OPEN ACCESS Freely available online

PLOS ONE

Using Multi-Instance Hierarchical Clustering Learning System to Predict Yeast Gene Function

Bo Liao*, Yun Li, Yan Jiang, Lijun Cai*

College of Information Science and Engineering, Hunan University, Changsha, Hunan, China

Abstract

Time-course gene expression datasets, which record continuous biological processes of genes, have recently been used to predict gene function. However, only few positive genes can be obtained from annotation databases, such as gene ontology (GO). To obtain more useful information and effectively predict gene function, gene annotations are clustered together to form a learnable and effective learning system. In this paper, we propose a novel multi-instance hierarchical clustering (MIHC) method to establish a learning system by clustering GO and compare this method with other learning system establishment methods. Multi-label support vector machine classifier and multi-label K-nearest neighbor classifier are used to verify these methods in four yeast time-course gene expression datasets. The MIHC method shows good performance, which serves as a guide to annotators or refines the annotation in detail.

Citation: Liao B, Li Y, Jiang Y, Cai L (2014) Using Multi-Instance Hierarchical Clustering Learning System to Predict Yeast Gene Function. PLoS ONE 9(3): e90962. doi:10.1371/journal.pone.0090962

Editor: Holger Fröhlich, University of Bonn, Bonn-Aachen International Center for IT, Germany

Received October 15, 2013; **Accepted** February 5, 2014; **Published** March 12, 2014

Funding: This work is supported by the Program for New Century Excellent Talents in University (Grant NCET-10-0365), National Nature Science Foundation of China (Grant 60973082, 11171369, 61272395,61370171), National Nature Science Foundation of Hunan Province (Grant 12JJ2041), the Planned Science and Technology Project of Hunan Province (Grant 2009FJ3195, 2012FJ2012) and supported by the Fundamental Research Funds for the Central Universities, Hunan university. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: dragonbw@163.com (BL); in_ljcai@yeah.net (LC)

Introduction

Genes are annotated in gene annotation databases [e.g., gene ontology (GO), KEGG, and MIPS], but the rate of gene identification is faster than gene annotation. Given that large amounts of identified genes, predicting the functions for un-annotated genes is a challenge. To date, many effective machine learning techniques are proposed. However, function prediction is different from the common machine learning tasks. A gene may have multiple functions and the function belongs to a set of genes. Function prediction belongs to the multi-label learning (MLL) task, and the common machine learning task is a single-instance single-label learning. Therefore, establishing an effective and learnable learning system for learning machines is necessary.

In this study, different types of data have different learning approaches. We choose yeast time-course gene expression datasets because they record gene responses to various environments. Therefore, when searching for functions of a gene according to their involvement in biological processes, measurements of changes in gene expression throughout the time course of a given biological response are particularly interesting [1].

Gene function prediction method for different purposes can be grouped into supervised and unsupervised methods. Unsupervised methods (i.e., clustering) do not usually use existing biological knowledge to find gene expression patterns. Eisen et al. [2] discovered classes of expression patterns and identified groups of genes that are regulated similarly. Ernst et al. [3,4] clustered short time series gene expression data using a predefined expression model. Ma et al. [5] used a data-driven method to cluster time-course gene expression data. Other popular clustering algorithms include hierarchical clustering (HC), K-means clustering, and self-organizing maps [6]. Supervised methods (i.e., classification) use existing biological knowledge, such as GO, to create classification models. Lagreid et al. [1] applied the If-Then Rule Model to recognize the biological process from gene expression patterns. GENEFAS [7] predicted functions of un-annotated yeast genes using a functional association network based on annotated genes. Clare [8] presented a hierarchical multi-label classification (HMC) decision tree method to predict *Saccharomyces cerevisiae* gene functions. Schietgat et al. [9] presented an ensemble method (i.e., CLUS-HMC-ENS), which learns multi-tree for predicting gene functions of yeast. Kim et al. [10] combined the predictions of functional networks with predictions from a Naive Bayes classifier. Vazquez et al. [11] predicted global protein function from protein–protein interaction networks. Deng et al. [12] predicted gene functions with Markov random fields using protein interaction data. Nabieva et al. [13] proposed the functional flow method, which is a network-flow based algorithm, to predict protein function with few annotated neighbors. Recently, Magi et al. [14] annotated gene products using weighted functional networks. Liang et al. [15] predicted protein function using overlapping protein networks. Mitsakakis et al. [16] predicted *Drosophila melanogaster* gene function using the support vector machines (SVMs).

The present study predicts gene function based on the assumption that genes participating in the same biological processes have similar expression profiles. We initially produce a non-noise system by selecting genes. Then, the multi-instance hierarchical clustering (MIHC) method is proposed to establish a learning system. Finally, multi-label support vector machine (MLSVM) and multi-label K-nearest neighbor (MLKNN) classifiers are used to predict the function of genes in time-course

