Journal of Biomedical Informatics 81 (2018) 102–111

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

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

Disease genes prediction by HMM based PU-learning using gene expression profiles

Ozra Nikdelfaz, Saeed Jalili*

Tarbiat Modares University, Computer Engineering Department, Islamic Republic of Iran

ARTICLE INFO

Keywords:
Disease gene prediction
Hidden Markov model
Positive-unlabeled learning
Gene expression profile

ABSTRACT

Predicting disease candidate genes from human genome is a crucial part of nowadays biomedical research. According to observations, diseases with the same phenotype have the similar biological characteristics and genes associated with these same diseases tend to share common functional properties. Therefore, by applying machine learning methods, new disease genes are predicted based on previous ones. In recent studies, some semi-supervised learning methods, called Positive-Unlabeled Learning (PU-Learning) are used for predicting disease candidate genes. In this study, a novel method is introduced to predict disease candidate genes through gene expression profiles by learning hidden Markov models. In order to evaluate the proposed method, it is applied on a mixed part of 398 disease genes from three disease types and 12001 unlabeled genes. Compared to the other methods in literature, the experimental results indicate a significant improvement in favor of the proposed method.

1. Introduction

DNA microarray is a collection of microscopic spots attached to a solid surface to measure the expression levels of genes. This technology enables researchers to study genes in the human genome, at the same time [1]. Analyzing gene expression levels in disease genes represents a consistent pattern of different expression levels in each disease [2]. Many complex diseases, namely, cancer, diabetes and cardiovascular have a severe impact on human health. Since these diseases are consequences of complicated interactions of multiple genes, predicting disease candidate genes are very important for understanding the mechanism of diseases and discovering the therapeutic targets. So far, several methods have been proposed to predict disease candidate genes based on different type of biological data such as sequence-based features [3,4], function [5–9] and network [10–14].

Commonly, learning techniques are applied on a set of binary labeled instances (i.e., positive and negative) to learn a model that discriminates positive instances from negative ones. But in the case of disease candidate genes prediction problem we have only a small set of positive instances (P) (i.e., disease genes) and a large set of unlabeled genes (U) (i.e., those genes that we are going to predict some of them positive) [15]. These type of problems are solved by applying positive-unlabeled learning (PU-learning) approach.

The existing methods are classified into two families. The first family of methods [16–20] proceeds in two steps. First, extracting some reliable negative (RN) instances, then applying a supervised or semi-supervised binary learning method. Mordelet et al. [16] proposed a bagging method (ProDiGe) that iteratively selects random subsets of genes (RS) from U and then learns several classifiers using bias support vector machine (SVM) to discriminate disease genes set from each subset RS. These classifiers were subsequently aggregated to generate the final classifier. Given that RS's are likely to contain less noise than the original set U, ProDiGe is able to perform better than classical binary classifiers that inconveniently take U as negative training data. Yang et al. [17] devised a multi-level PU learning algorithm (PUDI) to build a classifier with better performance for predicting disease candidate genes. They partitioned the unlabeled genes set into multiple positive and negative sets with confidence scores for building the classifier. Yang et al. [18] proposed an effective PU learning framework that integrates three biological data sources: gene expression data, gene ontology and human protein interaction data. Then they forged an ensemble of machine learning classifiers for disease candidate genes prediction. Yousef et al. [19] proposed a method which consists of four layers. In the first layer, feature vectors are created by taking the amino

* Corresponding author.
E-mail addresses: a.nikdel@modares.ac.ir (O. Nikdelfaz), sjalili@modares.ac.ir (S. Jalili).

https://doi.org/10.1016/j.jbi.2018.03.006
Received 15 August 2017; Received in revised form 22 November 2017; Accepted 12 March 2018
Available online 20 March 2018
1532-0464/

acid sequences of proteins into four different feature sets such as NA,[1] GA,[2] AC[3] and MA.[4] The second layer, selects negative genes, using cosine distance from unlabeled genes per feature vectors. It creates a set of reliable negative genes by extracting the intersection of reliable negative genes within four feature vectors. In the third layer, a SVM model per feature vector set is learned. In the last layer, a decision tree (C4.5) is applied as a fusion method to combine the results of four independent SVM predictors and to make the final decision. Jowkar et al. [20] proposed a Perceptron ensemble of graph-based pu-learning (PEGPUL) method. First, they extract a reliable set of positive and negative genes, and then build a similarity graph of genes using metric learning by multi-rank-walk method to perform inference from unlabeled genes. Finally, a Perceptron ensemble is learned from three classifiers: SVM, K-nearest neighbor (KNN) and decision tree.

The second clan of methods [21,22] reduces the problem into a learning problem with high one-sided noise by treating the unlabeled set as noisy negative set. Smalter et al. [21] applied SVM classifier using protein-protein interactions topological features in addition to sequence-derived and evolutionary features. Whilst Radivojac et al. [22] made three individual SVM classifiers using three types of features, namely protein-protein interactions network, protein sequence and protein functional information. Next, in order to predict disease candidate genes, a final classifier combines the predictions of individual classifiers.

In this research, we propose a method that first for each disease type, it clusters disease genes by using semantic similarity as the distance measure between genes, then for each cluster, a HMM model is learned and its threshold is calculated. The proposed method is compared with several previous practices such as Xu's [23], Smalter's [21], ProDiGe [16], PUDI [17], EPU [18], SFM [19] and PEGPUL [20]. Experimental results acknowledge improvements in both precision and recall metrics made over these past methods.

The rest of the paper is organized as follows: In Section 2, basic concepts such as semantic similarity calculation, hidden Markov model are introduced. The proposed method is explained in Section 3. In Section 4, the datasets, evaluation metrics and results are presented. Analysis and discussion is presented in Section 5. Section 6 concludes the paper.

2. Basic concepts

2.1. Semantic similarity calculation

A Gene Ontology (GO) defines concepts/classes that are used to describe gene functionality and relationships between these concepts. GO discriminates these functions along three aspects: biological processes (BP), cellular components (CC) and molecular functions (MF). Semantic similarity based on ontology is defined as the closeness in meaning between two ontology terms or two sets of terms annotating two genes. Semantic similarity measures have become important in bioinformatics as they provide a way of quantifying the functional correspondence and relevancy between genes that is complementary to both experimental evidence and sequence-based approaches. This is accomplished by augmenting genes with annotating terms of a chosen ontology and then quantifying the similarities between those terms. *Gene Ontology semantic similarity Tool* (GOssTo) is a tool to calculate semantic similarity between gene products according to the gene ontology [24]. Several semantic similarity measures have been proposed [25,26], which typically rely on: corpus-based or structure-based. We have utilized the second approach in the proposed method. In structure-based approach, information content (*IC*) of a term is computed from the number of its descendants in the GO structure [27]. The *IC* of term t_i is presented in relation (1), where $desc(t_i)$ means the number of descendants of term t_i, and *totalterms* is the number of terms in GO.

$$IC(t_i) = \frac{\log((desc(t_i)+1)/totalterms)}{\log(1/totalterms)} = 1 - \frac{\log(desc(t_i)+1)}{\log(totalterms)} \quad (1)$$

Suppose genes G_A and G_B are annotated with term sets $T_A\{t_1,t_2,t_3,...,t_m\}$ and $T_B\{t'_1,t'_2,t'_3,...,t'_n\}$, respectively, FS_{sim} [25] defines the functional similarity between G_A and G_B by relation (2).

$$FS_{sim}(G_A,G_B) = \frac{\sum_{t_i \in (T_A \cap T_B)} IC(t_i)}{\sum_{t_j \in (T_A \cup T_B)} IC(t_j)} \quad (2)$$

2.2. Hidden Markov model

An HMM which was first developed by [28], is a probabilistic model in which the system is assumed to be a Markov process with hidden states. HMM is used for representing probability distribution over sequences of observations.

