Neurocomputing 179 (2016) 161–168

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Somatic mutation detection using ensemble of flexible neural tree model

Bin Yang [a,*], Yuehui Chen [b]

[a] School of Information Science and Engineering, Zaozhuang University, Zaozhuang 277160, PR China
[b] School of Information Science and Engineering, University of Jinan, Jinan 250022, PR China

ARTICLE INFO

Article history:
Received 3 July 2014
Received in revised form
23 May 2015
Accepted 1 December 2015
Communicated by A. Abraham
Available online 11 December 2015

Keywords:
Flexible neural tree model
Next-generation sequencing technology
Somatic mutations
Particle swarm optimization
Ensemble learning

ABSTRACT

The advances on next-generation sequencing technology (NGS) have enabled researchers to detect somatic mutations. Much effect has been devoted to improve accuracy of discovering somatic mutations from tumour/normal NGS data. In this study, flexible neural tree model (FNT) is proposed to detect somatic mutations in tumour-normal paired sequencing data. To improve the classification accuracy further, a new classification ensemble approach based on Radial Basis Function (RBF) neural networks as nonlinear combination function is proposed. The proposed method is applied to real biological dataset from exome capture data and the whole genome shotgun data. Results show that the obtained FNT model has a fewer number of variables with reduced number of input features and with significant improvement in the detection accuracy using the proposed ensemble learning method. Our method also selects 10 import features for somatic mutation detection, which could be used to analyze NGS mutations further.

1. Introduction

Next-generation sequencing (NGS) technologies such as Roche 454, Illumina GA and ABI SOLiD, are revolutionizing cancer genomics research [31,16,30]. Now the technology could offer a powerful and cost effective approach to detect the mutations including single nucleotide polymorphisms (SNPs), indels, chromosomal rearrangements and copy number variations, which occur in diseases [19]. Somatic mutations where the variant was found in the tumour but not the normal, have long been recognized as a feature of tumour cells [20]. Somatic mutations are commonly carried in genomes of all cancer cells [37]. It is believed that tumour is derived from a single somatic cell that has accumulated multiple DNA mutations [32]. A subset of somatic mutations expected to initiate and maintain tumor growth, known as driver mutations, conferred selective growth advantage, are implicated in cancer development, whereas the remainder are passengers, and play no role in transforming normal cells into cancerous cells, fully understanding the mechanisms of the disease and the design of more effective treatments [33,28,17].

In disease study, NGS mutation identification algorithms, including Samtools [25], GATK [14], SOAPsnp [26], and bam2mpg [38], could align sequence reads from paired tumour/normal samples to reference human genome, and predict lots of variants. The problem is how to identify accurate somatic mutations from tens of millions of candidate ones. Moreover, error sources derived from experiment samples, sequencing platforms, aligned algorithm, uncertainties in read alignments and so on, lead to identifying somatic mutations more difficult. In recent years, a lot of computational methods have been proposed to identify somatic mutations from paired tumour/normal samples. In general these methods are classified into two categories. The first one is to detect somatic mutations according to the definition and parameters from mutation identification algorithms. Researchers often select those variants appearing in the tumour, but not in the normal sample. Then according to the parameters of each variation, such as sequencing depth, genotype quality, consensus quality, allele/strand balance, heterozygous rate and number of mutation reads, they determine filtering thresholds and select arbitrarily mutations [22,27]. Jia et al. [22] highlighted four critical parameters that could enhance the accuracy of called NGS mutations: quality and deepness, refinement and improvement of initial mapping, allele/strand balance, and examination of spurious genes. But these methods may lead to existence of different standards for the different diseases, platforms and algorithms. Löwer et al. [27] proposed a robust algorithm that assigned a single statistic, a false discovery rate (FDR), to each somatic mutation identified by NGS. This FDR confidence value could discriminate true mutations from erroneous calls.

Supervised machine learning algorithms are applied to identify somatic mutations. Compared with above methods, this kind of approaches performs better with low false positive rate [15]. Somatic mutation detection problem can be formalized as binary classification

* Corresponding author.
E-mail addresses: batsi@126.com (B. Yang), yhchen@ujn.edu.cn (Y. Chen).

http://dx.doi.org/10.1016/j.neucom.2015.12.001
0925-2312/

problem. Somatic mutations are classed as positions, while germline variants which are found in the tumour and the normal or wild-types (no variants found in either the tumour or the normal) are classed as non-somatic positions. The features of each mutation are collected from the parameters of aligning reads from paired tumour/normal samples to reference genome using mutation caller algorithms. Shah et al. [15] presented the comparison of four supervised machine learning algorithms (random forest, Bayesian additive regression tree, support vector machine and logistic regression) for somatic mutation detection with 106 features in tumour/normal NGS experiments. Huntsman et al. used a Random Forest classifier trained on validated SNVs from triple negative breast cancer exome capture data to remove false-positive calls [29,36].

Neural networks have been widely applied in pattern recognition for the reason that neural-networks based classifiers can incorporate both statistical and structural information and achieve better performance than the simple minimum distance classifiers [5]. However neural networks structure is difficult to regulate, it suffers from slow convergence characteristics and over-fitting phenomenon leading the decline of its generalization, and it is prone to be trapped in local minima [9]. Chen [10] proposed the flexible neural tree (FNT), which was widely used to solve classification problems such as face recognition [11], intrusion detection [7], breast cancer [7,8,34], online hand gesture recognition [21]. Compared with neural networks, a FNT model has two advantages. (1) The model can be seen as a flexible multi-layer feedforward neural network with over-layer connections and free parameters in activation functions, so it is powerful and flexible model to model complex systems. (2) The model can select the proper input variables or time-lags for constructing a model automatically, so as to select the input variables or features automatically.