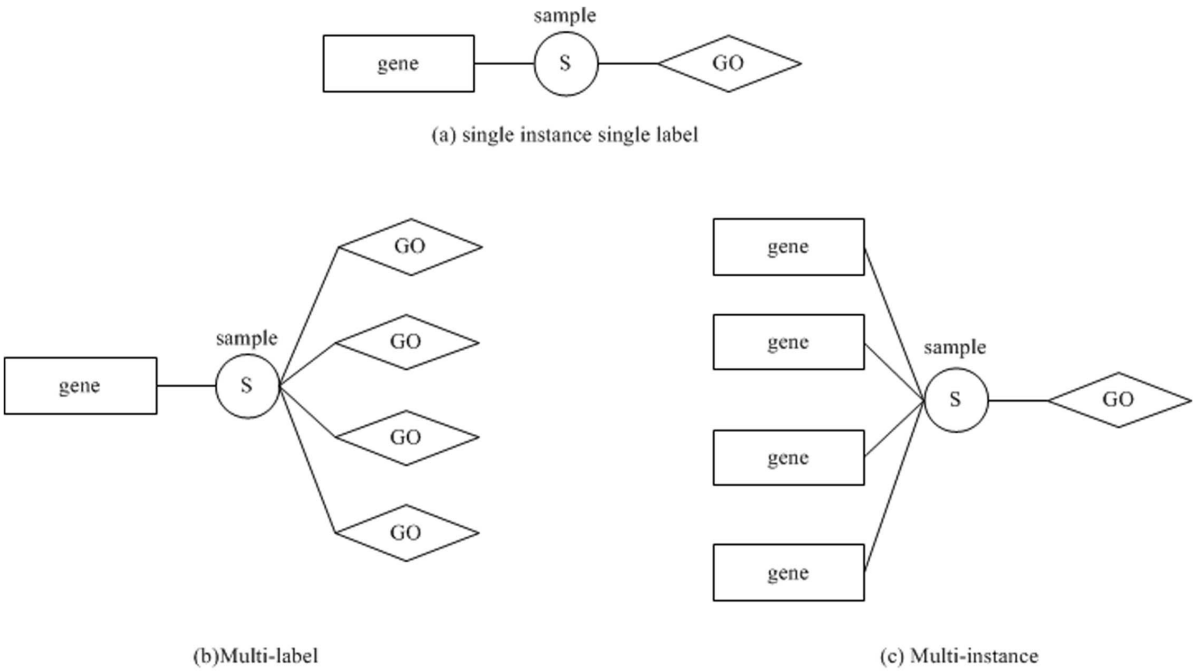

Figure 1. Three types of learning task. (a) A gene is treated as a sample and owns one GO term only, which is called the single instance single label. (b) A gene is treated as a sample and annotated by multiple GO terms. This relationship between gene and GO terms is multi-label. (c) Multiple genes are treated as samples and share the same GO term. The relationship between genes and GO term is called multi-instance.
doi:10.1371/journal.pone.0090962.g001

expression profile. The experiment proves the feasibility and efficiency of the proposed method.

Materials and Methods

Gene function prediction

In the GO database, the GO terms are organized as a directed acyclic graph (DAG). In the GO hierarchical structure, the genes are annotated at various levels of abstraction. When genes are annotated with the GO terms, the genes are annotated with the highest possible level of details, which corresponds to the lowest level of abstraction [17]. Therefore, the goal of gene function prediction is for the annotators to annotate genes with the highest level GO terms. However, we can only obtain extremely few positive genes with similar GO terms, and little information is available for a machine learning system. To obtain more positive genes and efficiently predict gene function, many researchers up-

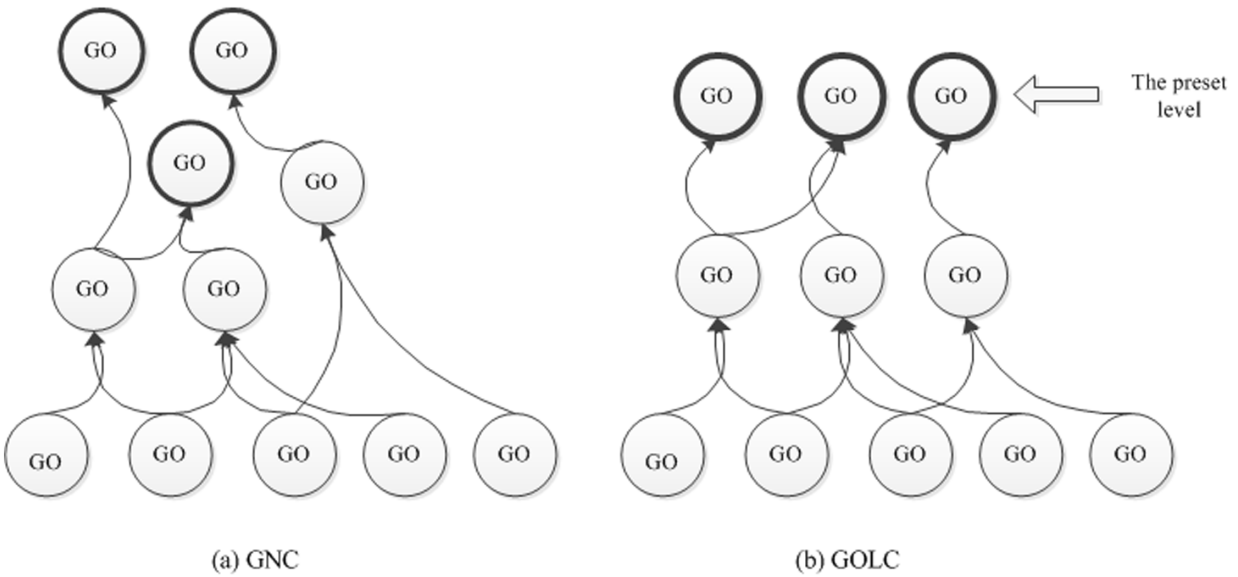

Figure 2. GO in the last level up-propagate along GO DAG. (a) The bold GO terms all own at least λ genes. (b) The bold GO terms are in the ℓ level of GO DAG.
doi:10.1371/journal.pone.0090962.g002

propagate gene annotation along a GO hierarchical structure and establish a learning system. The up-propagation approach can substantially group the following two methods: cluster genes to a certain GO level [8,9] and to a certain number [18].

Multi-instance learning (MIL) and MLL

Zhou et al. [19] provided a detailed description on MIL and MLL. MIL and MLL are used to learn the function of $f_{MIL}: 2^I \to \{-1,+1\}$ from the datasets $\{(X_i,y_i)|X_i \subseteq I, y_i \in \{-1,+1\}\}$ and $I=\{x_i|i=1,...,n\}$, and $f_{MLL}: I \to 2^L$ from the datasets $\{(x_i,Y_i)|x_i \in I, Y_i \subseteq L\}$ and $L=\{y_i|i=1,...,n\}$, respectively.

The relationships between genes and annotations are found in the GO database (Figure 1). Figure 1(b) shows that a gene is annotated by multiple GO terms, and Figure 1(c) shows that the genes are treated as instances of the sample with the annotation of GO. This GO term can be represented by those genes. Therefore, the relationships between gene and GO shown in Figures 1(b) and 1(c) are called multi-label and multi-instance, respectively.

MLSVM

SVM is an effective machine learning method. For classification problems, SVM implements a large margin classifier by solving a quadratic optimization program on the basis of the principle of structural risk minimization. Li et al. [20] adjusted the SVM to multi-label classification by improving the quadratic optimization program. Suppose (x_i, Y_i) is a training sample, where x_i is the feature vector and Y_i is the sample label. Let $\psi(x_i,y)=1$ if $y \in Y_i$ and $\psi(x_i,y)=-1$, otherwise the SVM classification problem model is described by the following optimization problem:

$$\begin{aligned} &\min_{w_y,b_y,\xi_{iy}} \frac{1}{2}\|w_y\|^2 + C\sum_{i=1}^{m} \xi_{iy}\tau_{iy} \\ &s.t. \quad \psi(x_i,y)\left(\left(w_y \cdot \varphi(x_i)\right)+b_y\right) \geq 1-\xi_{iy},\ i=1,...,m \\ &\xi_{iy} \geq 0, i=1,...,m \end{aligned} \tag{1}$$

where $w_y \cdot \varphi(x_i)$ is the inner product, $\varphi(x_i)$ is the function that maps x_i to a higher dimensional space $\mathcal{H}$, w_y and b_y are the parameters for representing a linear discriminant function in $\mathcal{H}$, ξ_{iy} is the non-negative slack variable introduced in the constraints to permit some training samples to be misclassified, C is the parameter to trade off the model complexity, and τ_{iy} is the amplification coefficient of the loss ξ_{iy} for handing the class imbalance problem [20,21].