In order to fully determine a discrete HMM, the following elements should be defined [29]:

- N: the number of distinct hidden states.
- M: the number of observation symbols.
- $\Pi = \{\pi_i\}$: the initial state vector where $\pi_i = P(q_1 = S_i), 1 \leqslant i \leqslant N$ is the probability of S_i being the first state of a state sequence.
- $A = \{a_{ij}\}$: the transition probability matrix in which an element a_{ij} represents the probability to go from state i to sate j: $a_{ij} = P(q_{t+1} = S_j | q_t = S_i), 1 \leqslant i,j \leqslant N$. Transition probabilities must satisfy the normal stochastic constraints: $\forall\, a_{ij} \geqslant 0, 1 \leqslant i,j \leqslant N$, $\sum_{j=1}^{N} a_{ij} = 1$.
- $B = \{b_j(k)\}$: the emission probability matrix where $b_j(k)$ specifies the likelihood of the kth observation symbol, v_k, in the alphabet when the model is in state S_j: $b_j(k) = P(O_t = v_k | q_t = S_j), 1 \leqslant j \leqslant N, 1 \leqslant k \leqslant M$.

3. The proposed method

Fig. 1 demonstrates the architecture of the proposed method that consists of two phases, namely, learning phase and prediction phase. The learning phase is composed of four steps. The first and second steps are done in parallel. In the first step, the genes of each disease type (DT_k) are clustered (CL_{kj}) by calculating semantic similarity between disease type based on a gene ontology. In the second step, for each disease (D_i), the expression levels of its genes in of all their time slots or conditions are quantized. It should be noticed that since each disease type includes some diseases, so, after clustering the genes of each disease are scattered between different clusters. Therefore in the third step, the quantized expression profiles of each disease (TSC_i) are mapped to their corresponding disease type clusters. In the last step, a HMM model ($\lambda_{ij} = (\Pi_{ij}, A_{ij}, B_{ij}, t_{ij})$) per cluster ($TSC_{ij}$) in each disease ($D_i$) is learned. Moreover, in order to tune the prediction of disease candidate genes in the future, the corresponding threshold (t_{ij}) of each learned HMM (λ_{ij}) is calculated from the training set of each cluster (TSC_{ij}).

In the predication phase, first, each time slot or condition expression level of each unlabeled genes are mapped to the closest value that are determined in the second step of learning phase, then, based on the learned HMM models and their corresponding thresholds (λ_{ij}, t_{ij}), the labels of unlabeled genes are predicted.

[1] Normalized Moreau-Broto autocorrelation.
[2] Geary auto correlation.
[3] Auto covariance.
[4] Moran auto-correlation.

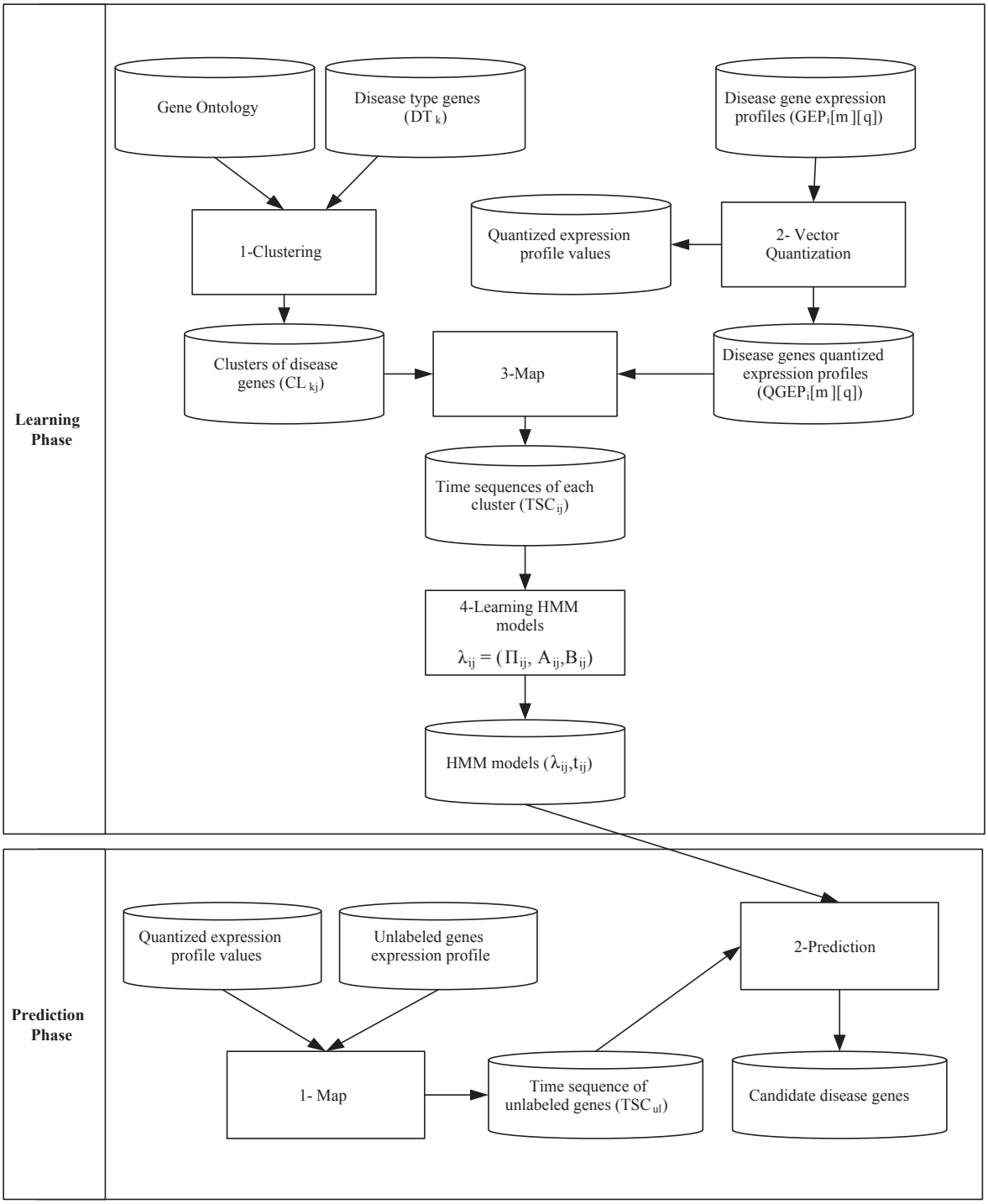


Fig. 1. The architecture of the proposed disease candidate genes prediction method.

3.1. Disease type genes clustering

Prior to applying a clustering method, a similarity/distance measure must be determined reflecting the degree of closeness between samples. We compute the semantic similarities between all disease type genes based on a gene ontology and relations (1) and (2) and store them in a matrix (MDT_k). Then, we apply a combination of K-means++ and K-means algorithms on each disease type (DT_k) using MDT_k to make a specific number of clusters (CL_{kj}). Algorithm 1 shows the proposed disease genes clustering method.

