

SCIENTIFIC REPORTS



OPEN

An Integrated Local Classification Model of Predicting Drug-Drug Interactions via Dempster-Shafer Theory of Evidence

Jian-Yu Shi¹ , Xue-Qun Shang², Ke Gao², Shao-Wu Zhang³ & Siu-Ming Yiu⁴

Drug-drug interactions (DDIs) may trigger adverse drug reactions, which endanger the patients. DDI identification before making clinical medications is critical but bears a high cost in clinics. Computational approaches, including global model-based and local model based, are able to screen DDI candidates among a large number of drug pairs by utilizing preliminary characteristics of drugs (e.g. drug chemical structure). However, global model-based approaches are usually slow and don't consider the topological structure of DDI network, while local model-based approaches have the degree-induced bias that a new drug tends to link to the drug having many DDI. All of them lack an effective ensemble method to combine results from multiple predictors. To address the first two issues, we propose a local classification-based model (LCM), which considers the topology of DDI network and has the relaxation of the degree-induced bias. Furthermore, we design a novel supervised fusion rule based on the Dempster-Shafer theory of evidence (LCM-DS), which aggregates the results from multiple LCMs. To make the final prediction, LCM-DS integrates three aspects from multiple classifiers, including the posterior probabilities output by individual classifiers, the proximity between their instance decision profiles and their reference profiles, as well as the quality of their reference profiles. Last, the substantial comparison with three state-of-the-art approaches demonstrates the effectiveness of our LCM, and the comparison with both individual LCM implementations and classical fusion algorithms exhibits the superiority of our LCM-DS.

Drug-drug interactions (DDIs) may occur unexpectedly when drugs are co-prescribed. The identification of DDIs is difficult in the process of drug development¹ because of both a small number of participants and a large number of drug pairs in clinical screening. Thus, a minority of DDIs could be usually identified in the clinical trial phase while a majority of them are only reported after the co-prescription of multi-drugs are made. On the one hand, DDIs would trigger adverse drug reactions, such as efficacy reduction and toxicity increment, such that the patients, who are treated with multiple drugs, are led into unsafe and even incorrect medications²⁻⁵. The broad-casting of adverse effects caused by DDIs cannot be neglected, because the number of potential DDIs is rising with the power of two along with the increasing number of approved drugs. On the other hand, the identification of DDIs is one of the crucial steps towards finding synergistic drug combinations⁶, which further cover the issues about drug targets, drug resistance and drug sensitivity⁷.

Consequently, it is imperative to identify DDIs before multi-drug co-prescriptions are made. However, experimental approaches (e.g. testing cytochrome P450⁸ or transporter-associated interactions⁹) are performed under the consideration of animal welfare with high costs in both money and time. In contrast, computational approaches can help screen DDI candidates among a large number of drug pairs with much lower costs. They have winning interests from both academy and industry recently¹⁰⁻¹².

Text-mining based computational approaches apply the techniques of text-mining to detect DDIs recorded in diverse text sources, such as scientific literatures¹³⁻¹⁵, electronic medical records¹⁶, and the Adverse Event

¹School of Life Sciences, Northwestern Polytechnical University, Xi'an, 710072, China. ²School of Computer Science, Northwestern Polytechnical University, Xi'an, 710072, China. ³School of Automation, Northwestern Polytechnical University, Xi'an, 710072, China. ⁴Department of Computer Science, The University of Hong Kong, Hong Kong, 999077, China. Correspondence and requests for materials should be addressed to J.-Y.S. (email: jianyushi@nwpu.edu.cn)

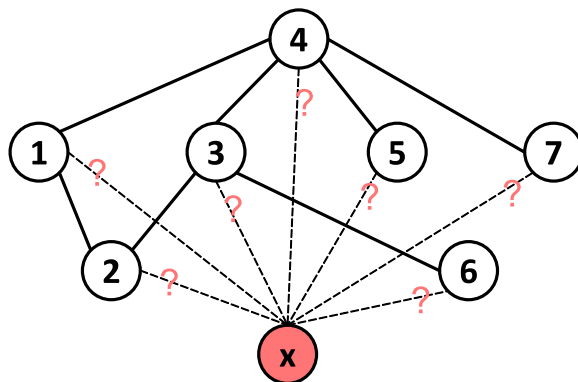


Figure 1. Illustration of LCM predicting DDI for a newly given drug. Nodes are drugs. The hollow nodes are known drugs and the solid lines between them denote their interactions. The node filled with red is the newly given drug. Our problem is to determine which drugs it is likely to interact with.

Reporting System of FDA. These approaches are particularly helpful when building a DDI database. However, they cannot predict or alert newly potential DDIs for multi-drug treatment, because they depend on the evidence, which reports the DDIs found in clinical treatments.

Many works in other areas have demonstrated that the utilization of pre-existing knowledge is promising to build a predictive model with the advantages of both low cost and good effectiveness. For example, the genes identified *in silico* as cancer prognostic biomarkers¹⁷ show their strong correlations with cancers, such that they have been widely used to build predictive models for diverse cancer-related alerts (e.g. survival of patients under PIK3CA-mutated Breast cancer¹⁸, tumor clinical phenotypes¹⁹, and recurrence of colorectal cancer²⁰). For another example similar to DDI prediction, the observation that similar drugs tend to interact with similar targets inspires the development of diverse models for predicting drug-target interaction^{21–24}. Analogously, by leveraging pre-existing drug properties (e.g. chemical structures²⁵, targets²⁶, drug classification codes²⁷ and side effects²⁸) which can be acquired before multiple drugs are co-prescribed, similarity-based computational approaches are able to provide predictive models for deducing potential DDIs. These approaches can be grouped as global model-based (e.g. global classification-based²⁷) and local model-based (e.g. naïve similarity-based approach²⁵ and network recommendation-based²⁸). They enable drug safety professionals to screen potential DDIs fast so as to help make appropriate clinical multi-drugs treatments.

Similarity-based approaches usually hold diverse assumptions. Global classification model-based approaches (GCM) accept that DDIs are globally distinct to non-DDIs²⁷. After treating DDIs and non-DDIs as positive and negative instances respectively, it trains a classifier to perform DDI prediction for new drugs. However, GCM neglects the topological relationship between DDIs and non-DDIs. Local model-based approaches (e.g. Naïve similarity-based approach²⁵ and label propagation²⁸) consider such a relationship to some extent and utilize the local topology of a DDI network. They usually run faster than GCM. Nevertheless, both of them have the intrinsic degree-induced bias that greatly tends to generate a high confidence of being a DDI to the pair of a newly-given drug having no known DDI and a known drug having many DDIs. Even though such a bias is probably useful to DDI prediction, it deserves a better utilization or relaxation.

Furthermore, though current approaches always utilize a single model to perform DDI prediction, they lack the effective combination of multiple predictions generated by different predictors/models (e.g. classifiers). The fact that different models may fit different data characteristics of pre-existing drug properties/knowledge should be considered.

To address the abovementioned issues, we first propose a novel local classification-based model (LCM) under the assumption that similar drugs tend to interact with the same drugs. Then, we design a novel supervised fusion algorithm based on the Dempster-Shafer theory of evidence (LCM-DS), which aggregates the results from different LCMs. Last, in the scenario of predicting potential DDIs for new drugs, the results demonstrate that the proposed LCM is superior to three state-of-the-art approaches greatly and its ensemble version LCM-DS outperforms both three individual LCM implementations and five classical fusion algorithms.