Input: D-yeast expression dataset
G-annotated gene set
L-GO term set of G
UG-un-annotated gene set
Output: Y-predicted annotation set of UG
Data process:
1 Let the first value of D be always equal to 0, smooth out spikes, and obtain D'.
2 Using G, L, and the Pearson correlation of D', establish the non-noise system
$S = \{(G_i, GO_i) \mid i = 1,...,M\}$.
Establish learning system using MIHC
TS = MIHC(S):
3 Initialize the label set $GOTerms = \{GO_i \mid GO_i \in S\}$
4 **For** $GO_i, GO_j \in GOTerms$
5 Calculate $D(GO_i, GO_j)$, $D(GO_i) = \overline{Corr(G_i)}$, and $D(GO_j) = \overline{Corr(G_j)}$ using Eqs. (5) and (6)
6 **If** $D(GO_i, GO_j) \leq \max\left(D(GO_i), D(GO_j)\right)$ then
7 $G_{new} = G_i \cup G_j$
8 Add (G_{new}, GO_{new}) to S
9 Remove (G_i, GO_i) and (G_j, GO_j) from S
10 **End**
11 If S no longer changes, return to S
12 **End**
Prediction function
13 Y = MLSVM(TS, UG) or Y = MLKNN(TS, UG)

Figure 3. MIHC algorithm and flow chart of function prediction.
doi:10.1371/journal.pone.0090962.g003

Table 1. Number of genes and classes in each learning system.

		Gene Number				Class Number			
Method	**Parameters**	**alpha**	**cdc15**	**cdc28**	**elution**	**alpha**	**cdc15**	**cdc28**	**elution**
GNC	$\lambda = 10$	1213	1284	224	2089	190	204	27	281
	$\lambda = 20$	1216	1294	221	2050	119	129	19	166
	$\lambda = 30$	1215	1297	236	2027	84	89	18	122
	$\lambda = 40$	1180	1267	204	2022	56	61	10	99
	$\lambda = 50$	1205	1207	183	2038	51	42	4	80
GOLC	$\iota = 1$	1334	1417	261	2269	16	15	13	15
	$\iota = 2$	1332	1417	261	2269	49	44	29	51
	$\iota = 3$	1324	1407	261	2263	177	182	85	209
	$\iota = 4$	1278	1341	241	2179	335	342	106	381
MIHC	null	1334	1417	261	2269	74	61	24	29

doi:10.1371/journal.pone.0090962.t001

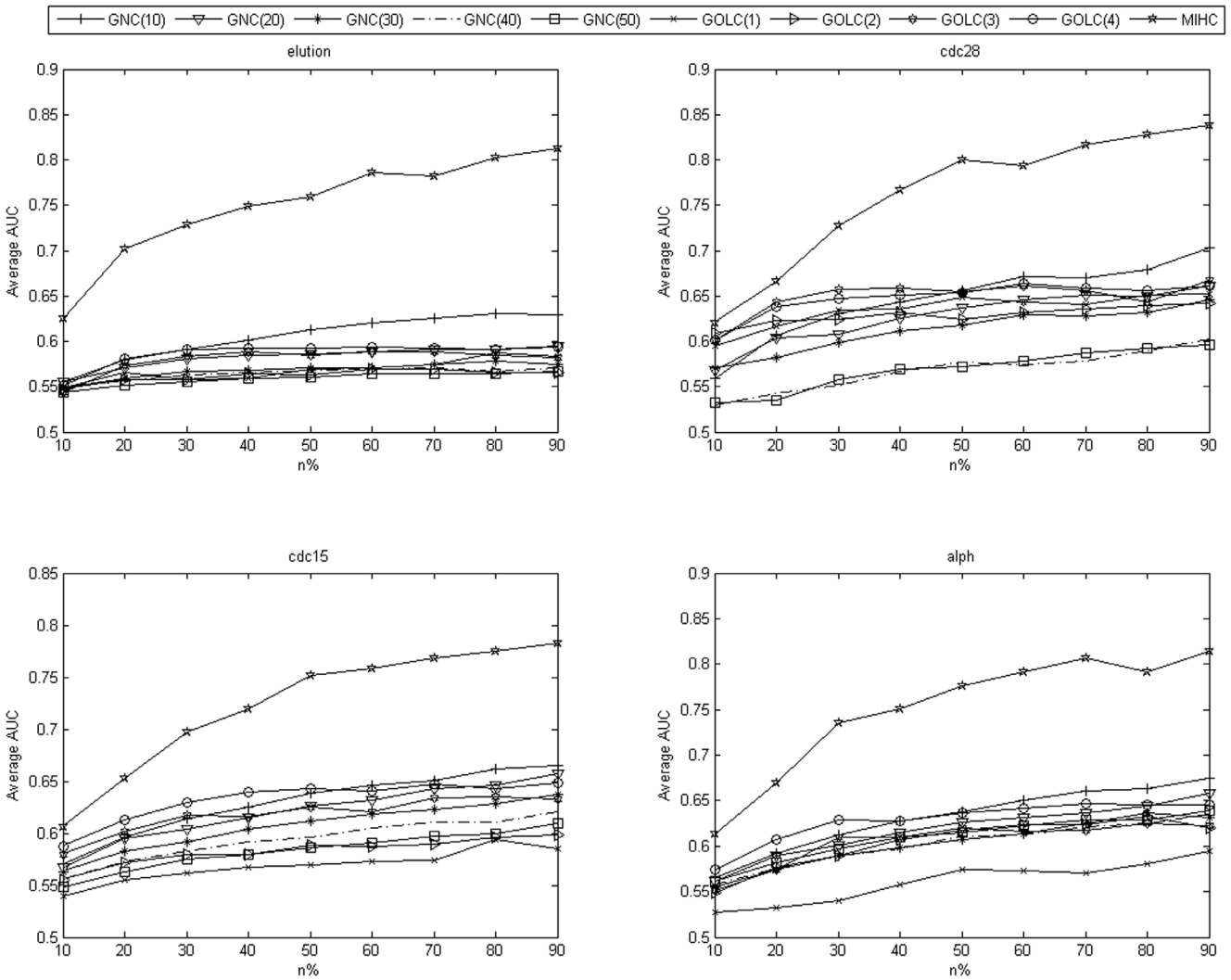

Figure 4. Average AUC obtained from each learning system by MLSVM in all datasets.
doi:10.1371/journal.pone.0090962.g004