In this paper, to reduce the false positive rate of identifying somatic mutations, a flexible neural tree model is proposed for somatic mutations detection using tumour-normal paired sequencing data. The hierarchical structure of the FNT model is evolved using genetic programming (GP) like tree structure-based evolutionary algorithm with specific instructions. The fine tuning of the parameters encoded in the structure is accomplished using particle swarm optimization (PSO). The proposed method interleaves both optimizations. Starting with random structures and corresponding parameters, it first tries to improve the structure and then as soon as an improved structure is found, it fine tunes its parameters. It then goes back to improving the structure again and, fine tunes the structure and rules' parameters. This loop continues until a satisfactory solution is found or a time limit is reached.

In general, combining the outputs of several classification models could improve on the performance of the single one, which is based on a suitable decomposition of the classification error. Chen [8] proposed the flexible neural tree models and their ensemble models for the detection of breast cancer, and the best accuracy was offered by the generalized ensemble method followed by the basic ensemble method. Cai [3] proposed the ensemble learning model using evolving learning to optimize the classifier. Saha [35] combined multiple classifiers using vote based classifier ensemble technique for named entity recognition, and genetic algorithm based technique has been proposed for weighted vote based classifier ensemble selection. Cai [4] introduced the evolving learning model to optimize the data distribution for classifier. In order to improve the accuracy of somatic mutation detection further, a new classification ensemble approach based on Radial Basis Function (RBF) neural networks as nonlinear combination function is proposed. PSO is used to optimize the parameters of RBF neural networks.

The paper is organized as follows. Section 2 gives the materials and methods about the FNT models. Section 3 presents some experiments for identifying somatic mutations. Some concluding remarks are presented in Section 4.

2. Materials and methods

2.1. The FNT model

In this research, a tree-structural based encoding method with specific instruction set is selected for representing a FNT model [10,12].

2.1.1. Flexible neuron instructor

The used function set F and terminal instruction set T for generating a FNT model are described as follows:

$$S = F \bigcup T = \{+_2, +_3, \ldots, +_N\} \bigcup \{x_1, \ldots, x_n\}, \quad (1)$$

where $+_i (i = 2, 3, \ldots, N)$ denotes non-leaf nodes' instruction and taking i arguments. $x_1, x_2, \ldots, x_n$ are leaf nodes' instructions and taking no other arguments. The output of a non-leaf node is calculated as a flexible neuron model (see Fig. 1). From this point of view, the instruction $+_i$ is also called a flexible neuron operator with i inputs.

In the creation process of neural tree, the operator is selected randomly from function set F and terminal instruction set T. If a non-terminal instruction, i.e., $+_i (i = 2, 3, \ldots, N)$ is selected, i real values are randomly generated and used for representing the connection strength between the node $+_i$ and its children. In this

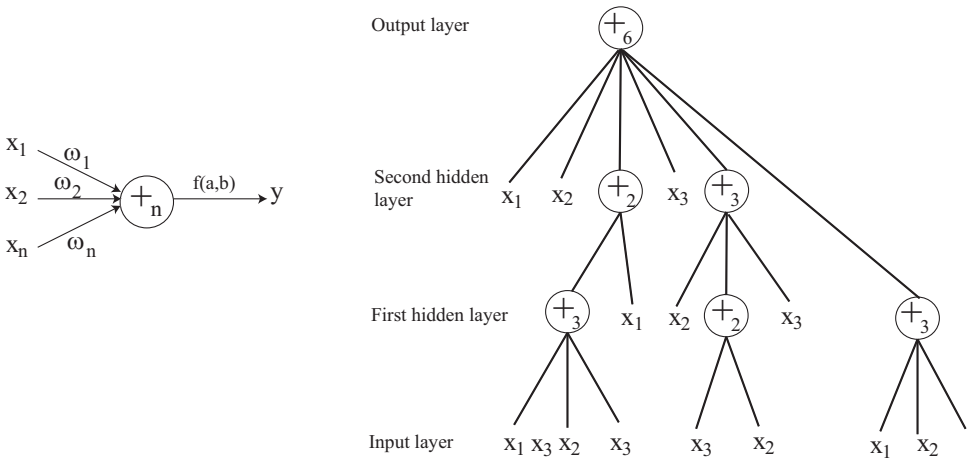

Fig. 1. A flexible neuron operator (left), a typical representation of neural tree with function instruction set $F = \{+_2, +_3, +_4, +_5, +_6\}$, and terminal instruction set $T = \{x_1, x_2, x_3\}$ (right).

study the flexible activation function used is

$$f(a_i, b_i, x) = e^{-\left(\frac{x-a_i}{b_i}\right)^2}. \quad (2)$$

Two adjustable parameters a_i and b_i are randomly created as flexible activation function parameters, and represent the position of the center of Gaussian function and $\sqrt{2}*\sigma$ (σ is standard deviation), respectively.

The output of a flexible neuron $+_n$ can be calculated as follows. The total excitation of $+_n$ is

$$net_n = \sum_{j=1}^{n} w_j * x_j \quad (3)$$

where $x_j (j = 1, 2, \ldots, n)$ are the inputs to node $+_n$. The output of the node $+_n$ is then calculated by

$$out_n = f(a_n, b_n, net_n) = e^{-\left(\frac{net_n - a_n}{b_n}\right)^2}. \quad (4)$$

The FNT can automatically design the networks structure and parameters. The network's input, output and number of layers do not need to be designed in advance. In the FNT, every node is selected randomly from the predefined instruction/operator sets S. If a leaf node is selected, this branch is terminated. If a non-leaf node $+i$ is selected, i children are created in the next layer (do not exceed pre-defined maximum depth of FNT). A typical flexible neural tree model is shown as Fig. 1. This framework allows input variables selection and over-layer connections. So the network is sparse, the connections between layers are not complete. The overall output of flexible neural tree can be computed from left to right by depth-first method, recursively.