Methods

Local Classification Model. Given m drugs, $D = \{d_i\}$, $i = 1, 2, \dots, m$, of which each has at least one DDI with others. Their pairwise interactions are accordingly arranged into an $m \times m$ binary symmetric matrix $\mathbf{A}_{m \times m} = \{a_{ij}\}$, in which $a_{ij} = a_{ji} \in \{0, 1\}$, $a_{ij} = 1$ if the interaction between d_i and d_j occurs, $a_{ij} = 0$ otherwise. Moreover, their pairwise similarities are organized into another $m \times m$ positive symmetric matrix $\mathbf{S}_{m \times m} = \{s_{ij}\}$, where $s_{ij} \in \mathbb{R}_+$ denotes the similarity between d_i and d_j . For a newly given drug d_x , which has no interaction with any drugs in D , its pairwise similarities to all d_i are also organized into a vector $\mathbf{S}_{1 \times m}^x$.

Our problem is to infer how likely the new drug d_x interacts with the drugs in D and it is represented as a set of local drug-specific classifications as follows.

In the local classification specific to drug d_i in D , we first label the drugs interacting with d_i as positive instances and other drugs in D as negative instances. For example in Fig. 1, when predicting how likely d_x interacts with d_4 , we assign d_1, d_3, d_5 and d_7 with positive labels, d_2 and d_6 with negative labels respectively. Then, we train a classifier

C specific to d_i by the labels \mathbf{L} of the drugs and their pairwise similarity matrix $\mathbf{S}_{m \times m}$. Finally, we apply the well-trained classifier C on the unlabeled instance d_x to obtain its label. Generally, the classifier simply outputs a single label to denote a positive or a negative instance. Because we need to know how likely d_x interact with a specific drug in D , the classifier is required to output a 2-dimensional decision profile vector $\mathbf{y}^x = C(x) = [p_+, p_-]$, where $p_+, p_- \in [0, 1]$ are the probabilities of d_x being a positive instance and a negative instance respectively, and they satisfy $p_+ + p_- = 1$.

The proposed LCM has a faster training and a requirement of less memory than GCM²⁷ because the number of instances handled by LCM is the number of drugs but not the number of drug pairs, which is usually huge, handled by GCM. Compared with NS²⁵ and LP²⁸, LCM is able to minimize their intrinsic degree-induce bias because the prediction for a new drug depends only on the distributions of positive instances and negative instances in feature or similarity space (see also Section 3.2).

Similarity Calculation. Drugs are popularly represented as binary profiles according to diverse drug properties, such as fingerprints of chemical structures and keyword occurrence lists of side effects. In the binary profile of a drug, each entry denotes the presence or absence of one of its concerned properties by 1 or 0 respectively. A classical similarity measure widely adopted by former works is Jaccard Index (also called as Tanimoto coefficient). Technically, the pairwise similarity between two drugs is defined as $S_{jaccard}(i, j) = |\mathbf{f}_i \cap \mathbf{f}_j| / |\mathbf{f}_i \cup \mathbf{f}_j|$, where the numerator is the number of common presence entries between \mathbf{f}_i and \mathbf{f}_j while the denominator is the number of presence entries in their binary union. Once a similarity matrix is given, it can be exploited to train a classifier and make the prediction.

Classifiers. Except for similarity, the classifier is another crucial factor in classification. When implementing LCM, we considered three classifiers, multi-label K nearest neighbors (MLKNN)²³, Regularized Least Squared classifier (RLS)^{24,29} and Support Vector Machines (SVM)³⁰, of which all can accept the form of similarity matrix as their input. Their brief introductions are shown in the following respectively. In addition, we refer to drugs as instances in the context of classification.

- MLKNN: Denote $N^j(x, K)$ as the set of K nearest neighbors of instance d_x , $n^j(x, K)$ as the number of neighbors interacting with d_j (having positive labels) among $N^j(x, K)$, and p_x^j as the probability that d_x interacts with d_j (a positive label). When d_x is a testing instance, $p_x^j \in [0, 1]$ defines its confidence score of being a positive instance as follows

$$p_x^j = \frac{\Pr[y^j = 1] \cdot \Pr[n^j(x, K) = k | y^j = 1]}{\sum_{t \in \{0, 1\}} \Pr[y^j = t] \cdot \Pr[n^j(x, K) = k | y^j = t]} \quad (1)$$

where $\Pr[y^j = t]$ is the prior probability of an instance having label t and $\Pr[n^j(x, K) = k | y^j = t]$ is the probability of d_x having k neighbors under the condition of a positive/negative label with respect to d_j . Two prior

probabilities can be directly estimated by $\Pr[y^j = 1] \approx (1 + c^j)/(m + 2)$ where c^j is the number of drugs

interacting with d_j and $\Pr[y^j = 0] = 1 - \Pr[y^j = 1]$. The conditional probability can be estimated by

$$\Pr[n^j(x, K) = k | y^j = t] = \frac{1 + \sum_i B[y^j(i) = t \& n^j(x, K) = k]}{(K + 1) + \sum_{k'=0}^K \sum_i B[y^j(i) = t \& n^j(x, K) = k']} \quad (2)$$

where $y^j(i) = t$ is the i -th drugs having label t , $B[S] = 1$ if statement S is correct and $B[S] = 0$ otherwise. Totally, we generate two probability tables, which account for positive instances and negative instances respectively. Each of them contains $K + 1$ probability entries, which correspond to the $K + 1$ possible values with respect to $n^j(x, K) = 0, 1, \dots, K$ respectively.

Note that, for a queried instance, the theoretical version of MLKNN uses the distances to other instances to find its top K neighbors³¹, while our input is a set of pairwise similarities between instances (organized into a similarity matrix). To bridge the gap, we need to turn similarities into distances by two points. First, the smaller the distance between two instances is, the greater their similarity is. In addition, the value of distance should be non-negative. Thus, the distance between two instances is finally defined as by 1- their similarity, such that the K nearest neighbors of an instance are just the top K most similar instances to it²³.

- RLS: Let \mathbf{D} be the set of the training instances (drugs), d_x be the testing instance, $\mathbf{Y}_j = \mathbf{A}(:, j)$ be the $m \times 1$ class label vector of training instances which are specific to drug d_j and correspond to the j -th column of the interaction matrix, and $\mathbf{K}(\mathbf{X}_1, \mathbf{X}_2)$ be the kernel matrix, which reflects the pairwise similarities between two groups of drugs. Specifically, $\mathbf{K}(\mathbf{D}, \mathbf{D}) = \mathbf{S}_{m \times m}$, which contains the pairwise similarities of \mathbf{D} , and $\mathbf{K}(d_x, \mathbf{D}) = \mathbf{S}_{1 \times m}^x$, which contains the pairwise similarities between d_x and m training drugs. RLS classifier is an elegant linear system, which has the order equal to the number of training instances²⁴. The trained RLS classifier outputs the confidence score $f_j(d_x)$ of how likely a given new drug d_x interacts with drug d_j as follows,

$$f_j(d_x) = \mathbf{K}(d_x, \mathbf{D})(\mathbf{K}(\mathbf{D}, \mathbf{D}) + \alpha \mathbf{I})^{-1} \mathbf{Y}_j \quad (3)$$

where \mathbf{I} is the $m \times m$ identity matrix and α is the regularization parameter (usually equal to 0.5) to prevent overfitting.