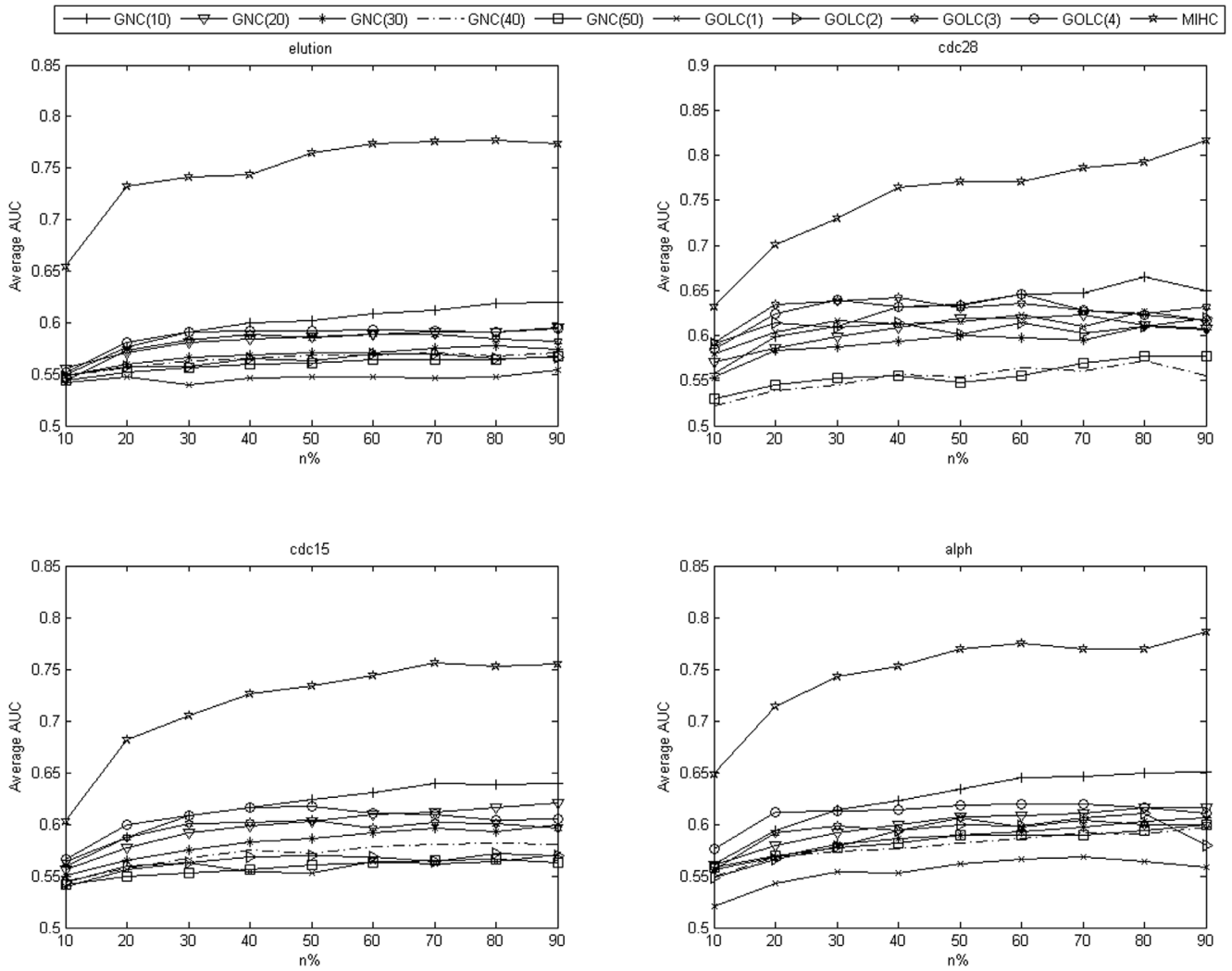

Figure 5. Average AUC obtained from each learning system by MLKNN for each expression dataset.
doi:10.1371/journal.pone.0090962.g005

Compared with the model proposed by Vapnik [22], the aforementioned model performs better in multi-label classification. Generally, multi-label classification is transformed to multiple binary classifications. Class imbalance problem is a considerable barrier for each binary classification. The parameter ξ_{iy} in Eq. (1) addresses this problem with a good performance.

MLKNN

The K-nearest neighbor is another stable popular machine learning method. This method performs more rapid classification than the SVM. Zhang et al. [23] improve KNN method for multi-label classification, which served as our motivation in our proposed model. In the MLKNN model, the candidate classes of a given test sample t_i are obtained by

$$Y_{tc}=\{y|\psi(x_i,y)=1,x_i\in KNN(t_i),y\in Y\} \quad (2)$$

where $KNN(t_i)$ is the k-nearest neighbor of t_i among the training set S. For each candidate class $y\in Y_{tc}$, the following likelihood score is calculated

$$Score(t_i.y)=\sum_{s_i\in KNN(t_i)} simScore(t_i,s_i)\psi(s_i,y) \quad (3)$$

where $simScore(t_i,s_i)$ is the similarity score of s_i to t_i. The labels of t_i are calculated by

$$Y_{ti}=\{y|Score(t_i,y)\geq 0\} \quad (4)$$

Gene selection

We are not interested in all genes in the gene expression profiles. In gene function prediction, we assume that genes participating in the same biological processes have similar expression profiles [2,24]. For this proposal, we select genes that are significantly correlated with each other in the same function. Let $G=\{gene_i|i=1,...,N\}$, $L=\{GO_i|i=1,...,M\}$, and $G(GO_i)=\{gene_j|gene_j\in G \text{ and annotated by } GO_i\}$, where N is the number of genes, and M is the number of GO terms. For each $GO_i\in L$, we draw a graph $graph_i=(v_i,e_i)$ for genes that significantly correlate with each other. $gene_i\in G(GO_i)$ represents the v_i of $graph_i$. An

Table 2. Average AUC obtained from cdc28 dataset by MLSVM.