In the first part of Algorithm 1 (Lines 1:4), K-means++ is applied to make initial set of cluster centers (CDT_k) per disease type (DT_k). To accomplish this, it chooses one center uniformly at random among the disease type genes, then for each disease type gene (x), it computes the minimum distance between x and centers (CDT_k) that has already been chosen by using MDT_k according to relation (3).

$$distance(x,CDT) = Min(MDT(x,c)) \ ,\forall\ c \in CDT \tag{3}$$

Next, it chooses one new disease type gene (x') as a new center with probability calculated using relation (4). These steps are repeated until the L centers are chosen.

$$probability(x',CDT) = \frac{Distance(x',CDT)^2}{\sum_{x\in P} Distance(x,CDT)^2} \quad (4)$$

In the second part of Algorithm 1 (Lines 5:12), K-means method is applied on disease type genes to make the final set of clusters (CL).

Algorithm 1. Disease type genes clustering

Input: P: Set of genes of a disease type,
$P = \{x^1,x^2,...,x^m\}$, L: Number of clusters
Output: CL: Set of clusters,
$CL = \{cl_1,cl_2,...,cl_L\}$
K-menas++ part
1: Choose first center c1 uniformly at random from P,and let CDT = {c1}
2: **loop** (L-1) times:
3: Choose next center $c_i = x' \in P$ with probability according to relation (4)
$CDT \leftarrow CDT \cup c_i$
4: **end loop**
K-means part
5: $CL \leftarrow CDT$
6: **loop** Until CL converge:
7: **loop** For all $x^t \in P$:
8: $b_i^t \leftarrow \begin{cases} 1 & if FS_{sim}(x^t,cl_i) = \\ & min_j(FS_{sim}(x^t,cl_j)) \\ 0 & otherwise \end{cases}$
9: **end loop**
10: **loop** For all $cl_i, i = 1,...,L$
11: $CL_i \leftarrow \sum_t b_i^t x^t / \sum_t b_i^t$
12: **end loop**
13: **end loop**

In order to determine the number of clusters in each disease type (k), a density metric is proposed which is calculated according to relation (5). The $Density_{kj}$ is the total closeness each pair of disease type genes in jth cluster of disease type k.

$$Density_{kj} = \sum_{a=1}^{n_{kj}} \sum_{b=1}^{n_{kj}} FS_{sim}(a,b) \quad (5)$$

where n_{kj} denotes the number of disease type genes in jth cluster of disease type k and $FS_{sim}(a,b)$ is defined by relation (2).

If there are L_k clusters in disease type k, their density values are summed up using relation (6).

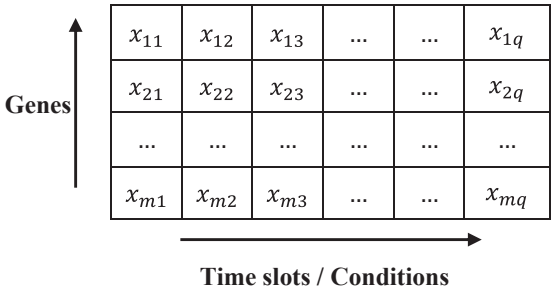
$$TotalDensity_k = \sum_{l=1}^{L_k} Density_{kl}(l) \quad (6)$$

For different values of L_k, the one, L_k' with minimum $TotalDensity_k$ is the final number of clusters according to relation (7).

$$L_k' = argmin(TotalDensity_k) \quad (7)$$

3.2. Gene expression profile quantization

Gene expression profiling (GEP) is an iterative process of measuring the expression level of all genes at once in different conditions or at different time slots. Fig. 2 shows a gene expression profile matrix. Each x_{ab} represents the expression level of ath gene in bth time slot or condition. Since there are many scattered expression levels of genes in a specific time slot or condition, so we map each expression level x_{ab} to fewer numbers. To achieve this goal, we use a sophisticated vector quantization (VQ) technique, K-means which is demonstrated in



x_{11}	x_{12}	x_{13}	...	...	x_{1q}
x_{21}	x_{22}	x_{23}	...	...	x_{2q}
...	...	...	...	...	...
x_{m1}	x_{m2}	x_{m3}	...	...	x_{mq}

Fig. 2. Gene expression profile matrix.

Algorithm 2. In Algorithm 2 (Lines 1:3), K-means is applied on input gene expression profile ($GEP[m][q]$) to identify NC discrete values in each time slot or condition. In the second part of Algorithm 2 (Lines 4:8), we enumerate on columns of $GEP[m][q]$ matrix and map each expression level x_{ab} to the closest centroid of column b, ($QGEP[m][q]$).

Algorithm 2. Time-slot value quantization

Input: GEP[m][q]: Matrix of gene expression profile, m: Number of disease type genes, q:Length of sequence, NC = Number of clusters
Output: QGEP[m][q]: Matrix of quantized gene expression profile, Centers[NC][q]: Centroid of clusters per time slot
1: **for** h = 1 to q **do**
2: Centers[∗][h] = K-means(GEP[∗][h], NC)
3: **Endfor**
4: **for** b = 1 to q
5: **for** a = 1 to m
6: QGEP[a][b] = Map to closest
7: centroid(Centers[∗][b])
8: **Endfor**
9: **Endfor**

3.3. Mapping quantized gene expression profiles

Since each disease type consists of several diseases, each disease appears all clusters of its corresponding disease type. So gene expression profiles which are quantized for each disease genes are mapped to their corresponding disease type clusters. Finally, in each disease, we have multiple clusters (TSC_{ij}) of quantized gene expression profiles.

3.4. Learning HMM models

In learning step, we consider each gene expression profile as an effective sequence that could be interpreted as an observations sequence in HMM. So we learn a series of HMM models, (λ_{ij}), one model for each cluster j of each disease D_i which are included in the corresponding disease type (DT_k). An important parameter in learning HMM models is the number of states (N). In order to determine the right number of states, several HMM models with various number of states ranging from two to the length of gene expression sequence are learned. After learning models for different number of states, we cross-validate each HMM model (λ_{ij}) and choose the one that has the best recall, precision and F_1 values.

We fit an ergodic HMM over the training data of each cluster (TSC_{ij}), using Baum-Welch learning algorithm [29]. The algorithm is an iteration procedure composed of two steps: E-step and M-step. Until the HMM learning parameters, (Π^k,A^k,B^k) converge, i.e., $P(O|\lambda^k)$ never decreased. At each iteration k, first in the E-step, all $\xi_t(i,j)$ are computed. $\xi_t(i,j)$ is the probability of being in state S_i at time t and transit to

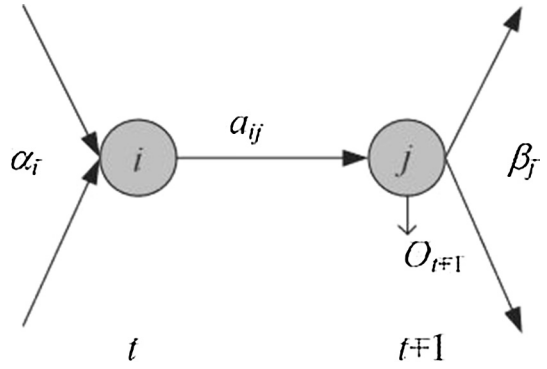


Fig. 3. Computation of arc probabilities, $\xi_t(i,j)$ [29].

state S_j at time $t+1$ (Fig. 3), given the whole observation sequence O^t and λ. $\xi_t(i,j)$ are computed by relation (8).

$$\xi_t(i,j) \equiv P(q_t = S_i, q_{t+1} = S_j | O, \lambda) \tag{8}$$

$\gamma_t(i)$ are computed for all states i, given current $\lambda = (\Pi^k, A^k, B^k)$ according to relation (9).