2.1.2. Structure optimization of models

Finding an optimal or near-optimal neural tree is formulated as an evolutionary search process. In this paper, we use the following neural tree variation operators:

(1) *Mutation*: We choose four mutation operators to generate offsprings from the parents.
  (1) *Change one terminal node*: randomly select one terminal node in the tree and replace it with another terminal node, which is also generated randomly.
  (2) *Grow*: select a random leaf in a hidden layer of the neural tree and replace it with a newly generated subtree.
  (3) *Prune*: randomly select a function node in the neural tree and replace it with a terminal node selected in the set T.
  (4) *Change all the terminal modes*: select each and every terminal node in the neural tree and replace it with another terminal node.
(2) *Crossover*: First two neural trees are selected according to the predefined crossover probability P_c and one nonterminal node in the hidden layer is randomly selected for each neural tree, and then the selected subtree is swapped.
(3) *Selection*: An EP-style tournament selection [6] is applied to select the parents for the next generation. Pairwise comparison is conducted for the union of μ parents and μ offsprings. For each individual, q opponents are chosen uniformly at random from all the parents and offspring. For each comparison, if the individual's fitness is no smaller than the opponent's, it receives a selection. Select μ individuals out of parents and offsprings, those with have most wins to form the next generation. This is repeated in each generation until a predefined number of generations or the best structure is found.

2.1.3. Parameter optimization

To find the optimal parameters set (weights and activation function parameters) of a FNT model, a number of global and local search algorithms namely genetic algorithm, evolutionary programming, gradient based learning method etc. can be employed [23,24]. Particle swarm optimization (PSO) [13] is selected due to its straightforward property and fast local search capability.

According to the structure of each FNT model created, we check all the parameters containing activation function parameters (a and b) and weights (w) as a vector $(a_1, b_1, a_2, b_2, \ldots, a_i, b_i, \ldots, w_1, w_2, \ldots w_j, \ldots)$, and count their number n_i ($i = 1, 2, \ldots, M$, M is the population size of FNT model). Particle swarm optimization is used to optimize the parameters of FNT model. According to the number of parameters, the particles are randomly generated initially. Each particle x_i represents a potential solution. A swarm of particles moves through space, with the moving velocity of each particle represented by a velocity vector v_i. At each step, each particle is evaluated and keep track of its own best position, which is associated with the best fitness it has achieved so far in a vector $Pbest_i$. The best position among all the particles is kept as $Gbest$. The new velocity for particle i is

$$v_i(t+1) = v_i(t) + c_1 r_1 (Pbest_i - x_i(t)) + c_2 r_2 (Gbest(t) - x_i(t)) \quad (5)$$

where c_1 and c_2 are positive constant and r_1 and r_2 are uniformly distributed random numbers in [0,1]. Based on the updated velocities, each particle changes its position according to the following equation:

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (6)$$

2.1.4. The general learning algorithm

The general learning procedure for designing a FNT model may be described as follows:

(1) Create the initial population (flexible neural trees and their corresponding parameters).
(2) Structure optimization by neural tree variation operators as described in Section 2.1.2. The fitness function is calculated by mean square error (MSE), which is described as follows:

$$Fit = \frac{1}{N} \sum_{i=1}^{N} (x_i' - x_i)^2 \quad (7)$$

  Where N is the total number of data, x_i' and x_i are the actual output and the FNT output of i-th sample.
(3) If the better structure is found, then go to step (4), otherwise go to step (2).
(4) Parameter optimization is achieved by PSO as described in Section 2.1.3. In this stage, the tree structure or architecture of flexible neural tree model is fixed, and the best tree is taken from the end of run of the similar to GP search. All the parameters used in the best tree formulated a parameter vector to be optimized by PSO.
(5) If the maximum number of iterations of PSO algorithm is reached, or no better parameter vector is found for a significantly long time (100 steps) then go to step (6); otherwise go to step (4).
(6) If satisfactory solution is found, then stop; otherwise go to step (2).

2.2. The ensemble classifier

Good ensemble members must be both accurate and diverse, which poses the problem of generating a set of classifiers with reasonably good individual performances and independently distributed classifications for the test points [13]. Diverse individual predictors can be obtained in several ways. The main ways are described as follows:

(1) Using different algorithms to learn from the data.
(2) Changing the internal structure of a given algorithm.

(3) Learning from different adequately-chosen subsets of the data set.

The probability of success in strategy (3), the most frequently used, is directly tied to the instability of the learning algorithm [1]. That is, the method must be very sensitive to small changes in the structure of the data and/or in the parameters defining the learning process. In particular, in the case of FNTs the instability comes naturally from the inherent data and training process randomness, and also from the intrinsic non-identifiability of the model. In what follows, three ensemble methods are employed for the traffic measurements data forecasting problems (N is the number of the FNT, $f_k(x)$ is the output of the ensemble FNT).

2.2.1. The basic ensemble method

The combining network output is to simply average them together. The basic ensemble method (BEM) output is defined:

$$f_{BEM} = \frac{1}{N}\sum_{k=1}^{N} f_k(x) \tag{8}$$

This approach by itself can lead to improved performance, but does not take into account the fact that some FNTs may be more accurate than others. It has the advantage that it is easy to understand and implement and can be shown not to increase the expected error.

2.2.2. The generalized ensemble method

A generalization to the BEM method is to find weights for each output that minimize the positive and negative classification rates of the ensemble. The general ensemble method (GEM) is defined:

$$f_{GEM} = \sum_{k=1}^{N} \alpha_k f_k(x) \tag{9}$$

where the α_k's are chosen to minimize the root mean square error between the FNT outputs and the desired values. Suppose N classifiers $(C_1, C_2, \ldots, C_N)$, we check corresponding parameters as a weights vector of the ensemble predictor $(\alpha_1, \alpha_2, \ldots, \alpha_N)$. For comparison purpose, the weights vector is optimized by using PSO algorithm as described in Section 2.1.3, and the constraint $(\alpha_1 + \alpha_2 + \cdots + \alpha_N = 1)$ is added during optimization procedure.