		Parameter setting for n%								
Method	**Parameters**	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
GNC	$\lambda = 10$	0.559	0.606	0.631	0.644	0.656	0.671	0.670	0.679	0.703
	$\lambda = 20$	0.569	0.603	0.607	0.625	0.638	0.646	0.650	0.650	0.661
	$\lambda = 30$	0.571	0.583	0.599	0.612	0.618	0.629	0.628	0.632	0.646
	$\lambda = 40$	0.529	0.543	0.552	0.567	0.577	0.574	0.578	0.589	0.602
	$\lambda = 50$	0.532	0.535	0.558	0.569	0.572	0.579	0.588	0.592	0.596
GOLC	$\iota = 1$	0.594	0.617	0.634	0.635	0.648	0.643	0.641	0.651	0.653
	$\iota = 2$	0.609	0.623	0.624	0.631	0.624	0.632	0.635	0.640	0.643
	$\iota = 3$	0.601	0.644	0.657	0.658	0.654	0.661	0.656	0.643	0.668
	$\iota = 4$	0.601	0.638	0.647	0.651	0.654	0.663	0.658	0.656	0.662
MIHC	null	**0.621**	**0.666**	**0.727**	**0.767**	**0.800**	**0.794**	**0.817**	**0.828**	**0.838**

doi:10.1371/journal.pone.0090962.t002

edge exists between $gene_i$ and $gene_j$ if $gene_i$ and $gene_j$ are significantly correlated. We define $graph_i^{\max} = (v_i^{\max}, e_i^{\max})$, which is the maximum clique of $graph_i = (v_i, e_i)$ and $G_i = (v'_i | v'_i \in v_i^{\max})$. However, the maximum clique problem is complete NP-hard [25–27]. In this paper, a greedy algorithm is used to deal with this problem, and the non-noise system of expression data and annotation are represented as $S = \{(G_i, GO_i) | i = 1,...,M\}$.

Learning system establishment method

Prior to the prediction of gene function, we establish a learning system for classification. Learning system establishment is the reconstitution of gene labels. GO DAG and MIPS are usually used to aid the establishment of learning systems. Clare et al. [8] and Schietgat et al. [9] established an MIPS-based learning system. Based on GO DAG, we use the same approach as those in [8] and [9]. We called this method GO level clustering (GOLC), which up-propagates the gene annotations to a preset GO level ℓ, such as the first level (i.e., $\ell = 1$) of the GO DAG, and cluster genes. In another approach, Hvidsten et al. [18] used the method called gene number clustering (GNC) to establish the learning system. The GNC method let the annotations up-propagate along the GO DAG until each annotation has at least λ genes ($\lambda = 10,20$ in [18]). Figure 2 shows the two aforementioned methods.

MIHC method

The HC method is a widely used machine learning technology in the clustering algorithm. Johnson [28] proposed the extensively studied hierarchical clustering scheme (HCS). The HCS initializes all sample dissimilarities and then forms a cluster from the two closest samples or clusters. These steps are repeated until all samples are clustered to one group. Therefore, we can set a terminal factor to stop the cluster rather than preset the number of groups. Thus, HCS is suitable for all kinds of datasets.

To establish a more effective and efficient learning system, we import HCS and propose the novel MIHC method to establish a new learning system with the inherent characteristics of non-noise system S by cluster GO terms. In this method, we treat the relationship GO_i between $gene_i \in G_i$ as multi-instance. Our samples (i.e., GO) are different from the traditional HC [28–30] because they are multi-instances not instances. Therefore, the distance of each sample is redesigned. According to [31], we define the distance as follows:

Table 3. Average AUC obtained from cdc28 dataset by MLKNN.

		Parameter setting for n%								
Method	**Parameters**	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
GNC	$\lambda = 10$	0.550	0.576	0.591	0.599	0.603	0.609	0.619	0.619	0.620
	$\lambda = 20$	0.556	0.571	0.581	0.585	0.586	0.589	0.591	0.591	0.595
	$\lambda = 30$	0.548	0.559	0.567	0.568	0.571	0.571	0.575	0.578	0.575
	$\lambda = 40$	0.548	0.557	0.563	0.565	0.569	0.569	0.571	0.567	0.571
	$\lambda = 50$	0.544	0.552	0.556	0.559	0.560	0.564	0.564	0.564	0.567
GOLC	$\iota = 1$	0.541	0.547	0.540	0.546	0.547	0.547	0.547	0.547	0.554
	$\iota = 2$	0.548	0.557	0.558	0.565	0.563	0.569	0.569	0.566	0.566
	$\iota = 3$	0.544	0.573	0.584	0.588	0.585	0.589	0.589	0.585	0.582
	$\iota = 4$	0.552	0.580	0.591	0.592	0.592	0.594	0.592	0.591	0.594
MIHC	null	**0.655**	**0.733**	**0.741**	**0.743**	**0.765**	**0.774**	**0.776**	**0.777**	**0.774**

