BMC Bioinformatics

PROCEEDINGS Open Access

ProDis-ContSHC: learning protein dissimilarity measures and hierarchical context coherently for protein-protein comparison in protein database retrieval

Jingyan Wang[1,2], Xin Gao[1*], Quanquan Wang[2], Yongping Li[2,3]

From The 2011 International Conference on Intelligent Computing (ICIC 2011)
Zhengzhou, China. 11-14 August 2011

Abstract

Background: The need to retrieve or classify protein molecules using structure or sequence-based similarity measures underlies a wide range of biomedical applications. Traditional protein search methods rely on a pairwise dissimilarity/similarity measure for comparing a pair of proteins. This kind of pairwise measures suffer from the limitation of neglecting the distribution of other proteins and thus cannot satisfy the need for high accuracy of the retrieval systems. Recent work in the machine learning community has shown that exploiting the global structure of the database and learning the contextual dissimilarity/similarity measures can improve the retrieval performance significantly. However, most existing contextual dissimilarity/similarity learning algorithms work in an unsupervised manner, which does not utilize the information of the known class labels of proteins in the database.

Results: In this paper, we propose a novel protein-protein dissimilarity learning algorithm, ProDis-ContSHC. ProDis-ContSHC regularizes an existing dissimilarity measure d_{ij} by considering the contextual information of the proteins. The context of a protein is defined by its neighboring proteins. The basic idea is, for a pair of proteins (i, j), if their context $\mathcal{N}(i)$ and $\mathcal{N}(j)$ is similar to each other, the two proteins should also have a high similarity. We implement this idea by regularizing d_{ij} by a factor learned from the context $\mathcal{N}(i)$ and $\mathcal{N}(j)$.
Moreover, we divide the context to hierarchial sub-context and get the contextual dissimilarity vector for each protein pair. Using the class label information of the proteins, we select the relevant (a pair of proteins that has the same class labels) and irrelevant (with different labels) protein pairs, and train an SVM model to distinguish between their contextual dissimilarity vectors. The SVM model is further used to learn a supervised regularizing factor. Finally, with the new **S**upervised learned **Dis**similarity measure, we update the **Pro**tein **H**ierarchial **Cont**ext **C**oherently in an iterative algorithm–**ProDis-ContSHC**.
We test the performance of ProDis-ContSHC on two benchmark sets, i.e., the ASTRAL 1.73 database and the FSSP/DALI database. Experimental results demonstrate that plugging our supervised contextual dissimilarity measures into the retrieval systems significantly outperforms the context-free dissimilarity/similarity measures and other unsupervised contextual dissimilarity measures that do not use the class label information.

Conclusions: Using the contextual proteins with their class labels in the database, we can improve the accuracy of the pairwise dissimilarity/similarity measures dramatically for the protein retrieval tasks. In this work, for the first time, we propose the idea of supervised contextual dissimilarity learning, resulting in the ProDis-ContSHC

* Correspondence: xin.gao@kaust.edu.sa
[1] King Abdullah University of Science and Technology (KAUST), Mathematical and Computer Sciences and Engineering Division, Thuwal, 23955-6900, Saudi Arabia
Full list of author information is available at the end of the article

algorithm. Among different contextual dissimilarity learning approaches that can be used to compare a pair of proteins, ProDis-ContSHC provides the highest accuracy. Finally, ProDis-ContSHC compares favorably with other methods reported in the recent literature.

Background

Proteins are linear chains of amino acids. The polypeptide chains are folded into complicated three-dimensional (3D) structures. With different structures, proteins are able to perform specific functions in biological processes [1-14]. To study the structure-function relationship, biologists have a high demand on protein structure retrieval systems for searching similar sequences or 3D structures [15]. Protein pairwise comparison is one of the main functions of such retrieval systems [16]. The need to retrieve or classify proteins using 3D structure or sequence-based similarity underlies many biomedical applications. In drug discovery, researchers search for proteins that share specific chemical properties as sources for new treatment. In folding simulations, similar intermediate structures might be indicative of a common folding pathway [17].