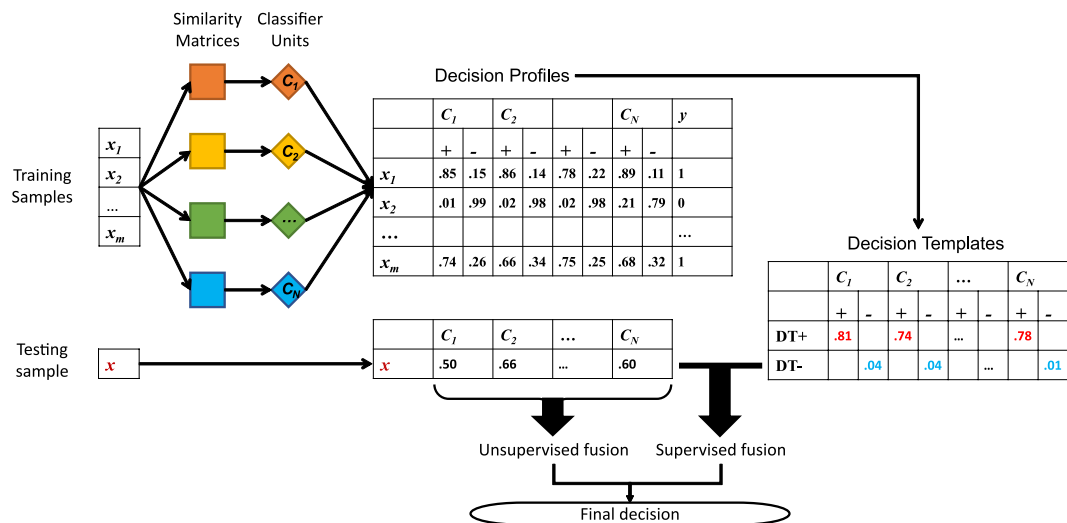


Figure 2. Architecture of classifier fusion.

- SVM: Similar to RLS, SVM is also a kernel-based classifier, which can perform the highly non-linear classification as a linear classification by kernel trick³⁰. Usually, the training of a binary d_j -specific SVM depends on the solution of the following optimization problem

$$\begin{aligned} \max_{\alpha} \sum_{k=1}^M \alpha_k^j - \frac{1}{2} \sum_{k=1}^M \sum_{i=1}^M \alpha_k^j \alpha_i^j y_k^j y_i^j \mathbf{K}(d_k, d_i), \\ \text{s.t. } 0 \leq \alpha_k^j \leq \gamma \text{ and } \sum_{k=1}^M \alpha_k^j y_k^j = 0 \end{aligned} \quad (4)$$

where $y_k^j \in \{0, 1\}$ is the d_j -specific label of the training drug d_k , M is the number of the training instances, \mathbf{K} is the kernel function, γ is the tunable parameter to reflect the trade-off between the training error and the margin of separation, and the variable α_k^j to be solved is the d_j -specific weight of d_k . Once the training of SVM is done, for the given testing instance d_x , it outputs the confidence score of how likely a given new drug d_x interacts with drug d_j by a linear operation as follows,

$$s_j(d_x) = \sum_{k=1}^M \alpha_k^j y_k^j \mathbf{K}(d_k, d_x) + b. \quad (5)$$

The abovementioned three classifiers shall be taken as the member classifiers when performing the integration of classifiers for DDI prediction in the next section.

Classifier fusion. In the context of classifier fusion, our problem is restated as the inference of how likely a given drug d_x interacts with a specific drug d_j by combining the evidences generated by a group of classifiers.

Formally, given m training instances $X = \{x_i\}$, $i = 1, 2, \dots, M$ with their labels $l_i \in \{1, 2, \dots, K\}$ where K is the number of classes (e.g. $K = 2$ in our problem). N classifiers $\{C_n\}$ $n = 1, \dots, N$ are trained by their similarity matrices and their labels. Classifier C_n generates a K -dimensional decision profile $\mathbf{y}_n^x \in \mathfrak{R}^K$ for an unlabeled instance x .

Denote $e_k(\mathbf{y}_n^x)$ as the evidence supporting the proposition “classifier C_n thinks that x is of class k ”. Classifier fusion combines the evidences $\{e_k(\mathbf{y}_n^x)\}$ to make a final decision of how likely x is of class k . The general architecture of classifier fusion is shown in Fig. 2.

The fusion rules of evidences can be categorized into unsupervised and supervised rules. Unsupervised rules combine those output evidences of x directly by arithmetical operations, such as Average, Product, Maximum and Minimum. For a given unlabeled instance x , the above four unsupervised rules are defined as follows

$$\begin{aligned} e_k^{ave}(\mathbf{y}^x) &= \sum_{n=1}^N e_k(\mathbf{y}_n^x) / N, \quad e_k^{prod}(\mathbf{y}^x) = \prod_{n=1}^N e_k(\mathbf{y}_n^x), \\ e_k^{max}(\mathbf{y}^x) &= \max_{n=1, \dots, N} \{e_k(\mathbf{y}_n^x)\}, \quad e_k^{min}(\mathbf{y}^x) = \min_{n=1, \dots, N} \{e_k(\mathbf{y}_n^x)\}. \end{aligned} \quad (6)$$

While supervised rules need to generate a training profile by the evidences of the training instances, and integrate the training profile with the evidences of x generated by classifiers to make the combined evidence of x . Decision Template is a popular supervised rule³² and has been applied in other related areas (e.g. drug-target interaction prediction³³). It combines the evidence of x from different classifiers by

$$e_k(\mathbf{y}^x) = 1 - \frac{1}{N} \sum_{n=1}^N (\mathbf{y}_n^x - \mathbf{DT}_k)^2 \tag{7}$$

where $\mathbf{DT}_k = \frac{1}{N_k} \sum_{i=1}^{N_k} \mathbf{y}_n^{x_i}$ is the decision template of class k , which is generated from the training instances. In details, x_i denotes the instances having class label $l_i = k$ in X , $\mathbf{y}_n^{x_i}$ is their discriminating profile, and N_k is the number of such instances and $\sum_{k=1}^K N_k = M$.

In the next section, we should introduce a novel supervised fusion based on the Dempster-Shafer Theory of Evidence.

Dempster-Shafer Theory of Evidence. When representing and combining measures from different sources (e.g. the decisions of multiple classifiers), the Dempster–Shafer (DS) theory of evidence provides a better frame of discernment than the Bayesian theory by generalizing Bayesian reasoning³⁴. This theory defines a set of mutually exhaustive and exclusive atomic hypotheses $\Theta = \{\theta_1, \dots, \theta_K\}$, and its power set 2^Θ that contains the empty set \emptyset , Θ itself and other subsets of Θ . For the K -dimensional decision profile generated by classifier C_n generates, each hypothesis θ_k represents that “ \mathbf{y}_n^x is of class k .” In the case of binary classification, $\Theta = \{+, -\}$ and its power set $2^\Theta = \{\emptyset, \{+\}, \{-\}, \Theta\}$.