doi:10.1371/journal.pone.0090962.t003

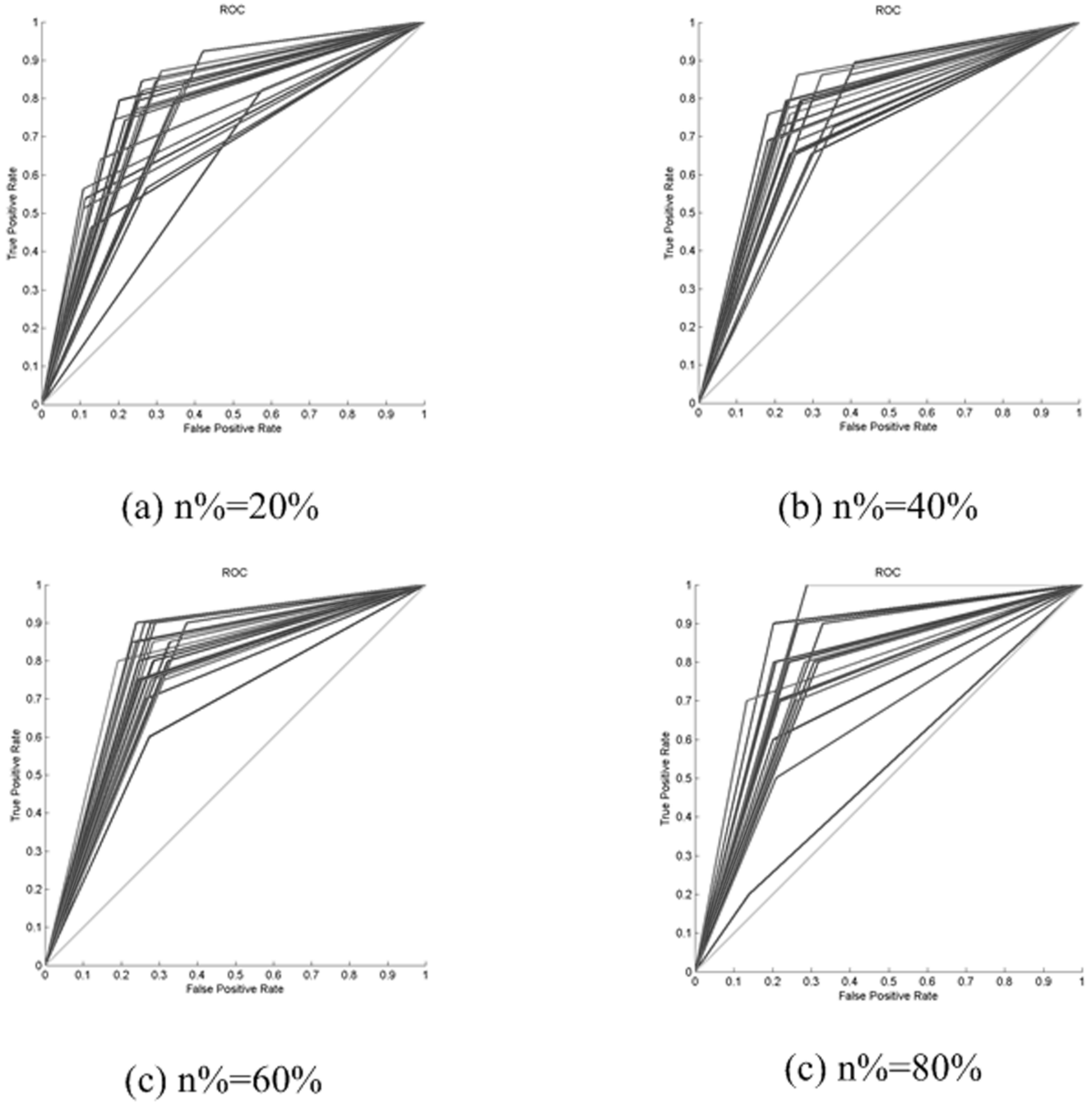

Figure 6. The ROC of a class obtained from the MIHC learning system by MLSVM for the cdc28 dataset. The ROC curves of the 20 repetitions of the experiment as well as the four subplots (a), (b), (c), and (d) with parameter n% = 20%, 40%, 60%, and 80%, respectively, are shown.
doi:10.1371/journal.pone.0090962.g006

$$Corr(G_i) = \sum \left| corr(gene_i, gene_j) \right|, where\ gene_i, gene_j \in G_i \quad (5)$$

$$D(GO_i, GO_j) = \overline{Corr(G_i \cup G_j)} \quad (6)$$

Where $corr(gene_i, gene_j)$ is the Pearson correlation of $gene_i$ and $gene_j$. Figure 3 shows the MIHC algorithm and flow chart of function prediction.

Results and Discussion

Data

The yeast time-course expression datasets used in this study are obtained from [32] (downloaded from http://genome-www.stanford.edu/cellcycle/data/rawdata/). The four datasets are yeast cell cycle expression data with different time points and circumstances. We use the method in [3], preprocess the raw data, and make the first value always equal to zero. Then, the average transformation $t_i = (t_i + t_{i-1})/2$ is used to smooth out the spikes. Gene annotation data can be obtained from GO [33] (download-

Table 4. Results of the TP, FP, TN, and FN in MIHC by MLSVM.

				First experiment results				Second experiment results			
n%	Train/Test	+	−	TP	FP	TN	FN	TP	FP	TN	FN
10%	Train	5	5								
	Test	44	207	19	45	162	25	22	30	177	22
20%	Train	10	10								
	Test	39	202	25	31	171	14	29	38	164	10
30%	Train	15	15								
	Test	34	197	24	40	157	10	26	44	153	8
40%	Train	20	20								
	Test	29	192	23	51	141	6	21	68	124	8
50%	Train	25	25								
	Test	24	187	17	54	133	7	20	55	132	4
60%	Train	29	29								
	Test	20	183	17	44	139	3	15	44	139	5
70%	Train	34	34								
	Test	15	178	11	41	137	4	12	50	128	3
80%	Train	39	39								
	Test	10	173	7	37	136	3	8	36	137	2
90%	Train	44	44								
	Test	5	168	4	40	128	1	5	36	132	0

1. '+' means positive sample number; '−' means negative sample number.

doi:10.1371/journal.pone.0090962.t004

ed from http://www.geneontology.org/GO.downloads.annotations.shtml). GO terms are composed of three disjointed DAGs, namely, biological process (BP), molecular function, and cellular component. We only use BP for this study because it is more complete than the two other disjointed DAGS.

Performance evaluation

Leave-one-out and leave-a-percent-out cross validation [14] approaches are two of the most extensively used methods for evaluating the performance of a function prediction algorithm. The former is usually used in a small dataset, whereas the latter is more suitable to a large dataset. The former method randomly leaves one sample of the experiment dataset for testing and assigns all of the other samples for training. This process is repeated many times. Meanwhile, the latter method splits the experiment dataset into two sets, namely, the training and testing sets. The training set is composed of a specified proportion of positive and negative samples, whose labels are known. Conversely, the labels of the testing set are concealed from the classifiers. The proportion of the training dataset is gradually increased to test the performance of the learning system. The true labels of the testing set are compared with the prediction labels to evaluate the performance of the system. We select the latter method to evaluate the MIHC method. To accurately measure the performance, the receiver operating characteristic (ROC) curve and area under the ROC curve (AUC) are introduced to quantify the results. The classifications are often based on continuous random variables. The probability of belonging in a class varies with different threshold parameters. That is, the values of true and false positive rates (TPR and FPR, respectively) vary with different threshold parameters. The ROC curve parametrically plots the TPR versus the FPR with varying parameters. The TPR and FPR are calculated by Equations (7) and (8).

Table 5. Average TPR and FPR in MIHC by MLSVM.

n%	TPR	FPR
10%	0.713	0.286
20%	0.733	0.255
30%	0.760	0.250
40%	0.757	0.264
50%	0.783	0.257
60%	0.808	0.272
70%	0.783	0.241
80%	0.745	0.234
90%	0.810	0.243

doi:10.1371/journal.pone.0090962.t005

$$TPR = TP/(TP + FN), \tag{7}$$

$$FPR = FP/(FP + TN), \tag{8}$$

Where TP, FP, TN, FN represent the number of true positive (TP), false positive (FP), true negative (TN), and false negative (FN) predictions, respectively. Therefore, the TPR and FPR can reflect the sensitivity and specificity of prediction. AUC is calculated to quantify the content of the ROC curves. A reliable and valid AUC estimate can be interpreted as the probability that the classifier will assign a higher score to a randomly chosen positive sample rather than to a randomly chosen negative sample.

