 10 runs of leave-one-out cross validation. The most common misclassification samples were T2 and T33,[1] one of which was among the six misclassified samples (T30, T33, T36, N8, N34, N36) previously reported (Alon et al., 1999; Furey et al., 2000). Based on the 135 genes, a set of rules (Table 5) were extracted which collectively misclassified only two samples (N39 and N40) in the entire data set. The predictive performance of ARAM for the colon tumor data set, compared with those obtained by SVM (Ben-Dor et al., 2000; Furey et al., 2000), clustering (Alon et al., 1999), and Emerging Patterns (Li & Wong, 2002) are summarized in Table 6.

6. Discussion

We have presented a systematic approach for learning and extracting knowledge from gene expression data based on a class of predictive self-organizing network models. Experiments based on the two gene data sets showed that fuzzy ARAM was able to produce interpretable rules with very high predictive power. The use of the feature selection methods enables us to reduce the number of features drastically before presenting the feature vectors for learning by the predictive neural networks. The effectiveness of the feature selection methods have been supported by our empirical experimental results. Specifically, the leave-one-out cross validation conducted on the AML/ALL data set produced perfect accuracy using just the top 10 and 15 features selected by the Fisher method. Whereas Furey et al. found that dimensionality reduction did not significantly improve the SVM's classification performance, our experiments showed that feature selection played an important role in deriving good prediction performance and concise rules for ARAM. While entropy-based discretization appears to outperform Fisher feature selection, we reckon that, for some problems, it may still be necessary to preserve real-valued features for the predictive modelling stage. It is thus advantageous that a predictive system can work with both discretized as well as continuous-valued features.

Even after feature selection, sometimes we still need to deal with a large number of features. The best prediction results for the colon tumor data set, for example, were achieved by using all 135 features provided by the entropy-based discretization method. Compared with slow learning, iterative optimizing, and search-based methods, the ARAM learning and rule extraction approach is extremely efficient. As an illustration, a complete set of leave-one-out cross validation experiments for all 62 samples of the colon tumor data set using 135 features took just one second on a SUN Ultra-10 machine.

The IF-THEN rules extracted from our system are similar in form to those produced by C4.5 decision tree system (Quinlan, 1993). However, ARAM rules and C4.5 rules function very differently. A C4.5 rule operates in isolation. A conclusion/prediction is made by a single rule as long as all of its conditions are satisfied. On the other hand, ARAM rules operate as an ensemble governed by a fuzzy choice principle under which each rule produces a real-valued choice function score and competes with each other to make a prediction. Although it may seem easier to interpret the 'precise' rules as in typical decision tree systems, it is in fact quite unnatural to make hard decisions by imposing exact boundaries on the gene expression values. The ARAM fuzzy choice function enables a rule to be partially activated even when not all of its conditions are satisfied. For real-valued features, the choice function enables a rule to be fully activated when the inputs fall into the specified ranges and partially activated with a degree of confidence that decreases gradually as the inputs deviate away from the specified ranges. The fuzzy choice function and the winner-take-all rule competition paradigm provide the nonlinearity necessary for modelling the gene expression complexity and serve to maximize generalization.

Our experimental results have been based on the two well-known and publicly available data sets that allow us to

[1] T, represents tumor tissue and N, represents normal tissue.

make comparison with the numerous results obtained by a wide range of the state-of-the-art methods. In our future work, we hope to work with much larger and complex data sets when they become available. As we have adopted a systematic approach and used a standard set of ARAM parameter values throughout all experiments, we expect to carry over the good performance to bigger and more challenging problems.

Predictive accuracy aside, we contend that the key strength of our approach lies in its ability to generate interpretable knowledge in an efficient manner. Having a systematic approach to extract interpretable rules, our next step would be to work with biologists and medical experts and refer to the rich medical literatures for interpreting and validating the knowledge discovered by the system. This will form the core of our future work.

Acknowledgements

The authors would like to thank Jinyan Li and Huiqing Liu for sharing their experience in applying feature selection methods to the two gene expression data sets. Acknowledgement also goes to the anonymous reviewers for their valuable comments and suggestions.

References

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., et al. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceeding of the National Academy of Science*, *96*, 6745–6750.

Ben-Dor, A., Bruhn, L., Friedman, N., Nachman, I., Schummer, M., & Yakhini, Z. (2000). Tissue classification with gene expression profiles. *Journal of Computational Biology*, *7*(3–4), 559–583.

Bishop, C. (1995). *Neural networks for pattern recognition*. Oxford, UK: Oxford UP.

Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., & Rosen, D. B. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, *3*, 698–713.

Carpenter, G. A., Grossberg, S., & Reynolds, J. H. (1991). ARTMAP: Supervised real time learning and classification by a self-organizing neural network. *Neural Networks*, *4*, 565–588.

Carpenter, G. A., & Ross, W. D. (1993). *ART-EMAP: A neural network architecture for object recognition by evidence accumulation network World Congress on Neural Networks, Portland, OR*, Vol. III. Hillsdale, NJ: Lawrence Erlbaum pp. 649–656.

Carpenter, G. A., & Tan, A. H. (1995). Rule extraction: From neural architecture to symbolic representation. *Connection Science*, *7*(1), 3–27.

Fayyad, U., & Irani, K. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the 12th International Conference on Machine Learing*, (pp. 1022–1029). San Francisco: Morgan Kaufmann.

Furey, T. S., Cristianiini, 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. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, *286*, 531–537.

He, J., Tan, A.-H., & Tan, C.-L. (2003). On machine learning methods for Chinese documents classification. *Applied intelligence: Special issue on text and web mining*, *18*(3), 311–322.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features, In *Proceedings of the 10th European Conference on Machine Learning (ECML'98)*, (pp. 137–142). Berlin: Springer.

Li, J., & Wong, L. (2002). Identifying good diagnostic gene groups from gene expression profiles using the concept of emerging patterns. *Bioinformatics*, *18*(5), 725–734.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann.

Rumelhart, D.E., Hinton, G., & Williams, R. (1986) Learning internal representations by error propagation. In *Parallel distributed processing: Explorations in the microstructures of cognitions* (pp. 318–362). Cambridge, MA: MIT Press.

Slonim, D. K., Tamayo, P., Mesirov, J. P., Golub, T. R., & Lander, E. S. (2000) Class prediction and discovery using gene expression data. In *Proceedings of the Fourth Annual Conference on Computational Molecular Biology (RECOMB)* (pp. 263–272). San Francisco: Morgan Kaufmann.

Tan, A.-H. (1995). Adaptive Resonance Associative Map. *Neural Networks*, *8*(3), 437–446.

Tan, A.-H. (1997). Cascade ARTMAP: Integrating neural computation and symboli knowledge processing. *IEEE Transactions on Neural Networks*, *8*(2), 237–250.

Tan, A. -H. (2001). Predictive self-organizing networks for text categorization. In *Proceedings of the Fifth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'01)* (pp. 66–77). Hong Kong.

Tan, A.-H., Ong, H.-L., Pan, H., Ng, J., & Li, Q.-X. (2004). Towards personalized web intelligence. *Knowledge and Information Systems*, *6*(5), 595–616.

Tan, A.-H., Soon, H.-S. (2000). Predictive Adaptive Resonance Theory and knowledge discovery in database. In *Proceedings of the Fourth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'00)* (pp. 173–176) Kyoto.

Williamson, J. R. (1996). Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps. *Neural Networks*, *9*(5), 881–897.

Zhang, H., Yu, C.-Y., & Burton, S. (2003). Cell and tumor classification using gene expression data: Construction of forests. *Proceedings, National Academy of Science*, *100*(7), 4168–4172.