Related work

The structural comparison problem in a protein structure retrieval system has been extensively studied. In [18], a rapid protein structure retrieval system named ProtDex2 was proposed by Aung and Tan [18] , in which they adopted the information retrieval techniques to perform rapid database search without accessing to each 3D structure in the database. The retrieval process was based on the inverted-file index constructed on the feature vectors of the relationship between the secondary structure elements (SSEs) of all the protein structures in the database. In order to evaluate the similarity score between a query protein structure and a protein structure in the database, they adopted and modified the well-known *$\Sigma(tf \times idf)$ scoring scheme* commonly used in document retrieval systems [19]. In [20,21], a 3D shape-based approach was presented by Daras et al. The method relied primarily on the geometric 3D structure of the proteins, which was produced from the corresponding PDB files, and secondarily on their primary and secondary structures. Additionally, characteristic attributes of the primary and secondary structures of the protein molecules were extracted, forming attribute-based descriptor vectors. The descriptor vectors were then weighted and an integrated descriptor vector was produced. To compare a pair of protein descriptor vectors, Daras et al. [20,21] used two metrics of similarity. The first one was based on the *Euclidean distance* [22] between the descriptor vectors, and the second one was based on *Mean Euclidean Distance Measure* [20,21].

Later, Marsolo and Parthasarathy presented two normalized, stand-alone representations of proteins that enabled fast and efficient object retrieval based on sequence or structure information [17,23]. For the range queries, they specified a range value r and retrieved all the proteins from the database which lied within a distance r to the query. In their work, distance referred to the standard *Euclidean distance* [22]. In [24], Sael et al. introduced a global surface shape representation by 3D Zernike descriptors for protein structure similarity search. In their study, three distance measures were used for comparing 3D Zernike descriptors of protein surface shapes, i.e., *Euclidean distance*, *Manhattan distance* [25], and *correlation coefficient-based distance*. A fast protein comparison algorithm IR Tableau was developed by Zhang et al. for protein retrieval purposes in [26], which leveraged the tableau representation to compare protein tertiary structures. IR tableau compared tableaux using feature indexing techniques. In IR Tableau [26], a number of similarity functions were applied for comparing a pair of protein vectors, i.e., *cosine similarity* [27], *Jaccard index* [28], *Tanimoto coefficient* [29], and *Euclidean distance*.

The basic components of a protein retrieval system includes a way to represent proteins and a dissimilarity measure that compares a pair of proteins. Most of the aforementioned studies focus on the feature representation of the proteins, while neglecting the comparison of the feature vectors. Such studies usually apply a simple similarity or dissimilarity measure for the comparison of the feature vectors, such as Euclidean Distance Measure used in [17,20,21,23,24,26]. Most of the existing protein comparison techniques suffer from the following two bottlenecks:

- The dissimilarity measure is a pairwise distance measure, which is computed only considering the query protein x_0 and a database protein x_i as $d(x_0, x_i)$. It does not consider other proteins in the database, neglecting the effects of the contextual proteins. If we consider the distribution of the entire protein database $X = \{x_j\}$, $j = 1 \dots N$ when computing the dissimilarity as $d(x_0, x_i|X)$, the retrieval performance may benefit from the contextual proteins $\{x_j\}$, $j \neq i$.
- The dissimilarity measure is computed in an unsupervised way, which does not use the known information

of the class labels $L = \{l_j\}$, $j = 1 \ldots, N$ in the database. Although we may have no idea about whether x_0 and x_i belong to the same class (having the same folding type etc., $l_0 = l_i$) or not ($l_0 \neq l_i$), we do know some prior information about other proteins L. In all of the previous studies, prior class labels L were not adopted to calculate the dissimilarity $d(x_0, x_i)$.

Due to these two bottlenecks, traditional protein retrieval systems using pairwise and unsupervised dissimilarity measure usually do not achieve satisfactory performance, even though many effective protein feature descriptors are developed and used. In this paper, we investigate the dissimilarity measure and propose a novel learning algorithm to improve the performance of a given dissimilarity measure.