2.2.3. The proposed classification combination method

In the BEM method, the linear combination function is used, while many nonlinear terms are added to the combination function [18]. Aladag et al. [2] proposed to use feed forward neural networks to combine forecasts obtained from different fuzzy time series forecasting models. In this paper, we first propose a new classification combination approach based on artificial neural networks to resolve the classification problem. Radial Basis Function (RBF) neural networks are employed as nonlinear combination function. The architecture of the RBF neural network and flowchart of classification combination method are shown in Fig. 2.

Classification produced from different FNT models are inputs of RBF neural networks and the output is combined classification. The number of inputs of the neural network is equal to the number of used classification models. For the neurons in the hidden layer and the output layer, the output functions are described as followed, respectively:

$$f_i = e^{-(x_1-\mu_1)^2/2\sigma_1^2} \star e^{-(x_2-\mu_2)^2/2\sigma_2^2} \cdots \star e^{-(x_N-\mu_N)^2/2\sigma_N^2} \tag{10}$$

$$y = f_1 W_1 + f_2 W_2 + \cdots + f_M W_M \tag{11}$$

Where x_i is the i-th input. The parameters (μ, σ, W) need be optimized. The number is $2NM+M$. In this study, PSO algorithm is used to train a neural network.

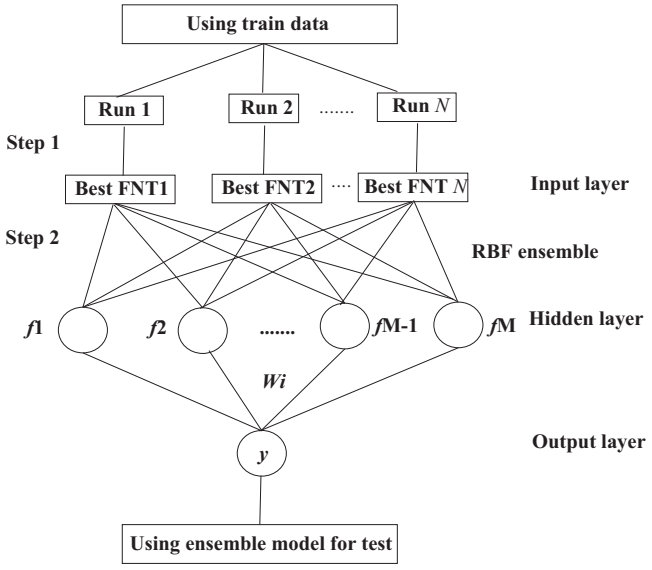

Fig. 2. The architecture of the RBF neural network and flowchart of classification combination method.

Table 1
Parameters used in the flexible neural tree model.

Parameter	Initial value
Population size *PS*	30
Overall mutation probability P_M	0.4
Mutation rate *mr*	0.4
Crossover rate T_c	0.7
Maximum local search steps	2000
Initial connection weights	rand [−1, 1]
Initial parameters a_i and b_i	rand [0,1]

3. Experimental results and illustrative examples

In the paper, we used two real biological datasets to train and test the performance of the FNT models for somatic mutation prediction. The first dataset comprises 48 triple negative breast cancer Agilent SureSelect v1 exome capture tumour/normal pairs sequenced using the Illumina genome analyzer as 76 bp pair-end reads [15,36]. This data consist of 1015 somatic mutations, 471 germline and 1883 wild-type positions. The second dataset is whole genome shotgun data, which consists of four whole human genome pairs sequencing using Life Technologies SOLiD system as 25-50 bp pair-end reads [15]. After the targeted positions revalidated, 113 somatic mutations, 57 germline mutations and 337 wild-types are obtained. In order to facilitate the comparative performance of our method, we use the 106 feature for each mutation from Samtools and GATK aligning reads from paired tumour/normal samples to reference genome, which is the same with Ding et al. [15]. The parameters used in the flexible neural tree model are chosen experientially and given in Table 1. In addition, all experiments are performed using a 1.8 GHz process with 1 GB of RAM.

Six criterions (*sensitivity*, *specificity*, *accuracy*, *precision*, *F−measure* and *G−mean*) are used to test the performance of the method. Firstly, we define four variables. *TP* is the number of somatic mutations correctly identified as somatic. *FP* is the number of the non-somatic mutations identified as somatic. *TN* is the number of the non-somatic mutations identified as non-somatic, while *FN* is the number of the somatic mutations identified as non-somatic. Six criterions are defined as follows:

$$sensitivity = \frac{TP}{TP+FN} \tag{12}$$

Table 2
The cross-validation results of Shah's methods [15], FNT, RBF and three ensemble methods.