The DS theory of evidence also assigns a belief mass function, called Basic Belief Assignment (BBA), to each element in 2^Θ . The BBA function $m: 2^\Theta \rightarrow [0, 1]$ is defined as

$$m(\emptyset) = 0 \text{ and } \sum_{A \in 2^\Theta} m(A) = 1 \tag{8}$$

where A is named as a composite hypothesis, which may contain an individual atomic hypothesis or multiple atomic hypotheses, and it satisfies $m(A) + m(\bar{A}) \leq 1$. In classification, A represents that “ \mathbf{y}_n^x is only of composite class A but none of its subsets” such that the conflict between evidences can be modeled. The BBA function $m(A)$ reflects how many relevant and available evidences support the composite hypothesis. The theory provides a combination rule $m = m_1 \oplus m_2$ for two BBAs m_1 and m_2 . It is defined as:

$$m(A) = Z^{-1} \sum_{B \cap C = A} m_1(B) \cdot m_2(C) \tag{9}$$

where $Z = \sum_{B \cap C = \emptyset} m_1(B) \cdot m_2(C)$.

Furthermore, this theory defines a belief function $\text{Bel}: 2^\Theta \rightarrow [0, 1]$, which is the sum of all the masses of subsets B of the set of interest A and satisfies $\text{Bel}(A) = \sum_{B \subseteq A} m(B)$. Suppose that a simple support function Bel satisfies $\text{Bel}(\Theta) = 1$ and its focus $F \subseteq \Theta$. We have $\text{Bel}(A) = 0$ if $F \not\subseteq A$ and $\text{Bel}(A) = b$ if $F \subseteq A$ & $A \neq \Theta$, where b is called Bel’s degree of support.

Therefore, a BBA can be considered as a generalization of a probability density function, while a Bel is a generalization of a probability function. Obviously, if A is an atomic hypothesis, $\text{Bel}(A) = m(A)$. Besides, in the case of $m(\theta_j) \neq 0$ for all the atomic hypotheses and $m(A) = 0$ for all the composite hypotheses, DS theory would simplify itself as the Bayesian probability theory.

DS-Based Fusion. Our problem is now to predict how likely a given drug d_x interacts with a specific drug d_j according to the evidences generated by N classifiers. Inspired by Rogova’s work³⁵, we consider the entry accounting for posterior probability of each class in the decision profile vector as a BBA (Equation 8) and design a novel DS-based fusion algorithm to address this problem in the following.

Define the reference profile \mathbf{R}_k^n w.r.t class k and classifier C_n as the mean vector of a set of decision profile vectors $\{C_n(x_k^{(m)})\}$, where $\{x_k^{(m)}\}$ are the training instances belonging to class k . Class-conditional probability distributions for all K classes can be estimated by both intra-class and inter-class distances between the decision profile vectors of instances and the class-specific reference profiles³⁶. Thus, the reference profiles can largely reflect the abilities of C_n in classification.

Define a function $s_k^n = \phi(\mathbf{R}_k^n, \mathbf{y}_n^x)$ as the likelihood measure of the decision profile \mathbf{y}_n^x w.r.t classifier C_n and class k . It measures the evidence that supports the hypothesis θ_k , while other measures $\{s_j^n\}$ where $j \neq k$ jointly represent the evidence which opposes θ_k or supports its negation $\bar{\theta}_k$.

Because the combined evidence $e_k(\mathbf{y}^x)$ finally indicates how likely the given instance x is of class k , we let s_k^n reflect two aspects of instance x , $y_n^x(k)$ and $\|\mathbf{R}_k^n - \mathbf{y}_n^x\|$, w.r.t class k . The former is the posterior probability output by classifier C_n while the latter defines the proximity between the posterior probability profile and the reference output profile of class k for classifier C_n . The norm in $\|\mathbf{R}_k^n - \mathbf{y}_n^x\|$ can be any form, such as L2-norm. Therefore, s_k^n can be defined as follows,

$$s_k^n = \frac{y_n^x(k) \cdot \exp(-\|\mathbf{R}_k^n - \mathbf{y}_n^x\|)}{\sum_{i=1}^K y_n^x(i) \exp(-\|\mathbf{R}_i^n - \mathbf{y}_n^x\|)} \tag{10}$$

The likelihood s_k^n can be treated as a simple support function with focus θ_k and its value is just the supporting degree for the focus. Therefore, the BBAs of classifier C_n for focus θ_k specific to class k can be defined as

$$m_k^n(\theta_k) = s_k^n, m_k^n(\bar{\theta}_k) = 0, m_k^n(\Theta) = 1 - s_k^n \tag{11}$$

On the opposite side, there are multiple measures $\{s_j^n\}$ where $j \neq k$ jointly measure how well the evidence $\bar{\theta}_k$ is supported. Thus, a combination of the simple support functions accounting for $\{s_j^n\}$ with focus $\bar{\theta}_k$ is need to define the BBAs. Obviously, the combined degree of support is $1 - \prod_{j \neq k} (1 - s_j^n)$. Thus, the corresponding BBAs of classifier C_n for focus $\bar{\theta}_k$ specific to class $j \neq k$ are

$$m_k^n(\theta_k) = 0, m_k^n(\bar{\theta}_k) = 1 - \prod_{i \neq k} (1 - s_i^n), m_k^n(\Theta) = \prod_{j \neq k} (1 - s_j^n). \quad (12)$$

Then, according to the combination rule of BBA (Equation 9), the evidence $e_k(\mathbf{y}_n^x) = m^n = m_k^n \oplus m_k^n$ supporting θ_k with respect to classifier C_n and class k is defined as:

$$e_k(\mathbf{y}_n^x) = \frac{s_k^n \cdot \prod_{j \neq k} (1 - s_j^n)}{1 - s_k^n \cdot \left[1 - \prod_{j \neq k} (1 - s_j^n)\right]} \quad (13)$$

Last, the combination of the evidences generated by N classifiers w.r.t class k is defined as

$$e_k(\mathbf{y}^x) = Z \prod_1^N w_k(n) e_k(\mathbf{y}_n^x), \quad (14)$$

where Z is the normalizing constant, and $w_k(n)$ is the weight of classifier C_n for class k among all the classifiers and is defined as

$$w_k(n) = \left| r_k^n(k) - \sum_{j \neq k} r_j^n(k) \right| \cdot \left| r_k^n(k) - \sum_{j \neq k} r_k^n(j) \right|. \quad (15)$$

In details, $w_k(n)$ consists of a reference-between class specificity and a reference-within class specificity w.r.t class k and classifier C_n , where $r_p^n(q)$ is the q -th element of the reference profile w.r.t class p generated by classifier C_n . The first specificity term indicates how dominant the reference value of class k in the reference profile of class k is to those in the reference profiles of other classes, while the second one reflects how dominant the reference value of class k is to other values in the reference profile of class k .

In summary, when combining the outputs of multiple classifiers for an unlabeled instance under DS theory, our approach, LCM-DS, considers three aspects, including its direct outputs of classifiers (posterior probabilities), the difference (or proximity) between its outputs and the reference outputs of the training instances, and the class-specific weights w.r.t classifiers.