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \tag{9}$$

Next, in the M-step, $\lambda(\Pi^k, A^k, B^k)$ is recalculated by considering all observation $O = \{O^1, O^2, ..., O^m\}$ using $\xi_t(i,j)$ and $\gamma_t(i)$ for all i,j from the last E-step by relations 10, 11, 12.

$$\pi_i = \frac{\sum_{k=1}^{m} \gamma_t^k(i)}{m} \tag{10}$$

$$a_{ij} = \frac{\sum_{k=1}^{m} \sum_{t=1}^{T_k - 1} \xi_t^k(i,j)}{m} \tag{11}$$

$$b_j(m) = \frac{\sum_{k=1}^{m} \sum_{t=1}^{T_k} \gamma_t^k(i) 1(O_t^k = v_m)}{\sum_{k=1}^{m} \sum_{t=1}^{T_k} \gamma_t^k(i)} \tag{12}$$

After learning HMM models, λ_{ij}, for clusters j of disease i their corresponding threshold (t_{ij}) are calculated by relation (13).

$$Threshold_{ij} = \frac{\sum_{l=1}^{n_{ij}} prob[l]}{n_{ij}}, n_{ij} = S_{ij} * 0.25 \tag{13}$$

where $prob_{ij}[l]$ is a vector with length S_{ij} that contains probability of every gene expression sequence in the training set which are sorted out in the ascending order and S_{ij} is the size of training set of cluster (TSC_{ij}).

3.5. Negative gene extraction

Since we want to compare the proposed method with other methods, in terms of recall, precision and F_1 metrics, we need some negative samples in the prediction phase. In order to extract a set of reliable negative genes (*RN*) from unlabeled genes (*U*), we follow a simple approach in which some unlabeled samples with the most dissimilarity to all positive samples are labeled as reliable negative. In other words, extracted *RN* samples are outliers with respect to positive samples.

Suppose $G_P(x_1, x_2, x_3, ..., x_m)$ and $G_U(y_1, y_2, y_3, ..., y_m)$ are gene expression profiles before quantization of a disease gene and an unlabeled gene, respectively. We consider the dissimilarity between G_P and G_U as the Euclidean distance between their corresponding gene expression profiles which is calculated by relation (14):

$$dis(G_P, G_U) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2} \tag{14}$$

By using relation (15), we calculate the average Euclidean distance of an unlabeled gene, G_U from all disease genes ($(G_P)_j$).

$$Avgdis[G_U] = \frac{\sum_{j=1}^{n} dis((G_P)_j, G_U)}{n} \tag{15}$$

where n is the number of disease genes (i.e., positive genes). Finally, farthest unlabeled genes from all disease genes are selected according to relation (16).

$$RN = \{G_U | Avgdis[G_U] > \theta\} \tag{16}$$

3.6. Disease candidate genes prediction

In this phase, the extracted reliable negative genes are removed from unlabeled genes ($UL = U - RN$). Then, the remaining unlabeled genes expression profiles are quantized according to the values of quantization (*Centers*) which are identified in the second step of learning phase. After that, the label of each unlabeled gene (TSC_{ul}) is predicted by using all HMM_{ij} models which are learned for all clusters (TSC_{ij}) based on the best number of states and its corresponding threshold in each cluster (λ_{ij}, t_{ij}). If at least one of the HMM models predict a unlabeled gene as positive, we consider that gene as a candidate disease gene. Otherwise it is considered as negative (non disease).

4. Experimental results

In this Section, first the experimental results of the proposed method per disease (D_i) are discussed. Furthermore, the influence of disease type genes clustering and number of clusters in gene expression profile quantization in predicting disease candidate genes are explored. Moreover, the results of the proposed method per disease type (DT_k) are compared with other methods. All computations are conducted on one Xeon CPU with 16 cores and 16 GB of RAM.

4.1. Datasets

The dataset provided by Yang et al. [18] is used in this study. All genes are extracted by combining GENECARD [30] and OMIM [31] disease genes data. We use 398 known disease type genes and 12001 unlabeled genes in three disease types: cancer, endocrine and cardiovascular (see Table 1).

GO data (release of November 2015) and a human gene annotation dataset (release of November 2015) were downloaded from the GO database [32].

Another dataset that is used in this paper is the microarray gene expression profiles (GEP). GEP specifications for each disease are shown in Table 2. All datasets which are mentioned in Table 2 are downloaded from National Center for Biotechnology Information (NBCI) [33].

Recall that the previous research use the datasets containing the genes of disease types until the year 2010 [30] to learn and examine their performance. But, unlike the previous methods, the proposed method is learned and evaluated on the datasets by year 2010 [30], and further evaluated by a set of new disease genes identified between years 2010 till 2016. The new confirmed disease genes are obtained from GENECARDS [34].

Table 1
Number of disease genes per disease type and unlabeled genes.

Label of genes	Disease type	No. of genes
Positive	Cancer	210
	Endocrine	81
	Cardiovascular	107
Unlabeled	–	12,001

Table 2
Microarray gene expression profiles of datasets specification.

Disease type	Disease	GEP dataset	No. of expressions per gene
Cancer	Prostate	GDS5805	6
	Lung	GDS1204	18
	Colorectal	GDS5029	18
Cardiovascular	Heart Failure	GDS651	37
Endocrine	Adrenal	GDS3556	9

4.2. Evaluation metrics

There are several metrics to measure the performance of disease candidate genes prediction methods such as precision, recall and F_1 metrics. These metrics are defined in relations 17, 18, 19, respectively.

$$Precision(P) = \frac{TP}{TP + FP} \tag{17}$$

$$Recall(R) = \frac{TP}{TP + FN} \tag{18}$$

$$F_1 = \frac{2*Precision*Recall}{Precision + Recall} \tag{19}$$

where TP[5] is the number of disease genes which are properly identified, FP[6] is the number of reliable negative genes which are identified as disease genes and FN[7] is the number of disease genes which are identified as reliable negative genes. Since in prediction the disease candidate genes problem, we are given just disease genes, the recall metric is the most important one. In other words, it represents the percentage of correctly identified disease genes.

4.3. Results

In this Section, the results of applying the proposed method on the dataset described in Section 4.1 are represented.

4.3.1. Disease type genes clustering results

In order to determine an appropriate number of clusters (L) in each disease type, the *TotalDensity* metric (relation (6)) is calculated for various numbers of clusters per disease type. The results are represented in Table 3. For instance, in cancer disease type, when the number of clusters is equal to two, we have a lowest value in terms of *TotalDensity*.

The reason behind the genes of each disease type clustering is that these genes could have various patterns. To explore the effectiveness of disease type clustering in predicting disease candidate genes, we learned another series of HMM models, one for each disease type (without clustering). Finally, we compared the results of predicting disease genes based on HMMs learned on with and without clustering. As can be seen in Figs. 4 and 5, disease type clustering significantly improves the values of precision, recall and F_1 metrics.