Model	Sensitivity	Specificity	Accuracy	Precision	F-measure	G-mean
RF	0.9901	0.9422	0.9567	0.8808	0.9323	0.9659
BART	0.9901	0.9584	0.9679	0.9112	0.9490	0.9741
SVM	0.9901	0.9405	0.9555	0.8777	0.9306	0.9650
Logit	0.9901	0.8704	0.9065	0.7672	0.8645	0.9283
RBF	0.9901	0.9426	0.9569	0.8816	0.9327	0.9661
FNT	0.9892	0.9588	0.9679	0.9119	0.9490	0.9739
BEM	0.9901	0.9592	0.9685	0.9128	0.9499	0.9746
GEM	0.9901	0.9600	0.9691	0.9145	0.9508	0.9750
The proposed ensemble	0.9911	0.9622	0.9709	0.9187	0.9535	0.9766

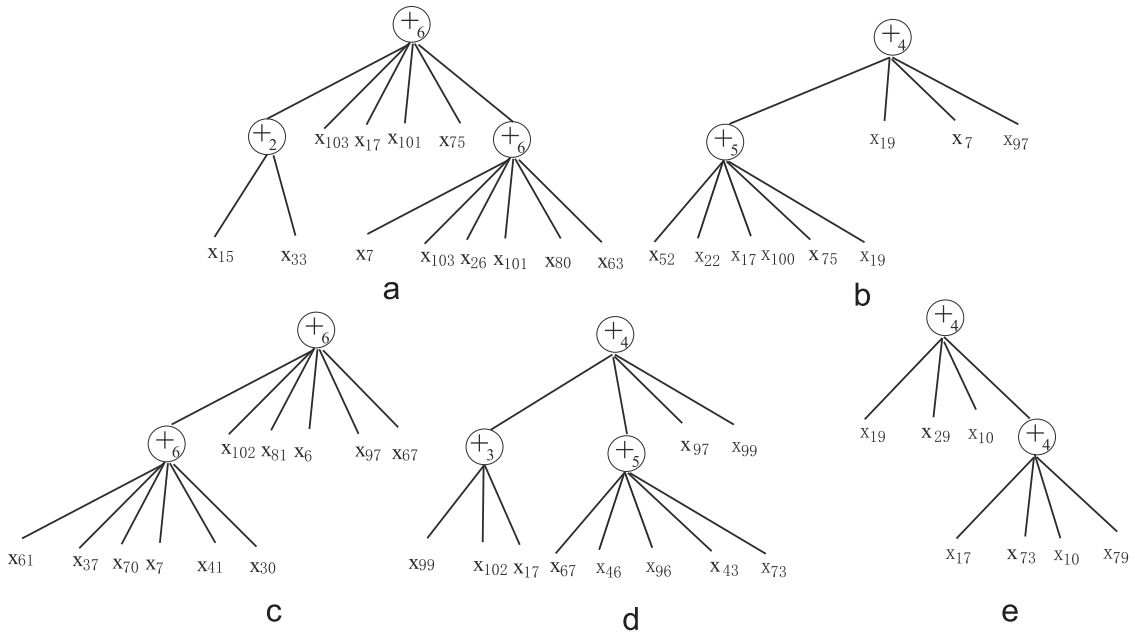

Fig. 3. The evolved FNT trees for five runs.

$$specificity = \frac{TN}{TN+FP} \tag{13}$$

$$accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{14}$$

$$precision = \frac{TP}{TP+FP} \tag{15}$$

$$F-measure = \frac{(1+\beta^2) \times sensitivity \times precision}{\beta^2 \times sensitivity + precision} \tag{16}$$

$$G-mean = \sqrt{sensitivity \times specificity} \tag{17}$$

3.1. Classification performance of our method

The two real biological datasets are normalized to the interval [0, 1] with following formula $x' = (x - x_{min})/(x_{max} - x_{min})$. 106 input variables are used for constructing a FNT model. The instruction sets used to create an optimal FNT classifier is $S = F \cup T = \{+2, +3, +4, +5, +6\} \cup \{x_1, x_2, \ldots, x_{106}\}$. Where x_i $(i = 1, 2, \ldots, 106)$ denotes one of the 106 features.

The exome capture data are used to test our method using 10 cross-validations. Table 2 depicts the detection performance of Shah's methods [15] consist of random forest (RF), Bayesian additive regression tree (BART), logistic regression (Logit) and support vector machine (SVM), RBF neural networks which is optimized by PSO algorithm, best FNT model and three ensemble methods (BEM, GEM and our proposed ensemble). From Table 2, it can be seen that FNT performs better than RF, SVM, Logit and RBF except BART, and our proposed ensemble method is more accurate than other ensemble methods.

To test of the generalization performance of our method, the exome capture data are used to train and the whole genome shotgun data as test data. The results of optimizing method of FNT model are associated with the initial population (flexible neural trees and their corresponding parameters). With the best classification, we could gain different FNT models during several experiments. In this experiment, we make five runs and gain five optimal classification models. The result is described in Fig. 3. From five optimal FNT model, we could select the best classification performance which is from Fig. 3(a). Table 2 depicts the detection performance of RF, BART, Logit and SVM, RBF, FNT model, BEM, GEM and our proposed ensemble. The detection performance of RF, BART, Logit and SVM are from [15].

From Table 3, we can see that the best FNT model is most accurate than other models. And the three ensemble methods are better than the other six single classifiers. Our proposed classification combination method performs better. We also adopt the ROC graph to evaluate the performance of our method. The ROC curve can be plotted by the true positive (TP) rate and false positive (FP) rate in Fig. 4. From Fig. 4, we can see that the proposed method performs better than other methods. From these testing

Table 3
Detection performance using Shah's methods, FNT, RBF and three ensemble methods.

Model	Sensitivity	Specificity	Accuracy	Precision	F-measure	G-mean
RF	0.8850	0.9518	0.9369	0.8403	0.8621	0.9178
BART	0.7876	0.9949	0.9487	0.9780	0.8725	0.8852
Logit	0.8496	0.9391	0.9191	0.8	0.8240	0.8932
SVM	0.7876	0.9949	0.9487	0.9780	0.8725	0.8852
RBF	0.8495	0.9568	0.9329	0.8496	0.8496	0.9015
FNT	0.8461	0.9898	0.9566	0.9596	0.8962	0.9151
BEM	0.8495	0.9949	0.95858	0.9796	0.9099	0.9193
GEM	0.8584	0.9949	0.9664	0.9798	0.9151	0.9241
The proposed ensemble	0.8672	0.9974	0.9704	0.9899	0.9245	0.9300