Declaration. A preliminary version of this work has been published as an extended abstract (DOI: 10.1109/BIBM.2016.7822571).

Data availability. Both the dataset and the codes of our LCM-DS can be freely downloaded from <https://github.com/JustinShi2016/ScientificReports2018>.

Experiments and Results

Settings. To validate the effectiveness of our approach, we adopted the DDI dataset in Zhang *et al.*'s work²⁸, which contains 569 drugs and 52,416 pairwise interactions between them. The original work also provides three similarity matrices, derived from PubChem fingerprints of drug chemical structures³⁷, a set of keywords of side effect recorded in SIDER³⁸, as well as a list of medical terms of off-label side effects³⁹ respectively. More details are shown in the original work²⁸. We directly adopted their average as the final similarity matrix, which is used to train predictive models.

Though there are several implementations of SVM, we selected LibSVM³⁰ because of its fast running as well as convenient usage. By regarding the similarity matrix as the pre-computed kernel matrix, we have only one tunable parameter, the cost C , of LibSVM. We investigated how C influences the prediction by simply tuning its value from a recommend list {0.125, 0.25, 0.5, 1, 2, 4, 8, 16, 32, 64, and 128} one by one. The predictions of DDI under 50% hold-out cross validation(CV) with 50 repetitions showed that the value of C doesn't influence the prediction substantially, i.e., the method is quite robust in terms of these variations. For simplicity, we set $C = 1$ when training a LibSVM model in all the subsequent experiments. Likewise, we set the regularization parameter $\alpha = 0.5$ in RLS, and set the number of nearest neighbors $K = 5$ in MLKNN²³. In addition, we adopted the L2-norm when calculating the proximity measure in Equation 10.

We adopted the Area Under the Precision-Recall curves (AUPR) as the measuring metric for DDI prediction, because the number of drugs interacting with d_j (positive instances) is significantly less than that of drugs not interacting with d_j (negative instances) in each d_j -specific classification. In such a case, AUPR performs a greater penalty on highly-scored false positive instances^{40,41} that the Area Under the Receiver Operating Characteristic Curve (AUC), which tends to generate an over-optimistic measure.

Comparison between LCM and state-of-the-art approaches. We first made a fair comparison with three state-of-the-art approaches, GCM²⁷, NS²⁵ and LP²⁸. During the comparison, we performed the exactly same rounds of hold-out CV as those used by Zhang *et al.*²⁸. In each round of hold-out CV, a fixed percentage (e.g. 25% hold-out ratio) of drugs were randomly selected as the testing drugs and all the DDIs associated with them are removed as well for validation. The remaining drugs were used as the training drugs and their pairwise DDIs were used to train predictive approaches. A toy diagram of hold-out CV is shown in Fig. 3. In addition, since GCM uses SVM as its classifier, we adopted SVM when implementing our LCM.

Approaches	Hold-out Ratio				
	15%	25%	50%	75%	85%
GCM (SVM)	0.5902	0.5871	0.5796	0.5776	0.5721
NS	0.7010	0.6992	0.6955	0.6986	0.6899
LP	0.7292	0.7282	0.7052	0.6736	0.6501
LCM (SVM)	0.8078	0.8046	0.7824	0.7376	0.7117

Table 1. Comparison under different ratios of hold-out cross validation (AUPR).

	d ₁	d ₂	d ₃	d ₄	d ₅	d ₆	d ₇
d ₁		1		1			
d ₂	1		1				
d ₃		1		1		1	
d ₄	1		1		1		1
d ₅				1			
d ₆			1				
d ₇				1			
x	?	?	?	?	?	?	?

Figure 3. Illustration of hold-out cross validation. Eight drugs having known DDIs are randomly split into a training set and a testing set. The former contains seven training drugs (denoted as d₁, d₂, ..., d₇) while the latter contains only one testing drug x. The pairwise DDIs between the training drugs are organized into an interaction matrix, of which the cells marked with '1's denote the interactions between the training drugs and the cells without mark denote non-interactions. The real interactions between 'x' and the training drugs are removed and marked with '?' (See also Fig. 1). In the round of CV, the task is to deduce how possibly the testing x interacts with the training drugs one by one. The procedure is repeated until all the eight drugs are taken as the testing drugs in turn.

Such a round of CV with a specific hold-out ratio was repeated 50 times under 50 different random seeds²⁸ and its result over the 50 repetitions was reported by the average of AUPR values, which were measured in all the rounds of the hold-out CV respectively. Totally, we performed five rounds of CV under 15%, 25%, 50%, 75% and 85% hold-out ratios (Table 1). The comparison reveals two observations: (1) all the local models, including NS, LP, and LCM, are better than GCM, because local models contain the topological information of DDI network whereas GCM does not; (2) LCM is remarkably superior to those state-of-the-art approaches with 6~22% improvement in terms of AUPR.

Secondly, to elucidate LCM's advantage further, we compared our LCM with GCM, of which both apply SVM to perform DDI prediction, in terms of training time. We run LCM and GCM²⁷ under those different hold-out ratios respectively in the computer, which is equipped with Intel 4700MQ (2.40 GHz) and 64-bit Windows 7 (Home Premium). Considering the fact that running of GCM cannot be performed with too many training instances (116,886 and 90,951 respectively) in the cases of 15% and 25% hold-out ratios, we randomly sampled the same number of training instances (40,470) as that in the case of 50% hold-out. Hence, the running time in such two scenarios of hold-out CV is approximately the same as that in the scenario of 50% hold-out CV. The results listed in Table 2 show that LCM runs significantly faster than GCM (with the same classifier, SVM), even subsampling is adopted.

We further made a theoretical investigation about computational complexity. The computational complexity of SVM falls into the range of $[O(n^2), O(n^3)]$, where n is the number of training instances. For m known drugs, GCM takes $m(m+1)/2$ drug pairs as the training instances, while our approach only takes m drugs as the training instances in each of m classification tasks. Thus, the computational complexity of GCM is larger and in the range of the closed interval $[O(m^4), O(m^6)]$. In contrast, the computational complexity of LCM is bounded by $[O(m \times m^2), O(m \times m^3)]$. Therefore, in terms of computational complexity, LCM outperforms GCM.

Thirdly, to illuminate why LCM achieve the better prediction than NS and LP, we performed an additional investigation by leave-one-out cross-validation (LOOCV). We took one drug as the only one testing drug and

Approaches	Hold-out Ratio				
	15%	25%	50%	75%	85%
GCM (SVM)	50.195	48.989	49.078	23.843	4.044
LCM (SVM)	7.954	6.356	2.047	0.349	0.168

Table 2. Comparison between LCM and GCM according to running time (seconds).

the remaining drugs as the training drugs in each round of LOOCV. For the known drug d_i of interest, we first ranked the testing drug d_x by its predicted score, which indicates how likely d_x interacts with d_i . For m known drugs ($m = 568$ here), d_x obtains m predicted scores. The higher the score is, the lower the value of rank is, and the higher the occurring chance of a DDI is. Usually, the top- n ranks of drug pairs are regarded as potential DDIs. We then calculated the correlation between the ranks and the degrees of all the known drugs. For drug d_i , its degree is the number of other known drugs interacting with it. Finally, we repeated the procedure until each of the drugs were taken as the testing drug in turn and recorded the average value of the correlations obtained in all the rounds of LOOCV.