Experiment analysis

The four yeast time-course expression datasets are as follows: alpha, cdc15, cdc28, and elution, which record the mRNA levels of 18, 24, 17, and 14 time points in the whole cell cycle under different circumstances, respectively. For each expression dataset, GNC ($\lambda = 10,20,30,40,50$), GOLC ($\ell = 1,2,3,4$), and MIHC methods are used to establish the learning system and compare their performances. The rationale for setting the value of the previously mentioned parameter is as follows. First, we want to determine whether different numbers and different levels of gene group remarkably change function prediction. Second, for the GOLC method, the error rate of a given level is accumulated if a deeper level gene function is required.

The number of genes in the MIHC learning system is consistent with the non-noise system, but other learning systems cannot maintain this feature. Table 1 shows the number of genes and classes for each learning system. The MIHC learning system also has better class features than other learning systems.

The MIHC learning system is tested on MLSVM and MLKNN classifiers. In the classification task, the MLL task is decomposed into a series of binary classification tasks. However, the negative samples are far more than the positive samples for each class. Therefore, class imbalance problem should be considered. Further information about the number of positive and negative samples in the cdc28 and elution experiment datasets are shown in Table S1 and Table S2 in the Supporting Information section. The training samples have to be balanced, that is, the same numbers of negative and positive samples are used for the training and testing sets. For each class, we randomly select n% of positive samples and the same number of negative samples as the training set, and the rest are for the testing set. The value of n% increased from 10% to 90%. If the number of positive samples in one class is very low (less than 10), the number of positive samples in the training set is increased gradually. The experiment is repeated 20 times (or more, and the mean value shows minimal changes) for each n, and the mean value is calculated. Given the class imbalance, a high accuracy can still be obtained when the classifier divides all the samples into negative. In this study, AUC is used to evaluate the performance of MIHC. We compare MIHC with GOLC and GNC. For each expression dataset, the average results obtained from each learning system by SVM and KNN classifiers are shown in Figures 4 and 5. Tables 2 and 3 show the results from cdc28 dataset. As the n% increased, the AUC value of MIHC increased drastically whereas those of GOLC and GNC increased slowly. These results prove that generally, the classes in the MIHC learning system are more interesting and the genes therein have more correlation power compared with those in the classes in the two other learning systems. This result can be explained as follows. Genes are transcribed into mRNA and then into proteins. To a certain extent, the level of mRNA can reflect the amount of protein being generated. However, this amount may be influenced by several factors, such as the decomposition of the speed of mRNA and the switching off of proteins. Cells are so efficient that only the necessary proteins are composed. Therefore, variances in gene expression match the active level of biological process. GNC and GOLC cluster GO by up-propagating it along with the GO DAG. Meanwhile, the MIHC method treats the gene expression profile as the feature of GO and clusters GO to ensure superior performance. Moreover, when GO is further up-propagated, the information that reflects the correlation between genes may be lost. Only the GO dataset determines which genes own which GO and whether or not the gene exerts a certain function of the GO in the experiment dataset. However, we assume that genes exert all their GO because the datasets in our study consist of cell cycle expression data. Compared to GNC and GOLC, MIHC relies on statistical correlation. Consequently, MIHC is less concerned about whether or not the gene exerts the function. This problem will be certainly considered in the future study.

Lastly, to obtain a satisfactory explanation in a real-world problem, the ROC curve of a class obtained from the MIHC learning system for the cdc28 dataset is shown in Figure 6. The results for the TP, FP, TN, and FN are shown in Table 4 (given that the experiment is repeated 20 times, only the middle-level results for the 2 repetitions are shown in Table 4; the average TPR and FPR in the 20 repetitions of the experiment are shown in Table 5). In Figure 6, the ROC curves of the 20 repetitions of the experiment as well as the four subplots (a), (b), (c), and (d) with parameter n% = 20%, 40%, 60%, and 80%, respectively, are displayed. As n increases, the number of positive samples in the testing set decreases. The classifier sometimes pays a greater price to identify as many positive samples in the testing set as possible. The sample distribution in training set may also influence the prediction result. The ROC curve in subplot (d) occasionally exhibits unsatisfactory performance. The ROC curves of all the datasets are presented in Figures S1 to S8, and the average TPR and FPR of all the datasets are shown in Tables S3 to S10 in the Supporting Information section.

Conclusion

In this paper, we propose the MIHC method to establish a learning system, which is verified by SVM and KNN using four yeast gene expression datasets. In the MIHC method, Pearson correlation is the distance between multi-instance samples, and HC is used to cluster the samples. Compared with other learning system establishment methods, the MIHC learning system exhibits better performance because the samples are more easily recognized. This method also maintains data integrity with non-noise system. To our knowledge, this study is the first to use HC algorithm to cluster multi-instance samples.

Supporting Information

Figure S1 ROC curves are obtained from cdc28 dataset by MLSVM. The ROC curves of each learning system, generated by average TPR and FPR, as well as the four subplots (a), (b), (c), and (d) with parameter n% = 20%, 40%, 60%, and 80%, respectively, are shown.
(TIF)

Figure S2 ROC curves are obtained from cdc28 dataset by MLKNN. The ROC curves of each learning system, generated by average TPR and FPR, as well as the four subplots (a), (b), (c), and (d) with parameter n% = 20%, 40%, 60%, and 80%, respectively, are presented.
(TIF)

Figure S3 ROC curves are obtained from cdc15 dataset by MLSVM. The ROC curves of each learning system, generated by average TPR and FPR, as well as the four subplots (a), (b), (c), and (d) with parameter n% = 20%, 40%, 60%, and 80%, respectively, are shown.
(TIF)

Figure S4 ROC curves are obtained from cdc15 dataset by MLKNN. The ROC curves of each learning system, generated by average TPR and FPR, as well as the four subplots (a), (b), (c), and (d) with parameter n% = 20%, 40%, 60%, and 80%, respectively, are displayed.
(TIF)

Figure S5 ROC curves are obtained from alpha dataset by MLSVM. The ROC curves of each learning system, generated by average TPR and FPR, as well as the four subplots (a), (b), (c), and (d) with parameter n% = 20%, 40%, 60%, and 80%, respectively, are displayed.
(TIF)

Figure S6 ROC curves are obtained from alpha dataset by MLKNN. The ROC curves of each learning system, generated by average TPR and FPR, as well as the four subplots (a), (b), (c), and (d) with parameter n% = 20%, 40%, 60%, and 80%, respectively, are presented.
(TIF)

Figure S7 ROC curves are obtained from elution dataset by MLSVM. The ROC curves of each learning system, generated by average TPR and FPR, as well as the four subplots (a), (b), (c), and (d) with parameter n% = 20%, 40%, 60%, and 80%, respectively, are shown.
(TIF)

Figure S8 ROC curves are obtained from elution dataset by MLKNN. The ROC curves of each learning system, generated by average TPR and FPR, as well as the four subplots (a), (b), (c), and (d) with parameter n% = 20%, 40%, 60%, and 80%, respectively, are displayed.
(TIF)