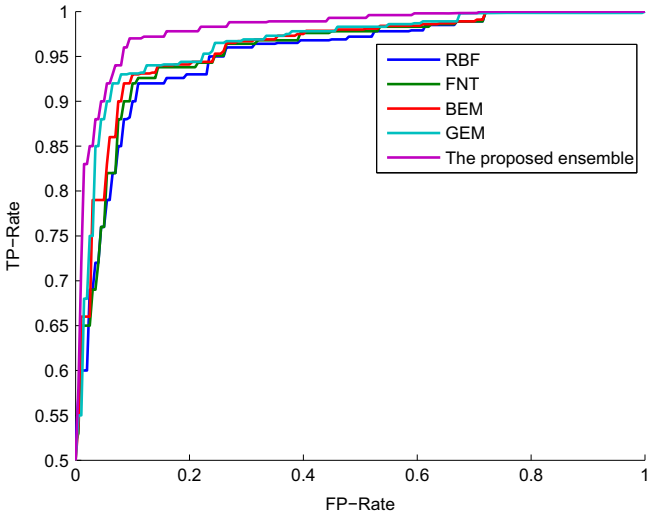

Fig. 4. The ROC curves of different classifiers.

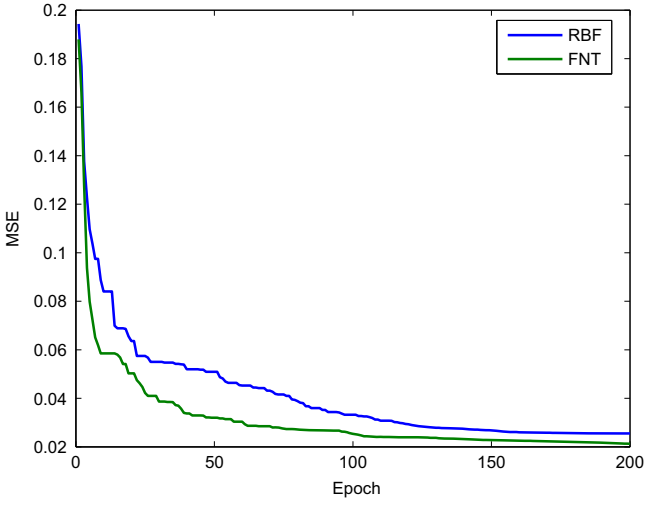

Fig. 5. The convergence procedure using RBF and FNT.

results, it can be seen that, using the limited dataset, the trained models should generalize well and are likely robust to overfitting.

3.2. Analysis of the convergence procedure

We also analyze the convergence procedure of the FNT model and our proposed ensemble. The parameter *Maximum epoch* is set as 200, and MSE is selected as fitness function.

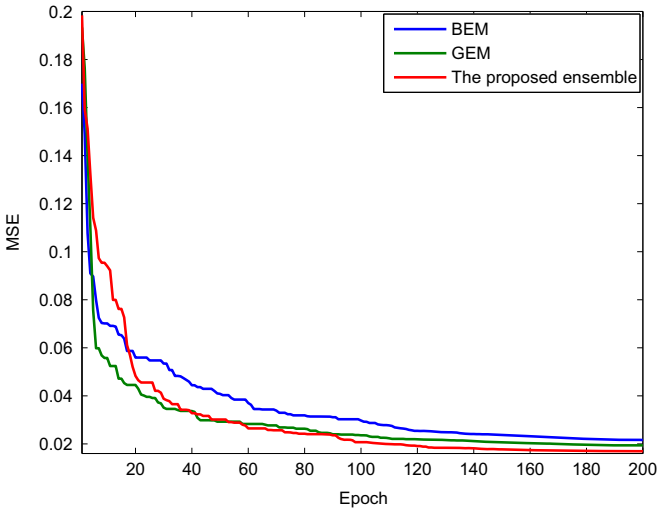

Fig. 6. The convergence procedure using three ensemble methods: BEM, GEM and our proposed ensemble.

Table 4
The import features selected by the FNT (F_i means the normalized version of the i-th feature [15]).

Feature	The definition of feature
x_7	Sum of squares of reference base qualities
x_{15}	Sum of squares of tail distance for reference bases
x_{17}	Sum of squares of tail distance for non-reference bases
x_{26}	Sum of reference base qualities
x_{33}	Sum of squares of non-reference mapping qualities
x_{63}	AF: allele frequency for each non-ref allele
x_{75}	Allelic depths for the non-ref allele
x_{80}	$P(D\|G_i = bb)$, phred-scaled
x_{101}	Sum of squares of non-reference base quality ratio F29/F9
x_{103}	Sum of squares of non-reference mapping quality ratio F33/F13

From Fig. 5, FNT has faster descent speed than RBF, and the last MSE using FNT is much lower than RBF. The performance of convergence procedure using three ensemble methods: BEM, GEM and our proposed ensemble is illustrated in Fig. 6. BEM and GEM are very simple and linear, so in the first 30 epochs, BEM and GEM have the faster convergence speed. Due to our proposed ensemble method with complex nonlinear structure, our method has the strong learning ability. Thus from 50-th epoch, our method converges faster. But our method will take much time due to the high computational cost.

3.3. Input/feature selection with FNT

It is often difficult to select features for the classification problem, especially when the feature space is large. A fully connected neural network classifier usually cannot do this. In the perspective

Table 5
Detection performance using RBF with all features (RBF-all) and RBF with 10 features (RBF-10).

Model	Sensitivity	Specificity	Accuracy	Precision	F-measure	G-mean	Runtime
RBF-all	0.8495	0.9568	0.9329	0.8496	0.8496	0.9015	180 s
RBF-10	0.8584	0.9797	0.9527	0.9238	0.8899	0.9170	45 s

of FNT framework, the nature of model construction procedure allows the FNT to identify important input features in building a classification that is computationally efficient and effective.