4.3.2. Gene expression profile quantization results

Before starting learning HMM models an other important parameter that should be determined is the number of discrete points in gene expression quantization (i.e., *NC* in Algorithm 2). So we quantize the gene expression profiles by applying Algorithm 2 considering various numbers of discrete points. Fig. 6 shows that by increasing the number of discrete points, the recall metric, which is the most important metric in this context, is improved. As can be seen, the recall values of HMM

[5] True Positive.
[6] False Positive.
[7] False Negative.

Table 3
Identifying the number of clusters with minimum *TotalDensity* per disease type.

Disease type	No. of disease genes	number of clusters	*TotalDensity*	The number of clusters with minimum *TotalDensity*
Cancer	210	2	**1054.12**	2
		3	1523.24	
		4	1793.56	
Cardiovascular	107	2	419.67	3
		3	**300.7**	
		4	350.67	
Endocrine	81	2	**297.28**	2
		3	307.08	

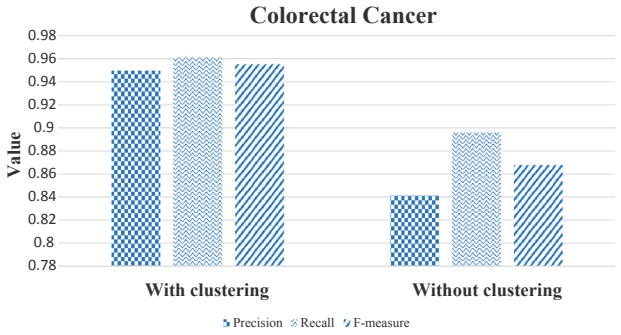


Fig. 4. The effect of disease type clustering on predicting the colorectal cancer disease.

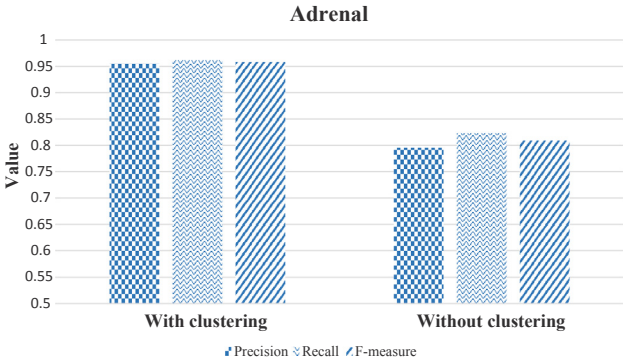


Fig. 5. The effect of disease type clustering on predicting the adrenal disease.

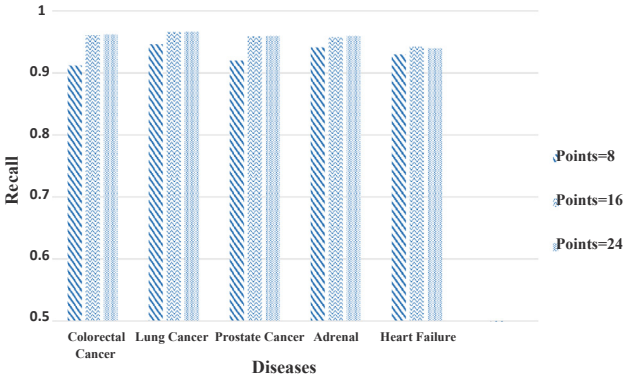


Fig. 6. Comparison of disease candidate genes prediction results for different number of discreate points (*NC*) in gene expression quantization.

models with 8 discrete points in quantizing gene expression profiles are lower than the recall values of HMM models with 16 and 24 clusters. It is also clear that the recall values of HMM models with 16 and 24 discrete points are almost the same, therefore, for sake of simplicity $NC = 16$ is chosen.

4.3.3. Learning and predicting results per disease

We employ 10-fold cross validation technique to evaluate the performance of the proposed method. Using this technique, the dataset is divided into 10 subsets, and the proposed method is repeated 10 times. Each time, one of the 10 subsets is used as the test set and the other subsets are considered as a training set. Then the average of precision, recall and F_1 metrics across all 10 trials are computed.

We learn a HMM model (λ_{ij}) for each cluster j of disease i. The parameters of all $\lambda = (\Pi, A, B)$ models are initialized as follows:

- State transition probabilities (A)
  $A = [a_{ij}]$ where $a_{ij} = \frac{1}{N}$
- Observation probabilities (B)
  $B = [b_j(m)]$ where $b_j(m) = \frac{1}{M}$
- Initial state probabilities (Π)
  $\Pi = [\pi_i]$ where $\sum_{i=1}^{n} \pi_i = 1$, π_i are randomly selected.

In order to determine the number of states per disease, we learn some HMM models with different number of states on each disease. Then, the number of states corresponding to the learned HMM models with maximum recall, precision and F_1 values is chosen. The best state numbers for all diseases are presented in Table 4.

After λ_{ij} initialization, we apply the proposed method on all clusters in each disease (TSC_{ij}) and learn their corresponding λ_{ij} using 10-fold cross validation technique. After that, the threshold (t_{ij}) for each cluster by using corresponding HMM model (λ_{ij}) is determined according to relation (13).

The proposed method is evaluated with different number of reliable negative genes. At first, we set the number of reliable negative genes per disease equal to the number of corresponding disease type genes. For instance, in prostate cancer, the number of reliable negative genes is set to 210 genes. So the values of precision, recall and F_1 are 41.3%, 95.7% and 57.69%. The results of all diseases are represented in Table 5.

After that, the number of reliable negative genes is set lower than the number of disease genes. Table 6 presents the results and comparison to Table 5, Table 6 shows a high rise of precision, recall and F_1 values. So it confirms our hypothesis about the number of reliable negative genes, unlabeled genes having more distance from disease genes are more reliable to be considered as negative genes.

Another parameter that affect learning HMM models is the number of iterations in applying the Baum–Weltch algorithm. So the impact of various iteration numbers on learning HMM models is explored. For instance, in colorectal cancer, with the iteration number equal to 150, recall value is higher than other iteration numbers. In heart failure, the iteration numbers equal to 80, 100 and 150 have the best recall, so the iteration number 80 is enough. Figs. 7 and 8 show that by increasing the iteration number, the values of recall, precision and F_1 are increased, in other words they are converged.

Table 4
Best number of states in learned HMM with maximum F_1 value.

Disease name	Best No. of States
Prostate Cancer	5
Lung Cancer	9
Colorectal Cancer	13
Heart Failure	25
Adrenal	4

Table 5
Evaluation results when the number of reliable negative genes are equal to the number of disease genes.

Disease	No. of disease genes	No. of negative genes	P	R	F_1
Prostate Cancer	210	210	41.3	95.7	57.69
Colorectal Cancer	210	210	69.2	99.0	81.46
Lung Cancer	210	210	65.1	97.9	78.19
Heart Failure	107	107	17.0	100	29.0
Adrenal	81	81	95.0	96.0	95.0

Table 6
Evaluation results when the number of reliable negative genes are less than the disease genes.