Table S1 Number of positive and negative samples in MIHC from the cdc28 dataset.
(XLS)

Table S2 Number of positive and negative samples in MIHC from the elution dataset.
(XLS)

Table S3 Average TPR and FPR obtained from the cdc28 dataset by MLKNN.
(XLS)

Table S4 Average TPR and FPR obtained from the cdc28 dataset by MLSVM.
(XLS)

Table S5 Average TPR and FPR obtained from the cdc15 dataset by MLKNN.
(XLS)

Table S6 Average TPR and FPR obtained from the cdc15 dataset by MLSVM.
(XLS)

Table S7 Average TPR and FPR obtained from the alpha dataset by MLKNN.
(XLS)

Table S8 Average TPR and FPR obtained from the alpha dataset by MLSVM.
(XLS)

Table S9 Average TPR and FPR obtained from the elution dataset by MLKNN.
(XLS)

Table S10 Average TPR and FPR obtained from the elution dataset by MLSVM.
(XLS)

Author Contributions

Conceived and designed the experiments: BL YL. Performed the experiments: BL YL. Analyzed the data: YJ LC. Contributed reagents/materials/analysis tools: YJ LC. Wrote the paper: BL YL.

References

1. Lægreid A, Hvidsten TR, Midelfart H, Komorowski J, Sandvik AK (2003) Predicting gene ontology biological process from temporal gene expression patterns. Genome research 13: 965–979.
2. Eisen MB, Spellman PT, Brown PO, Botstein D (1998) Cluster analysis and display of genome-wide expression patterns. Proceedings of the National Academy of Sciences 95: 14863–14868.
3. Ernst J, Nau GJ, Bar-Joseph Z (2005) Clustering short time series gene expression data. Bioinformatics 21: i159–i168.
4. Ernst J, Bar-Joseph Z (2006) STEM: a tool for the analysis of short time series gene expression data. BMC bioinformatics 7: 191.
5. Ma P, Castillo-Davis CI, Zhong W, Liu JS (2006) A data-driven clustering method for time course gene expression data. Nucleic Acids Research 34: 1261–1269.
6. Tibshirani R, Hastie T, Eisen M, Ross D, Botstein D, et al. (1999) Clustering methods for the analysis of DNA microarray data. Dept Statist, Stanford Univ, Stanford, CA, Tech Rep.
7. Chen Y, Xu D (2004) Global protein function annotation through mining genome-scale data in yeast Saccharomyces cerevisiae. Nucleic Acids Research 32: 6414–6424.
8. Clare A, King RD (2003) Predicting gene function in Saccharomyces cerevisiae. Bioinformatics 19: ii42–ii49.
9. Schietgat L, Vens C, Struyf J, Blockeel H, Kocev D, et al. (2010) Predicting gene function using hierarchical multi-label decision tree ensembles. BMC bioinformatics 11: 2.
10. Kim WK, Krumpelman C, Marcotte EM (2008) Inferring mouse gene functions from genomic-scale data using a combined functional network/classification strategy. Genome Biol 9: S5.
11. Vazquez A, Flammini A, Maritan A, Vespignani A (2003) Global protein function prediction from protein-protein interaction networks. Nature biotechnology 21: 697–700.
12. Deng M, Zhang K, Mehta S, Chen T, Sun F (2003) Prediction of protein function using protein-protein interaction data. Journal of Computational Biology 10: 947–960.
13. Nabieva E, Jim K, Agarwal A, Chazelle B, Singh M (2005) Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. Bioinformatics 21: i302–i310.
14. Magi A, Tattini L, Benelli M, Giusti B, Abbate R, et al. (2012) WNP: a novel algorithm for gene products annotation from weighted functional networks. PloS one 7: e38767.
15. Liang S, Zheng D, Standley DM, Guo H, Zhang C (2013) A novel function prediction approach using protein overlap networks. BMC systems biology 7: 61.
16. Mitsakakis N, Razak Z, Escobar MD, Westwood JT (2013) Prediction of Drosophila melanogaster gene function using Support Vector Machines. BioData mining 6: 8.
17. Khatri P, Drăghici S (2005) Ontological analysis of gene expression data: current tools, limitations, and open problems. Bioinformatics 21: 3587–3595.
18. Hvidsten TR, Komorowski HJ, Sandvik AK, Lægreid A (2001) Predicting gene function from gene expressions and ontologies; pp.299–310.
19. Zhou Z-H, Zhang M-L (2006) Multi-instance multi-label learning with application to scene classification; pp.1609–1616.
20. Li Y-X, Ji S, Kumar S, Ye J, Zhou Z-H (2012) Drosophila gene expression pattern annotation through multi-instance multi-label learning. Computational Biology and Bioinformatics, IEEE/ACM Transactions on 9: 98–112.
21. Cortes C, Vapnik V (1995) Support-vector networks. Machine learning 20: 273–297.
22. Vapnik V (2006) Estimation of dependences based on empirical data: Springer.
23. Zhang M-L, Zhou Z-H (2007) ML-KNN: A lazy learning approach to multi-label learning. Pattern Recognition 40: 2038–2048.
24. Iyer VR, Eisen MB, Ross DT, Schuler G, Moore T, et al. (1999) The transcriptional program in the response of human fibroblasts to serum. Science 283: 83–87.
25. Östergård PR (2002) A fast algorithm for the maximum clique problem. Discrete Applied Mathematics 120: 197–207.
26. Eblen JD, Phillips CA, Rogers GL, Langston MA (2012) The maximum clique enumeration problem: algorithms, applications, and implementations. BMC bioinformatics 13: S5.
27. Punnen AP, Zhang R (2012) Analysis of an approximate greedy algorithm for the maximum edge clique partitioning problem. Discrete Optimization 9: 205–208.
28. Johnson SC (1967) Hierarchical clustering schemes. Psychometrika 32: 241–254.
29. Murtagh F (1983) A survey of recent advances in hierarchical clustering algorithms. The Computer Journal 26: 354–359.
30. Langfelder P, Horvath S (2012) Fast R functions for robust correlations and hierarchical clustering. Journal of statistical software 46.

31. Zhou Z-H (2004) Multi-instance learning: A survey. AI Lab, Department of Computer Science and Technology, Nanjing University, Tech Rep.
32. Spellman PT, Sherlock G, Zhang MQ, Iyer VR, Anders K, et al. (1998) Comprehensive identification of cell cycle–regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. Molecular biology of the cell 9: 3273–3297.
33. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, et al. (2000) Gene Ontology: tool for the unification of biology. Nature genetics 25: 25–29.