From the best FNT model (Fig. 3(a)), we could gain 10 import features for somatic mutation detection. The definitions of features are listed in Table 4. From Table 4, we can see that the features mainly refer to the base qualities, mapping qualities, allelic depths and tail distance. These features almost consistent with the critical parameters that could enhance the accuracy of called NGS mutation [22].

To further verify the validity of the features selected by FNT, we use 10 import features to extracting a subset of the original data. RBF neural networks are used to somatic mutation detection with the subset containing train and test data. The results are listed in Table 5. It is seen that the features selected by the FNT could make classifier builded more computationally efficient and effective.

4. Concluding remarks

To detect somatic mutations in tumour-normal paired NGS sequencing data, flexible neural tree model (FNT) is proposed in this paper with a focus on improving the classification performance by three ensemble methods and reducing the input features.

From the architecture perspective, a FNT can be seen as a flexible multi-layer feedforward neural network with over-layer connections and free parameters in activation functions, and it is convenient to model and revolved complex classification problem. As evident from Tables 2 and 3, the FNT model performs better than other supervised machine learning algorithms consisting of random forest, Bayesian additive regression tree, logistic regression and support vector machine, RBF neural networks. To reduce the false positive rate, we propose a new classification combination approach based on artificial neural networks which applies Radial Basis Function neural networks as nonlinear combination function. Tables 2 and 3 show that our proposed ensemble method is more accuracy than two traditional ensemble methods and single classifiers.

In the process of constructing a classifier, the FNT model can select the proper input variables or time-lags automatically, so as to identify important features of problem resolved. By the FNT, we select the 10 features from 106 inputs. And experimental result shows that RBF neural networks with the 10 features selected are more computationally efficient and effective than with 106 features. The 10 features selected could be as critical parameters of analysis and detection of NGS mutation.

Acknowledgements

This research was supported by the PhD research startup foundation of Zaozhuang University (No. 1020702), Shandong Provincial Natural Science Foundation, China (No. ZR2015PF007), and partially supported by the Key Subject Research Foundation of Shandong Province and the Shandong Provincial Key Laboratory of Network Based Intelligent Computing.

References

[1] A. Abraham, N.S. Philip, P. Saratchandran, Modeling chaotic behavior of stock indices using intelligent paradigms, Int. J. Neural Parallel Sci. Comput. 11 (1–2) (2003) 143–160.
[2] C.H. Aladag, E. Egrioglu, U. Yolcu, Forecast combination by using artificial neural networks, Neural Process. Lett. 32 (2010) 269–276.
[3] Q. Cai, H. He, H. Man, Hybrid learning based on multiple self-organizing maps and genetic algorithm, in: International Joint Conference on Neural Networks (IJCNN), 2011, pp. 2313–2320.
[4] Q. Cai, H. He, H. Man, Imbalanced evolving self-organizing learning, Neurocomputing 133 (2014) 258–270.
[5] R. Chellappa, C.L. Wilson, S. Sirohey, Human and machine recognition of faces: a survey, Proc. IEEE 83 (5) (1995) 705–740.
[6] K. Chellapilla, Evolving computer programs without subtree crossover, IEEE Trans. Evol. Comput. 1 (1997) 209–216.
[7] Y.H. Chen, A. Abraham, B. Yang, Feature selection and classification using flexible neural tree, Neurocomputing 70 (1–3) (2006) 305–313.
[8] Y.H. Chen, A. Abraham, Y. Zhang, Ensemble of flexible neural trees for breast cancer detection, Int. J. Inf. Technol. Intell. Comput. 1 (1) (2006) 187–201.
[9] Y.H. Chen, B. Yang, Q.F. Meng, Small-time scale network traffic prediction based on flexible neural tree, Appl. Soft Comput. 12 (2012) 274–279.
[10] Y.H. Chen, B. Yang, J. Dong, A. Abraham, Time series forecasting using flexible neural tree model, Inf. Sci. 174 (3/4) (2005) 219–235.
[11] Y.H. Chen, S.Y. Jiang, A. Abraham, Face recognition using dct and hybrid flexible neural tree, in: 2005 International Conference on Neural Networks and Brain, vol. 3, 2005, pp. 1459–1463.
[12] Y.H. Chen, B. Yang, J.W. Dong, Nonlinear system modelling via optimal design of neural trees, Int. J. Neural Syst. 14 (2) (2004) 125–137.
[13] Y.H. Chen, B. Yang, A. Abraham, Flexible neural trees ensemble for stock index modeling, Neurocomputing 70 (4–6) (2007) 697–703.
[14] M.A. DePristo, E. Banks, R. Poplin, K.V. Garimella, J.R. Maguire, C. Hartl, A. A. Philippakis, G. del Angel, M.A. Rivas, M. Hanna, A. McKenna, T.J. Fennell, A. M. Kernytsky, A.Y. Sivachenko, K. Cibulskis, S.B. Gabriel, D. Altshuler, M.J. Daly, A framework for variation discovery and genotyping using next-generation DNA sequencing data, Nat. Genet. 43 (5) (2011) 491–498.
[15] J. Ding, A. Bashashati, A. Roth, A. Oloumi, K. Tse, T. Zeng, G. Haffari, M. Hirst, M. A. Marra, A. Condon, S. Aparicio, S.P. Shah, Feature-based classifiers for somatic mutation detection in tumour-normal paired sequencing data, Bioinformatics 28 (2) (2012) 167–175.
[16] L. Ding, M.C. Wendl, D.C. Koboldt, E.R. Mardis, Analysis of next-generation genomic data in cancer: accomplishments and challenges, Hum. Mol. Genet. 19 (R2) (2010) R188–R196.
[17] S.A. Frank, M.A. Nowak, Problems of somatic mutation and cancer, Bioessays 26 (3) (2004) 291–299.
[18] P.S.A. Freitas, A.J.L. Rodrigues, Model combination in neural-based forecasting, Eur. J. Oper. Res. 173 (2006) 801–814.
[19] L. Goh, G.B. Chen, I. Cutcutache, B. Low, B.T. Teh, S. Rozen, P. Tan, Assessing matched normal and tumor pairs in next-generation sequencing studies, PLoS One 6 (3) (2011) e17810.
[20] C. Greenman, P. Stephens, R. Smith, G.L. Dalgliesh, et al., Patterns of somatic mutation in human cancer genomes, Nature 446 (7132) (2007) 153–158.
[21] Y.N. Guo, Q.H. Wang, A. Abraham, Flexible neural trees for online hand gesture recognition using surface electromyography, J. Comput. 7 (5) (2012) 1099–1103.
[22] P. Jia, F. Li, J. Xia, H. Chen, H. Ji, W. Pao, Z. Zhao, Consensus rules in variant detection from next-generation sequencing data, PLoS One 7 (6) (2012) e38470.
[23] S. Kimura, K. Ide, A. Kashihara, Inference of s-system models of genetic networks using a cooperative coevolutionary algorithm, Bioinformatics 21 (2005) 1154–1163.
[24] S. Kirkpatrick Jr, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671–680.
[25] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin, The sequence alignment/map format and SAMtools, Bioinformatics 25 (16) (2009) 2078–2079.
[26] R. Li, Y. Li, X. Fang, H. Yang, J. Wang, K. Kristiansen, J. Wang, SNP detection for massively parallel whole-genome resequencing, Genome Res. 19 (6) (2009) 1124–1132.
[27] M. Löwer, B.Y. Renard, J. de Graaf, M. Wagner, C. Paret, C. Kneip, O. Treci, M. Diken, C. Britten, S. Kreiter, M. Koslowski, J.C. Castle, U. Sahin, Confidence-based somatic mutation evaluation and prioritization, PLoS Comput. Biol. 8 (9) (2012) e1002714.
[28] N.L. Nehrt, T.A. Peterson, D. Park, M.G. Kann, Domain landscapes of somatic mutations in cancer, BMC Genom. 13 (Suppl 4) (2012) S9.