If such a correlation is significantly high, we say that the predictive model can be replaced by a degree only-based model. Thus, we investigated whether the ranks achieved by the predicting approaches are strongly correlated with the degrees of drug nodes in a DDI network. Considering that the relationship between the rank and the degree could be non-linear, we adopted Spearman's correlation to assess it. Our investigation shows that the Spearman correlations of NS and LP are up to 0.998 and 0.983, whereas that of our LCM is 0.851. The extremely high correlations (>0.98) of both NS and LP indicate that they tend to recommend those drugs having many known DDIs as the interacting partners for a newly queried drug. The comparison reveals that the prediction achieved by a degree-related model would be greatly approximate to those achieved by NS and LP, but significantly different from that achieved by LCM.

The underlying reason is that both NS and LP involve the multiplication, which is correlated to the sum of the pairwise similarities between other existing drugs interacting with d_i . As a result, their predictions are dependent on the number of positive instances (existing drugs interacting with d_i) when predicting how possibly a newly-given drug d_x interacts with an existing drug d_i . As a result, both of them have the degree-induced bias that leads their prediction to tend to rank the pairs between a newly-given drug and the drugs having many DDIs with top priorities. By contrast, the multiplication involved in LCM is usually related to the similarity matrix and a few of instances supporting the discriminate boundary in the case of SVM. Consequently, LCM only depends on the positive instances and the negative instances, of which both are located on the discriminate boundary, such that it is able to relax or minimize this bias.

Furthermore, we made a case study to show how the bias affects the prediction and demonstrate the ability of our LCM to relax the bias. We focused on the drug 'Amoxapine' which interacts with 7 known drugs having meanwhile different numbers of DDIs. We removed the interactions of 'Amoxapine' and predicted its interacting drugs. In an ideal prediction, it is anticipated that the ranks of the drug pairs between 'Amoxapine' and its interacting partners should be ≤ 7 . We then extracted two of its interacting partner drugs, 'Paroxetine' and 'Fluvoxamine', which have the most and the least numbers of DDIs (444 and 101) respectively, and checked the real prediction achieved by NS, LP, and LCM. For the pair of 'Paroxetine' and 'Amoxapine', NS and LP generate rank 25 and rank 22 respectively whereas our LCM gives rank 4. Thus, our LCM generates the correct prediction (rank < 7) but they cannot. For the pair of 'Fluvoxamine' and 'Amoxapine', NS and LP give 366 and 361 whereas our LCM gives 204. Even all these approach cannot give a correct prediction, our LCM still gives a significantly higher rank than both NS and LP for the queried drug pairs. Similar predictions were able to be found in other cases. Consequently, our LCM is able to relax such a degree-induced bias.

Validation of LCM-DS. In this section, we shall first show how the factors, including the posterior probabilities (\mathbf{y}_n^x) directly output by a classifier, the proximity between them and the reference profiles ($\|\mathbf{R}_k^n - \mathbf{y}_n^x\|$) of the training instances, and the classifier weight ($w_k(n)$), affect the performance of LCM-DS respectively.

To investigate the influence of these three factors, we built three variants of LCM-DS, of which each variant has the lack of a unique factor respectively. Then, we run and compared them with the regular LCM-DS (Fig. 4). The comparison shows that the lack of any of them decrease the predicting performance and the absence of the posterior probability factor causes the biggest decrement.

We made a case study to demonstrate the importance of the factors. Two drugs, 'Prostacyclin' and 'Amikacin', were chosen to investigate the predicted scores, which indicate how likely these two drugs interact with the training drugs. We sorted the predicted scores to rank the drug pairs, in which their partners are the training drugs, and reported the average ranks of the positively labeled drug pairs (Table 3). The less, the better.

Three observations on these two drugs can be drawn: (1) all these factors contribute to the prediction because the absence of any of them increase the average ranks of DDIs for the selected drugs; (2) the factor of post-probability plays, as anticipant, the most important roles in LCM-DS because its absence causes the biggest increment of average ranks; and (3) LCM-DS integrating them achieves the best performance because it generates the smallest average ranks. Totally, the comparison demonstrates that LCM-DS is an effective fusion rule, which is able to integrate all the individual factors contributing to the prediction to obtain a better prediction.

Moreover, we made a deeper investigation on LCM-DS by comparing it with both its member classifiers and classical fusion rules. The member classifiers are MLKNN, RLS, and SVM. The classical fusion rules include four unsupervised fusion rules (i.e. Average, Maximum, Product, Minimum) and one supervised fusion rule (Decision

Drug	Average Rank			
	LCM-DS	No Post-probability	No proximity	No Classifier Weight
Prostacyclin	90.16	104.94	92.23	94.39
Amikacin	52.79	60.45	58.80	53.23

Table 3. Comparison between LCM-DS and its variants in terms of average rank.

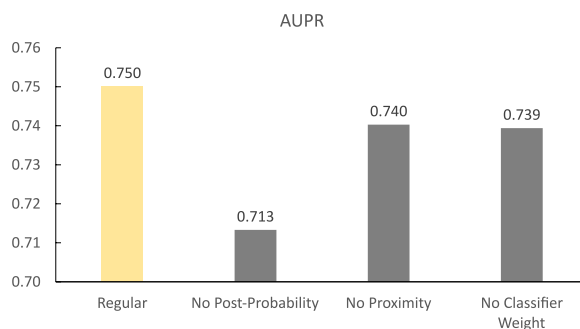


Figure 4. The influence of three factors in LCM-DS under the 85% ratio of Hold-Out Cross Validation with 50 Repetitions. The first label, 'Regular', denotes the regular LCM-DS without removing any factors. The other labels denote the variants of LCM-DS, which have no post-probability factor, no proximity factor and no classifier weight factor respectively.

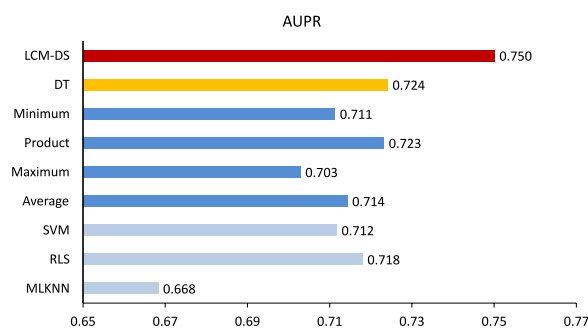


Figure 5. Comparison with Individual Classifiers and Classical Fusion Rules under the 85% ratio of Hold-Out Cross Validation with 50 Repetitions.

Template, DT³²). Three individual classifiers were implemented under the framework of LCM and integrated into LCM-DS. In details, the similarity-based version of MLKNN was originally implemented by our previous work²³, which developed an approach for predicting drug-target interactions. RLS was directly implemented by Octave codes. The implementation of SVM was by compiling and building the source codes of LibSVM³⁰ into the Octave interface. All the fusion rules were also implemented by Octave codes. See also Section 2.3 for more technical details about the classifiers and see also Section 2.4 for more technical details about the fusion rules. We performed 85% hold-out CV again in the comparison (Fig. 5).