Disease	No. of disease genes	No. of negative genes	P	R	F_1
Prostate Cancer	210	95	92	95	93
Colorectal Cancer	210	191	93	94	93
Lung Cancer	210	150	91.1	95	93.3
Heart Failure	107	89	91.87	94.23	93.03
Adrenal	81	81	94.79	99.47	97.07

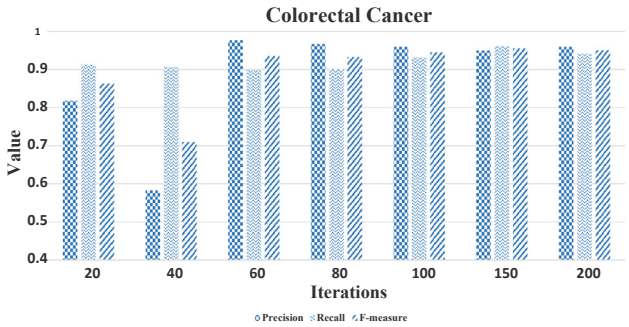


Fig. 7. Comparison of the evaluation metrics for different iteration numbers when learning HMM models in colorectal cancer.

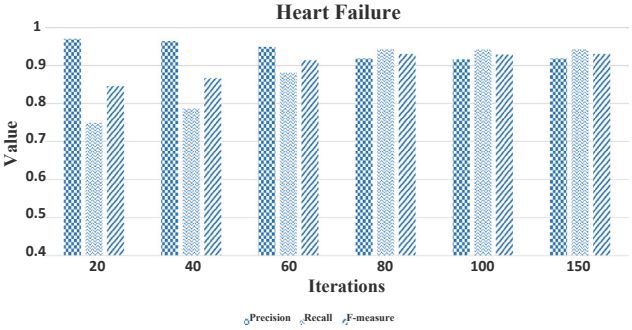


Fig. 8. Comparison of the evaluation metrics for different iteration numbers when learning HMM models in heart failure.

4.4. Evaluation of the proposed method with new disease genes

Although many methods are proposed to predict disease genes, all of them have never been evaluated on new disease genes (disease genes identified and announced by biomedical researchers between 2010 and 2016 years). Thus, three diseases are selected, namely, prostate cancer, colorectal cancer and adrenal, as detailed case studies. The HMM models (λ_{ij}) are learned by using disease genes of year 2010. Also the thresholds (t_{ij}) are calculated by using training set. In this evaluation, new disease genes are treated as unlabeled genes in prediction phase. For instance, in prostate cancer, there are 191 disease genes are

Table 7
The quantitative evaluation of predicted disease candidate genes identified by the proposed method across years 2010 and 2016 datasets.

Disease	No. of new disease genes	TP	FN	R
Prostate Cancer	191	178	13	93.19
Colorectal Cancer	240	229	11	95.41
Adrenal	9	9	0	100

Table 8
The overlap between identified reliable negative genes based on disease genes (version 2010) and new disease genes set (version 2016) by the proposed method.

Disease	No. of RN genes (2010)	No. of overlap with new disease genes (2016)	Percent (%)
Adrenal	81	0	0
Colorectal Cancer	191	6	3.14
Prostate Cancer	95	3	3.15

identified after the year 2010 which do not exist in other studies. By applying the proposed method on new disease genes, in prostate cancer the learned HMM models predict 178 of 191 genes as disease candidate genes and only 13 genes are labeled as negative genes, so the recall value is 93.19%. In colorectal cancer, 229 of 240 genes are labeled as disease candidate genes and lastly, in adrenal disease, the proposed method is predicted all new disease genes as disease candidate genes. All detailed results are presented in Table 7.

Another experiment is conducted to calculate the overlap between identified reliable negative genes based on disease genes dataset of 2010 and new disease genes dataset identified during the period from 2010 and 2016 years. Table 8 represents the results. The absence of reliable negative genes in the new disease genes dataset shows that the reliable negative genes are chosen correctly from unlabeled genes of 2010 dataset by the proposed method. So the proposed method, has done its job prefect. For instance, in prostate cancer, only 3 of 95 reliable negative genes (3.15%) exist in new disease genes set and the other 92 genes are still non disease genes.

4.5. Comparison with other works per disease type

The proposed method is compared with seven other methods, namely, Xu's [23], Smalter's [21], ProDiGe [16], PUDI [17], EPU [18], SFM [19] and PEGPUL [20]. The results of Xu's [23], Smalter's [21], ProDiGe [16], PUDI [17] methods are extracted from EPU [18] research. The results of aforementioned methods except PEGPUL method [20] are reported based on disease types. In the proposed method, the results of each disease type are calculated as the average of precision, recall and F_1 values of its corresponding gene diseases.

Among six mentioned methods, the proposed method appears to be the most robust method, since it yields greater recall, precision and F_1 values compared to the other methods for all disease types.

In cancer disease type, Fig. 9 shows that the proposed method outperforms the other methods and has done 12.7%, 11.7% and 12.4% better than EPU method in (the best method of previous methods) terms of precision, recall and F_1 metrics, respectively.

In cardiovascular disease type according to Fig. 10, the proposed method in terms of precision and F_1 metrics is better than EPU [18]. But in recall metric, ProDiGe method is 2.07% higher than the proposed method.

In endocrine disease type, according to Fig. 11 the proposed method increased the values of precision (6.8%), recall (8.0%) and F_1 (7.4%) in comparison to EPU [18].

The results of the PEGPUL method [20] are reported without considering disease types (i.e., disease/non-disease). Therefore, the average of the proposed method in three disease types are compared with PEGPUL results that are shown in Table 9. As can be seen, the proposed method in terms of precision, recall and F_1 metrics is 17.79%, 7.05% and 13.87% better than PEGPUL method [20].

5. Analysis and discussion

The absence of negative set (non disease genes) is a challenging issue in predicting disease candidate genes by binary machine learning techniques. However, there exist many methods to predict disease candidate genes which most of them use a binary classification technique. These methods depend on the set of reliable negative genes extracted from unlabeled genes. If the set of reliable negative genes contains unknown disease genes, the learned model based on binary classification will not perform correctly. The most important features of the proposed method are:

1. Utilizing HMM models which are one-class classifiers.
2. Considering gene expression sequence instead of a single gene expression value.

The HMM models are learned based on only disease genes (i.e., positive samples), therefore the set of reliable negative genes will not

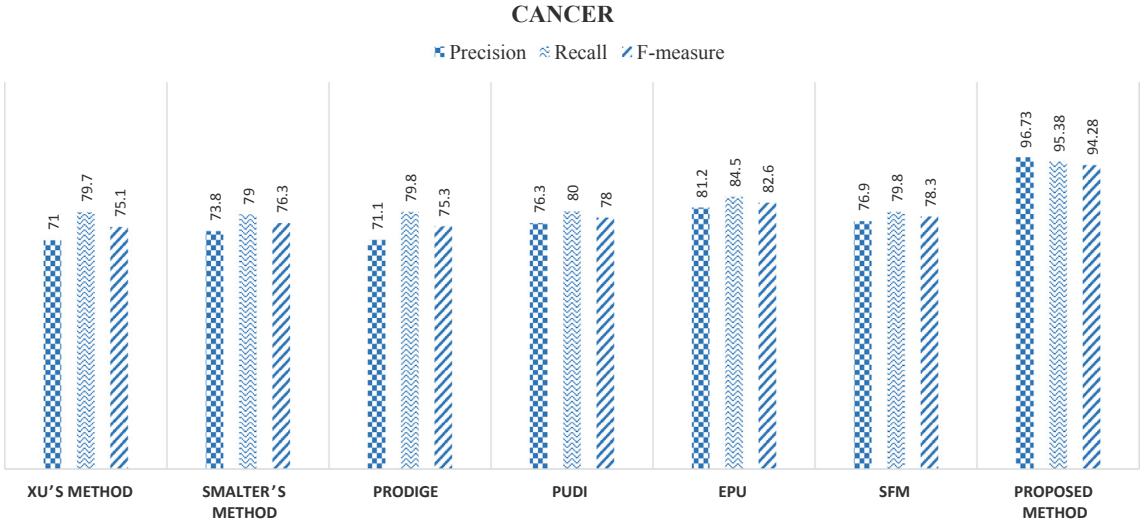


Fig. 9. Comparison of cancer disease candidate genes prediction methods.