[29] M.K. McConechy, J. Ding, M.C. Cheang, K.C. Wiegand, et al., Use of mutation profiles to refine the classification of endometrial carcinomas, J. Pathol. 228 (1) (2012) 20–30.
[30] C. Meldrum, M.A. Doyle, R.W. Tothill, Next-generation sequencing for cancer diagnostics: a practical perspective, Clin. Biochem. Rev. 32 (4) (2011) 177–195.
[31] R. Nielsen, J.S. Paul, A. Albrechtsen, Y.S. Song, Genotype and SNP calling from next-generation sequencing data, Nat. Rev. Genet. 12 (6) (2011) 443–451.
[32] G. Perkins, T.A. Yap, L. Pope, A.M. Cassidy, J.P. Dukes, R. Riisnaes, C. Massard, P. A. Cassier, S. Miranda, J. Clark, K.A. Denholm, K. Thway, et al., Multi-purpose utility of circulating plasma DNA testing in patients with advanced cancers, PLoS One 7 (11) (2012) e47020.
[33] E.D. Pleasance, R.K. Cheetham, P.J. Stephens, D.J. McBride, S.J. Humphray, et al., PA comprehensive catalogue of somatic mutations from a human cancer genome, Nature 463 (7278) (2010) 191–196.
[34] A. Rajini, V.K. David, A comparative performance study on hybrid swarm model for microarray data, Int. J. Comput. Appl. 30 (6) (2011) 10–14.
[35] S. Saha, A. Ekba, Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition, Data Knowl. Eng. 85 (2013) 15–39.
[36] S.P. Shah, A. Roth, R. Goya, et al., The clonal and mutational evolution spectrum of primary triple-negative breast cancers, Nature 486 (7403) (2012) 395–399.
[37] M.R. Stratton, P.J. Campbell, P.A. Futreal, The cancer genome, Nature 458 (2009) 719–724.
[38] J.K. Teer, L.L. Bonnycastle, P.S. Chines, N.F. Hansen, N. Aoyama, et al., Systematic comparison of three genomic enrichment methods for massively parallel DNA sequencing, Genome Res. 20 (2010) 1420–1431.

Bin Yang is the teacher in Zaozhuang University. He received his B.Sc. and Master degree in School of Information Science and Engineering from University of Jinan. He received his Ph.D. in School of Information Science and Engineering from Shandong University. His research interests include hybrid computational intelligence and their applications in time-series prediction, system identification and gene regulatory network.

Yuehui Chen was born in 1964 in Shandong Province of China. He received his B.Sc. degree in mathematics/automatics from the Shandong University of China in 1985, and Master and Ph.D. degrees in electrical engineering from the Kumamoto University of Japan in 1999 and 2001. During 2001–2003, he had worked as the Senior Researcher of the Memory-Tech Corporation at Tokyo. Since 2003 he has been a member at the Faculty of Electrical Engineering in Jinan University, where he is currently head of the Laboratory of Computational Intelligence. His research interests include Evolutionary Computation, Neural Networks, Fuzzy Logic Systems, Hybrid Computational Intelligence and their applications in time-series prediction, system identification, intelligent control, intrusion detection systems, web intelligence and bioinformatics. He is the author and co-author of more than 70 technique papers. Professor Yuehui Chen is a member of IEEE, the IEEE Systems, Man and Cybernetics Society and the Computational Intelligence Society, a member of Young Researchers Committee of the World Federation on Soft Computing, and a member of CCF Young Computer Science and Engineering Forum of China. More information at: http://cilab.ujn.edu.cn.