The comparison demonstrates that (1) the performance of individual classifiers varies and RLS achieves the best classifiers in this case of hold-out CV; (2) Former fusion rules may (e.g. Product, DT) or may not (e.g. Average, Maximum, Minimum) outperform individual classifiers; (3) DS wins the best among both member classifiers and classical fusion rules with the significant improvement. In summary, the proposed supervised DS-based fusion rule is effective.

Discussion

DDIs frequently induce adverse drug reactions or occasionally facilitate better drug co-prescriptions. DDI identification before making clinical medications is critical but bears a high cost in clinics. Computational approaches have exhibited their ability on screening DDI candidates among a large number of drug pairs by utilizing preliminary characteristics of drugs. However, global model-based approaches are usually slow and neglect the topological structure of a DDI network, while local model-based models have the degree-induced bias.

To address these two issues, we have presented a novel local classification-based model (LCM) in the scenario of predicting DDI candidates for new drugs, which have no existing DDI with known drugs. For a specific drug

having known DDIs, an LCM treats drugs having and having no interaction with it as positives and negative instances respectively, and trains a set of small-size classifiers to discriminate how likely a newly-given drug interacts with the drug of interest. Compared with the global classification-based model, LCM shows the advantages of theoretically faster running and practically better performance. Compared with two other local model-based approaches (naïve similarity-based and label propagation-based approaches), LCM is able to relax their intrinsic bias because the prediction for a new drug depends on the distributions or discriminant boundaries of positive instances and negative instances in the feature/similarity space.

More importantly, to address the issue that computational approaches lack an effective ensemble method to combine results from multiple predictors, we have designed a novel supervised fusion algorithm (LCM-DS) to aggregate the outputs of multiple classifiers for an unlabeled instance based on the Dempster-Shafer theory of evidence. Our LCM-DS integrates three factors from multiple classifiers, including the posterior probabilities output by individual classifiers, the proximity between the decision profiles of given instances and the reference profiles, as well as the quality of the reference profiles, which jointly contribute to the final decision.

Finally, both the experiments of DDI prediction and the case study demonstrate that the present LCM outperforms three state-of-the-art approaches, including one global model-based approach and two local model-based approaches, and its fusion version, LCM-DS, is superior to both all of its member classifiers and five classical fusion algorithms.

In the coming future, we shall improve our approaches in two aspects. First, LCM is of a supervised learning model, which treats unknown drug pairs as negative instances. In fact, a few of unknown drug pairs could be DDIs. Thus, a semi-supervised learning model⁶ or a one-class learning model should be considered. Secondly, other pre-existing knowledge should be considered in the proposed LCM-DS. Especially, the essence of DDI is strongly correlated with drug-binding proteins, such as drug targets and enzymes, which attend in different pathways. Thus, the integration of drug target-based^{21,22,42,43} and/or pathway-based⁷ similarities into the current similarities would be helpful to improve DDI prediction and even to reveal the underlying mechanism of DDI occurrence. In addition, because our LCM-DS actually provides an effective framework for combining decisions from different pre-existing sources, it can be easily applied in similar areas (e.g. lncRNA-disease association prediction⁴⁴).

References

- Wienkers, L. C. & Heath, T. G. Predicting *in vivo* drug interactions from *in vitro* drug discovery data. *Nature reviews. Drug discovery* **4**, 825–833, <https://doi.org/10.1038/nrd1851> (2005).
- Leape, L. L. *et al.* Systems analysis of adverse drug events. ADE Prevention Study Group. *Jama* **274**, 35–43 (1995).
- Businaro, R. Why we need an efficient and careful pharmacovigilance. *J. Pharmacovigilance* **1**, 4 (2013).
- Karbownik, A. *et al.* Pharmacokinetic drug-drug interaction between erlotinib and paracetamol: A potential risk for clinical practice. *Eur J Pharm Sci.* **102**, 55–62 (2017).
- Mulroy, E., Highton, J. & Jordan, S. Giant cell arteritis treatment failure resulting from probable steroid/antiepileptic drug-drug interaction. *N Z Med J.* **130**, 102–104 (2017).
- Chen, X. *et al.* NLLSS: Predicting Synergistic Drug Combinations Based on Semi-supervised Learning. *PLoS computational biology* **12**, e1004975, <https://doi.org/10.1371/journal.pcbi.1004975> (2016).
- Sun, X., Bao, J., You, Z., Chen, X. & Cui, J. Modeling of signaling crosstalk-mediated drug resistance and its implications on drug combination. *Oncotarget* **7**, 63995–64006, <https://doi.org/10.18632/oncotarget.11745> (2016).
- Veith, H. *et al.* Comprehensive characterization of cytochrome P450 isozyme selectivity across chemical libraries. *Nat Biotechnol* **27**, 1050–1055, <https://doi.org/10.1038/nbt.1581> (2009).
- Huang, S. M., Temple, R., Throckmorton, D. C. & Lesko, L. J. Drug interaction studies: study design, data analysis, and implications for dosing and labeling. *Clin Pharmacol Ther* **81**, 298–304, <https://doi.org/10.1038/sj.cpt.6100054> (2007).
- Wiśniowska, B. & Polak, S. The Role of Interaction Model in Simulation of Drug Interactions and QT Prolongation. *Curr Pharmacol Rep.* **2**, 339–344 (2016).
- Zhou, D., Bui, K., Sostek, M. & Al-Huniti, N. Simulation and Prediction of the Drug-Drug Interaction Potential of Naloxegol by Physiologically Based Pharmacokinetic Modeling. *CPT: pharmacometrics & systems pharmacology* **5**, 250–257, <https://doi.org/10.1002/psp4.12070> (2016).
- Yu, H. *et al.* Predicting and Understanding Comprehensive Drug-Drug Interactions via Semi-Nonnegative Matrix Factorization. *Bmc Syst Biol* **12** (2018).
- Bui, Q. C., Sloot, P. M., van Mulligen, E. M. & Kors, J. A. A novel feature-based approach to extract drug-drug interactions from biomedical text. *Bioinformatics* **30**, 3365–3371, <https://doi.org/10.1093/bioinformatics/btu557> (2014).
- Ben Abacha, A. *et al.* Text mining for pharmacovigilance: Using machine learning for drug name recognition and drug-drug interaction extraction and classification. *Journal of biomedical informatics* **58**, 122–132, <https://doi.org/10.1016/j.jbi.2015.09.015> (2015).
- Zhang, Y. *et al.* Leveraging syntactic and semantic graph kernels to extract pharmacokinetic drug drug interactions from biomedical literature. *Bmc Syst Biol* **10**(Suppl 3), 67, <https://doi.org/10.1186/s12918-016-0311-2> (2016).
- Duke, J. D. *et al.* Literature Based Drug Interaction Prediction with Clinical Assessment Using Electronic Medical Records: Novel Myopathy Associated Drug Interactions. *PLoS computational biology* **8** (2012).
- Li, J. *et al.* Identification of high-quality cancer prognostic markers and metastasis network modules. *Nature communications* **1** (2010).
- McGee, S. R., Tibiche, C., Trifiro, M. & Wang, E. Network Analysis Reveals A Signaling Regulatory Loop in the PIK3CA-mutated Breast Cancer Predicting Survival Outcome. *Genomics, proteomics & bioinformatics* **15**, 121–129, <https://doi.org/10.1016/j.gpb.2017.02.002> (2017).
- Wang, E. *et al.* Predictive genomics: A cancer hallmark network framework for predicting tumor clinical phenotypes using genome sequencing data. *Semin Cancer Biol* **30**, 4–12 (2015).
- Gao, S. *et al.* Identification and Construction of Combinatory Cancer Hallmark-Based Gene Signature Sets to Predict Recurrence and Chemotherapy Benefit in Stage II Colorectal Cancer. *JAMA oncology* **2**, 37–45, <https://doi.org/10.1001/jamaoncol.2015.3413> (2016).
- Chen, X., Liu, M. X. & Yan, G. Y. Drug-target interaction prediction by random walk on the heterogeneous network. *Molecular bioSystems* **8**, 1970–1978, <https://doi.org/10.1039/c2mb00002d> (2012).
- Chen, X. *et al.* Drug-target interaction prediction: databases, web servers and computational models. *Briefings in bioinformatics*, <https://doi.org/10.1093/bib/bbv066> (2015).