Recent research in machine learning points out that contextual information can be used to improve the dissimilarity or similarity measures. This kind of algorithms are called contextual or context-sensitive dissimilarity learning [30-34]. Unlike the traditional pairwise distance $d(x_0, x_i)$ which only considers the two refereed proteins x_0 and x_i, contextual dissimilarity also considers the contextual proteins X when computing the dissimilarity $d(x_0, x_i|X)$. The existing contextual similarity learning algorithms can mainly be classified into the following two categories:

Dissimilarity regulation

The first contextual dissimilarity measure (CDM) was proposed by Jegou et al. in [30,31]. They introduced the CDM, which significantly improved the accuracy of the image search problem. CDM measure took the local distribution of the vectors into account and iteratively estimated the distance update terms in the spirit of Sinkhorns scaling algorithm [35], thereby modified the neighborhood structure. This regularization was motivated by the observation that a good ranking was usually not symmetric in an image search system. In this paper, we will focus on this type of contextual dissimilarity learning.

Similarity transduction on graph

In [32,33], Bai et al. provided a novel perspective to the shape retrieval tasks by considering the existing shapes as a group and studying their similarity measures to the query shape in a graph structure. For a given similarity measure, a new similarity was learned through graph transduction. The learning was done in an iterative manner so that the neighbors of a given shape influenced the final similarity to the query. The basic idea is actually related to the PageRank algorithm, which forms a foundation of Google Web search. This method is further improved by Wang et al. in [36]. Similar learning algorithms were also used to rank proteins in a protein database as in [37,38]. Kuang et al. proposed a general graph-based propagation algorithm called MotifProp to detect more subtle similarity relationship than the pairwise comparison methods. In [38], Weston et al. reviewed RankProp, a ranking algorithm that exploited the global network structure of similarity relationship among proteins in a database by performing a diffusion operation on a protein similarity network with weighted edges.

The drawbacks of the above algorithms lay on two folds. On the one hand, such algorithms do not utilize the class label information of the database images L, and thus work in an unsupervised way. The only one used L is [38]. However, the algorithm proposed in [38] had basically the same framework as [32,33,37], i.e., protein label information L was only used to estimate the parameters. On the other hand, the “context” is fixed in the iterative algorithms of most of the transduction methods [32,33,37,38]. A better way is to update the context using the learned similarity measures as in [30,31].

To overcome these drawbacks, we develop a novel contextual dissimilarity learning algorithm to improve the performance of a protein retrieval system. The novel dissimilarity measure is regularized by the dissimilarity of the contextual proteins (neighboring proteins), while the contextual proteins are updated using the learned dissimilarities coherently. The basic idea comes from [39,40], which assume that if two local features in two images are similar, their context is likely to be similar. In comparison to [30,31], which use neighborhood as a single context, we partition the neighborhood into several hierarchical sub-context corresponding to the learned dissimilarities. With the sub-context, we compute the dissimilarity of sub-context of a pair of proteins and construct the hierarchial sub-contextual dissimilarity vector. Moreover, using the label information L, we select pairs of proteins belonging to the same classes $\{(x_i, x_j)|l_i = l_j\}$ as the relevant protein pairs. We also select the irrelevant protein pairs $\{(x_k, x_l)|l_k \neq l_l\}$.

Finally, we train a support vector machine (SVM) [41] to distinguish between the relevant and the irrelevant protein pairs. The output of the SVM will further be used to regularize the dissimilarity in an iterative manner.

Methods

This section describes our contextual protein-protein dissimilarity learning algorithm, which utilizes the contextual proteins and class label information of the database proteins to index and search protein structures efficiently. We will demonstrate that our idea is general in the sense that it can be used to improve the existing similarity/dissimilarity measures.

Protein structure retrieval framework

In a protein retrieval system, the query and the database proteins are firstly represented as feature vectors. Here, we denote the query protein feature vector as x_0 and database protein feature vectors as $X = \{x_1, x_2, \ldots, x_N\}$, where N is the number of proteins in the database. Then, based on a distance measure $d_{0i}= d(x_0, x_i)$, we compute the distance of x_0 and all the proteins in the database, i.e., $\{d_{01}, d_{02}, \ldots, d_{0N}\}$. The database proteins are then ranked according to the distances. The k most similar ones are returned as the retrieval results. We illustrate the outline of the protein retrieval system in Figure 1.