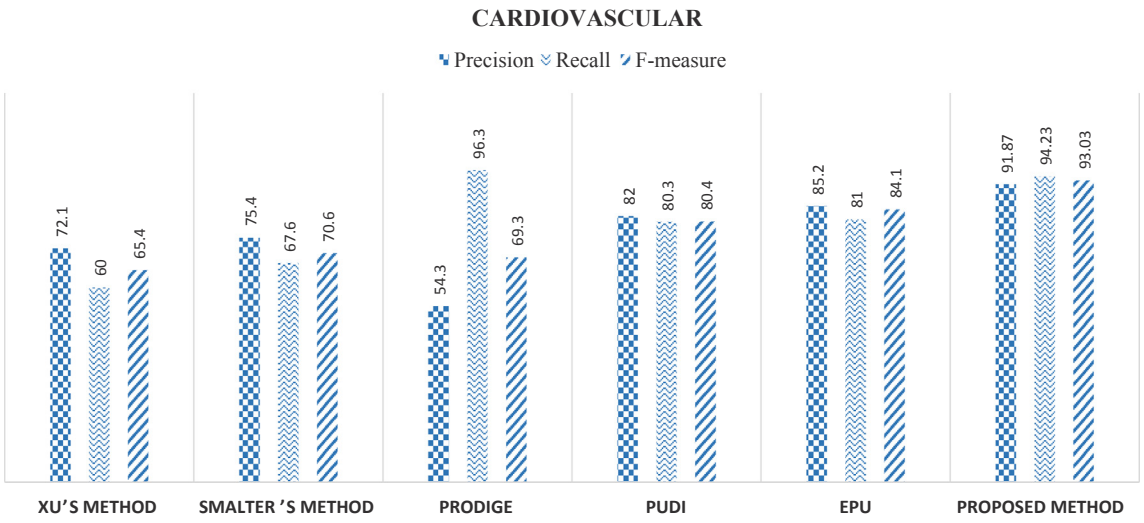


Fig. 10. Comparison of cardiovascular disease candidate genes prediction methods.

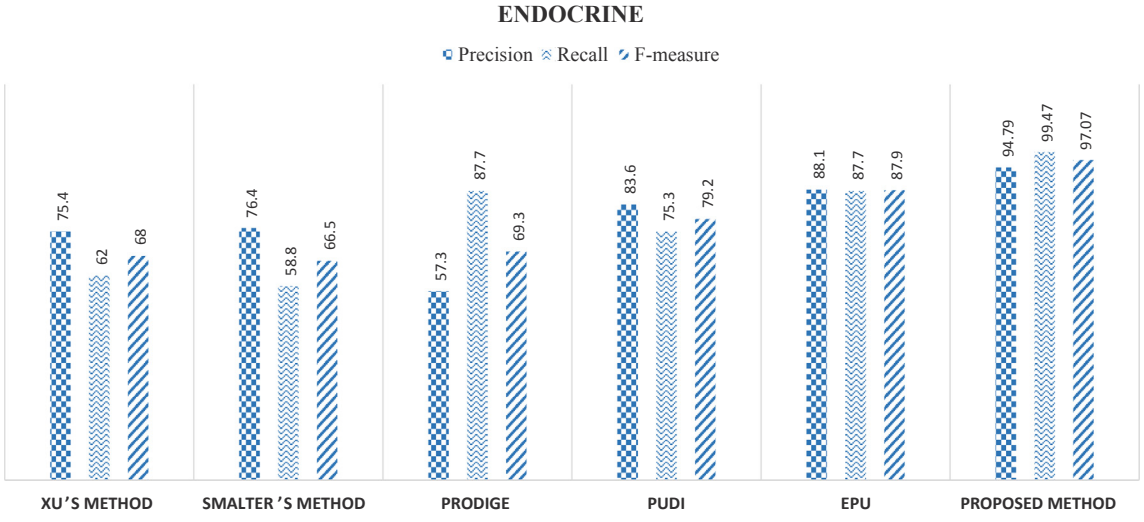


Fig. 11. Comparison of endocrine disease candidate genes prediction methods.

Table 9
Comparison between the proposed method and PEGPUL method [20] in terms of disease candidate genes prediction.

Method	P	R	F_1
The proposed method	**94.46**	**96.36**	**94.79**
PEGPUL	76.67	89.31	82.49

participate in the learning HMM models.

By considering the gene expression profiles as observation sequences, HMM models are the best choice for modeling them. Moreover, by clustering disease genes and learning a HMM model for each cluster, the prediction accuracy increases drastically. In order to discriminate disease candidate genes from non disease genes and minimizing the FN, thresholds are calculated based on training set of clusters in each disease.

Finally, the effectiveness of the proposed method is evaluated using new disease genes identified between 2010 and 2016 years. The results are shown in Table 7. As can be seen, the accuracy of the proposed method is high which we can trust.

6. Conclusion

In this paper, we proposed a new disease candidate genes prediction method based on hidden Markov model and gene expression sequences. The experimental results indicated the effectiveness of designing a model for disease candidate genes prediction problem using a one class classification method, HMM. Since the proposed method did not use any reliable negative sample set in the learning phase, the effectiveness of disease candidate genes prediction is improved. In order to compare the proposed method with other methods, a set of reliable genes which is extracted based on Euclidean distance from unlabeled genes is used only in the prediction phase. For improving the accuracy of predicting disease candidate genes and considering different patterns of disease type genes, we partitioned disease type genes. Then for each cluster, a HMM is learned.

Knowledge of which genes cause which disorders will simplify diagnosis of patients and using this knowledge leads to discover new drugs to tackle disease genes associated with a specific disease [35]. Also, the proposed method play a significant role in gene selection methods in various biomedical problems [36–38]. In the future, we apply the proposed method on other diseases. Although we can employ some other genomic information to further improve the proposed method, such as PPI network.

Conflict of interest

The authors declare that they have no conflict of interest.