23. Shi, J. Y., Yiu, S. M., Li, Y. M., Leung, H. C. M. & Chin, F. Y. L. Predicting drug-target interaction for new drugs using enhanced similarity measures and super-target clustering. *Methods* **83**, 98–104, <https://doi.org/10.1016/j.ymeth.2015.04.036> (2015).
24. Shi, J.-Y., Li, J.-X. & Lu, H.-M. Predicting existing targets for new drugs base on strategies for missing interactions. *Bmc Bioinformatics* **17**, 282, <https://doi.org/10.1186/s12859-016-1118-2> (2016).
25. Vilar, S. *et al.* Similarity-based modeling in large-scale prediction of drug-drug interactions. *Nature protocols* **9**, 2147–2163, <https://doi.org/10.1038/nprot.2014.151> (2014).
26. Luo, H. *et al.* DDI-CPI, a server that predicts drug-drug interactions through implementing the chemical-protein interactome. *Nucleic Acids Res* **42**, 46–52 (2014).
27. Cheng, F. & Zhao, Z. Machine learning-based prediction of drug-drug interactions by integrating drug phenotypic, therapeutic, chemical, and genomic properties. *Journal of the American Medical Informatics Association: JAMIA* **21**, e278–286, <https://doi.org/10.1136/amiainl-2013-002512> (2014).
28. Zhang, P., Wang, F., Hu, J. & Sorrentino, R. Label Propagation Prediction of Drug-Drug Interactions Based on Clinical Side Effects. *Sci Rep* **5**, 12339, <https://doi.org/10.1038/srep12339> (2015).
29. Ancona, N., Maglietta, R., D'Addabbo, A., Liuni, S. & Pesole, G. Regularized Least Squares Cancer classifiers from DNA microarray data. *Bmc Bioinformatics* **6**(Suppl 4), S2, <https://doi.org/10.1186/1471-2105-6-S4-S2> (2005).
30. Chang, C.-C. & Lin, C.-J. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* **2**(27), 21–27 (2011). 27.
31. Zhang, M. L. & Zhou, Z. H. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recogn* **40**, 2038–2048 (2007).
32. Kuncheva, L. I., Bezdek, J. C. & Duin, R. P. W. Decision templates for multiple classifier fusion: an experimental comparison. *Pattern Recogn* **34**, 299–314 (2001).
33. Yan, X. Y. & Zhang, S. W. Identifying drug-target interactions with decision template. *Current Protein & Peptide Science* **18**, <https://doi.org/10.2174/1389203718666161108101118> (2017).
34. Singh, R. & Murad, W. Protein disulfide topology determination through the fusion of mass spectrometric analysis and sequence-based prediction using Dempster-Shafer theory. *Bmc Bioinformatics* **14**(Suppl 2), S20, <https://doi.org/10.1186/1471-2105-14-S2-S20> (2013).
35. Rogova, G. Combining the results of several neural network classifiers. *Neural Networks* **7**, 777–781 (1994).
36. Mandler, E. & Schümann, J. Combining the Classification Results of Independent Classifiers Based on the Dempster/Shافر Theory of Evidence. *Machine Intelligence and Pattern Recognition* **7**, 381–393 (1988).
37. Fu, G. *et al.* PubChemRDF: towards the semantic annotation of PubChem compound and substance databases. *Journal of cheminformatics* **7** (2015).
38. Kuhn, M., Campillos, M., Letunic, I., Jensen, L. J. & Bork, P. A side effect resource to capture phenotypic effects of drugs. *Molecular systems biology* **6**, 343, <https://doi.org/10.1038/msb.2009.98> (2010).
39. Tatonetti, N. P., Ye, P. P., Daneshjou, R. & Altman, R. B. Data-driven prediction of drug effects and interactions. *Science translational medicine* **4**, 125ra131, <https://doi.org/10.1126/scitranslmed.3003377> (2012).
40. Davis, J. & Goadrich, M. In *the 23rd international conference on Machine learning* 233–240 (2006).
41. Jiao, Y. & Du, P. Performance measures in evaluating machine learning based bioinformatics predictors for classifications. *Quantitative Biology* **4**, 320–330 (2016).
42. Shi, J.-Y., Liu, Z., Yu, H. & Li, Y.-J. Predicting Drug-Target Interactions via Within-Score and Between-Score. *BioMed Research International* **2015**, Article ID 350983, 9 pages, <https://doi.org/10.1155/2015/350983> (2015).
43. Shi, J.-Y., Li, J.-X., Chen, B.-L. & Zhang, Y. Inferring Interactions between Novel Drugs and Novel Targets via Instance-Neighborhood-Based Models. *Current Protein & Peptide Science* **19**, 488–497 (2018).
44. Shi, J. Y., Huang, H., Zhang, Y. N., Long, Y. X. & Yiu, S. M. Predicting binary, discrete and continued lncRNA-disease associations via a unified framework based on graph regression. *BMC medical genomics* **10**, 65, <https://doi.org/10.1186/s12920-017-0305-y> (2017).

Acknowledgements

This work was supported by RGC Collaborative Research Fund (CRF) of Hong Kong (C1008-16G), National High Technology Research and Development Program of China (No. 2015AA016008), the Fundamental Research Funds for the Central Universities of China (No. 3102015ZY081), the Program of Peak Experience of NWPU (2016), the Seed Foundation of Innovation and Creation for Graduate Students in Northwestern Polytechnical University (No. ZZ2018170, ZZ2018235), China National Training Programs of Innovation and Entrepreneurship for Undergraduates (No. 201710699330) and partially supported by the National Natural Science Foundation of China (No. 61473232, 91430111).

Author Contributions

J.Y.S. and S.W.Z. conceived and designed the experiments; X.Q.S. and K.G. collected the data and performed the experiments; J.Y.S. analyzed the data and contributed codes; J.Y.S. and S.M.Y. wrote the paper. All authors read and approved the final manuscript.

Additional Information

Competing Interests: The authors declare no competing interests.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2018