References

[1] T. Nguyen, A. Khosravi, D. Creighton, S. Nahavandi, Hidden markov models for cancer classification using gene expression profiles, Inf. Sci. 316 (2015) 293–307.
[2] P. Maji, E. Shah, S. Paul, RelSim: an integrated method to identify disease genes using gene expression profiles and PPIN based similarity measure, Inf. Sci. 384 (2017) 110–125.
[3] E.A. Adie, R.R. Adams, K.L. Evans, D.J. Porteous, B.S. Pickard, Speeding disease gene discovery by sequence based candidate prioritization, BMC Bioinformatics 6 (1) (2005) 1.
[4] S.J. Furney, B. Calvo, P. Larranaga, J.A. Lozano, N. Lopez-Bigas, Prioritization of candidate cancer genes-an aid to oncogenomic studies, Nucleic Acids Res. 36 (18) (2008) e115.
[5] J. Freudenberg, P. Propping, A similarity-based method for genome-wide prediction of disease-relevant human genes, Bioinformatics 18 (suppl 2) (2002) S110–S115.
[6] F.S. Turner, D.R. Clutterbuck, C.A. Semple, Pocus: mining genomic sequence annotation to predict disease genes, Genome Biol. 4 (11) (2003) 1.
[7] P. Zhang, J. Zhang, H. Sheng, J.J. Russo, B. Osborne, K. Buetow, Gene functional similarity search tool (GFSST), BMC Bioinformatics 7 (1) (2006) 1.
[8] D. Shriner, T.M. Baye, M.A. Padilla, S. Zhang, L.K. Vaughan, A.E. Loraine, Commonality of functional annotation: a method for prioritization of candidate genes from genome-wide linkage studies, Nucleic Acids Res. 36 (4) (2008) e26.
[9] Y. Li, J.C. Patra, Integration of multiple data sources to prioritize candidate genes using discounted rating system, BMC Bioinformatics 11 (1) (2010) 1.
[10] J. Chen, B.J. Aronow, A.G. Jegga, Disease candidate gene identification and prioritization using protein interaction networks, BMC Bioinformatics 10 (1) (2009) 1.
[11] J. Zhao, T.-H. Yang, Y. Huang, P. Holme, Ranking candidate disease genes from gene expression and protein interaction: a Katz-centrality based approach, PloS One 6 (9) (2011) e24306.
[12] C. Wu, J. Zhu, X. Zhang, Integrating gene expression and protein-protein interaction network to prioritize cancer-associated genes, BMC Bioinformatics 13 (1) (2012) 182.
[13] J. Li, L. Wang, M. Guo, R. Zhang, Q. Dai, X. Liu, C. Wang, Z. Teng, P. Xuan, M. Zhang, Mining disease genes using integrated protein–protein interaction and gene–gene co-regulation information, FEBS Open Bio 5 (1) (2015) 251–256.
[14] B. Liu, M. Jin, P. Zeng, Prioritization of candidate disease genes by combining topological similarity and semantic similarity, J. Biomed. Informat. 57 (2015) 1–5.
[15] L. Cerulo, C. Elkan, M. Ceccarelli, Learning gene regulatory networks from only positive and unlabeled data, BMC Bioinformatics 11 (1) (2010) 263–270.
[16] F. Mordelet, J.-P. Vert, ProDiGe: prioritization of disease genes with multitask machine learning from positive and unlabeled examples, BMC Bioinformatics 12 (1) (2011) 1.
[17] P. Yang, X.-L. Li, J.-P. Mei, C.-K. Kwoh, S.-K. Ng, Positive-unlabeled learning for disease gene identification, Bioinformatics 28 (20) (2012) 2640–2647.
[18] P. Yang, X. Li, H.-N. Chua, C.-K. Kwoh, S.-K. Ng, Ensemble positive unlabeled learning for disease gene identification, PloS One 9 (5) (2014) e97079.
[19] A. Yousef, N.M. Charkari, SFM: a novel sequence-based fusion method for disease genes identification and prioritization, J. Theor. Biol. 383 (2015) 12–19.
[20] G.-H. Jowkar, E.G. Mansoori, Perceptron ensemble of graph-based positive-unlabeled learning for disease gene identification, Comput. Biol. Chem. 64 (2016) 263–270.
[21] A. Smalter, S.F. Lei, X.-W. Chen, Human disease-gene classification with integrative sequence-based and topological features of protein-protein interaction networks, in: The Proceedings of IEEE International Conference on Bioinformatics and Biomedicine, 2007, pp. 209–216.
[22] P. Radivojac, K. Peng, W.T. Clark, B.J. Peters, A. Mohan, S.M. Boyle, S.D. Mooney, An integrated approach to inferring gene–disease associations in humans, Proteins: Struct. Funct. Bioinf. 72 (3) (2008) 1030–1037.
[23] J. Xu, Y. Li, Discovering disease-genes by topological features in human protein–protein interaction network, Bioinformatics 22 (22) (2006) 2800–2805.
[24] H. Caniza, A.E. Romero, S. Heron, H. Yang, A. Devoto, M. Frasca, M. Mesiti, G. Valentini, A. Paccanaro, GOssTo: a stand-alone application and a web tool for calculating semantic similarities on the gene ontology, Bioinformatics 30 (15) (2014) 2235–2236.
[25] C. Pesquita, D. Faria, H. Bastos, A.E. Ferreira, A.O. Falcão, F.M. Couto, Metrics for go based protein semantic similarity: a systematic evaluation, BMC Bioinformatics 9 (5) (2008) 1.
[26] X. Chen, R. Yang, J. Xu, H. Ma, S. Chen, X. Bian, L. Liu, A sensitive method for computing go-based functional similarities among genes with shallow annotation, Gene 509 (1) (2012) 131–135.
[27] N. Seco, T. Veale, J. Hayes, An intrinsic information content metric for semantic similarity in WordNet, in: European Conference on Artificial Intelligence, vol. 16, 2004, pp. 1089.
[28] L.E. Baum, T. Petrie, Statistical inference for probabilistic functions of finite state markov chains, Ann. Math. Stat. 37 (1966) 1554–1563.
[29] E. Alpaydin, Introduction to Machine Learning, third ed., MIT Press, 2014.
[30] M. Safran, I. Dalah, J. Alexander, N. Rosen, T.I. Stein, M. Shmoish, N. Nativ, I. Bahir, T. Doniger, H. Krug, et al., GeneCards version 3: the human gene integrator, Database 2010 (2010) baq020.
[31] V.A. McKusick, Mendelian inheritance in Man and its online version, OMIM, Am. J. Hum. Genet. 80 (4) (2007) 588–604.
[32] Gene ontology consortium: going forward, Nucleic Acids Res. 43 (D1) (2015) D1049. Available: <https://doi.org/10.1093/nar/.gku1179/> .
[33] N.C. for Biotechnology Information (NCBI)[Internet], National Library of Medicine (US), National Center for Biotechnology Information, Bethesda (MD), 2015. Available: < https://www.ncbi.nlm.nih.gov/ > .
[34] D. Lancet et al., Genecards Human Gene Database, 2016. Available: < http://www.genecards.org/ > .
[35] U. Iqbal, T.-H. Chang, P.-A. Nguyen, S. Syed-Abdul, H.-C. Yang, C.-W. Huang, S. Atique, W.-C. Yang, M. Moldovan, W.-S. Jian, et al., Benzodiazepines use and breast cancer risk: a population-based study and gene expression profiling evidence, J. Biomed. Inform. 74 (2017) 85–91.
[36] V. Elyasigomari, D. Lee, H.R. Screen, M.H. Shaheed, Development of a two-stage gene selection method that incorporates a novel hybrid approach using the cuckoo optimization algorithm and harmony search for cancer classification, J. Biomed. Inform. 67 (2017) 11–20.
[37] Y. Chen, Z. Zhang, J. Zheng, Y. Ma, Y. Xue, Gene selection for tumor classification using neighborhood rough sets and entropy measures, J. Biomed. Inform. 67 (2017) 59–68.
[38] W. Li, L. Zhu, H. Huang, Y. He, J. Lv, W. Li, L. Chen, W. He, Identification of susceptible genes for complex chronic diseases based on disease risk functional SNPs and interaction networks, J. Biomed. Inform. 74 (2017) 137–144.