ProDis-ContSHC: the contextual dissimilarity learning algorithm

In this section, we will introduce the novel contextual protein-protein dissimilarity learning algorithm. We first give the definition of the hierarchical context of a protein, which will be used to compute the contextual dissimilarity and regularize the dissimilarity measure. Then a more discriminative regularization factor is learned using the class labels of the database proteins. Finally, we propose the **S**upervised regulating of **Pro**tein-protein **Dis**similarity and updating of the **H**ierarchical **Cont**ext **C**oherently in an iterative manner, resulting in the ProDis-ContSHC algorithm.

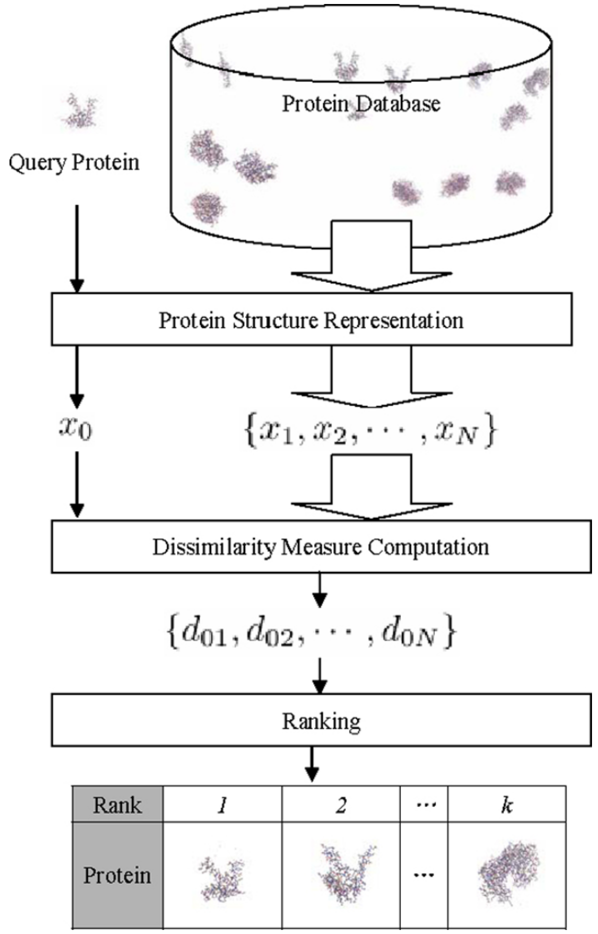
Protein Database
Query Protein
Protein Structure Representation
x_0
$\{x_1, x_2, \cdots, x_N\}$
Dissimilarity Measure Computation
$\{d_{01}, d_{02}, \cdots, d_{0N}\}$
Ranking
Rank
1
2
···
k
Protein
···

Figure 1 Flowchart of protein retrieval systems.

Using hierarchical context to regularize the dissimilarity measure

Here, we define a protein x_i's context as its K nearest neighbors $\mathcal{N}(i)$. The dissimilarity between two sets of context is measured by the contextual dissimilarity as

$$r_{ij} = \frac{1}{K^2} \sum_{m \in \mathcal{N}(i), n \in \mathcal{N}(j)} d_{mn} \tag{1}$$

The contextual dissimilarity is illustrated in Figure 2(a).

Furthermore, instead of averaging all the pairwise dissimilarities between the two context $\mathcal{N}(i)$ and $\mathcal{N}(j)$, we propose the hierarchical context by splitting the context $\mathcal{N}(i)$ to P "sub-context" $\mathcal{N}_p(i), p = \{1, \cdots, P\}$ according to their distances to x_i. To be more specific, sub-context $\mathcal{N}_p(i)$ is defined as

$$\begin{aligned} \mathcal{N}_p(i) = \{x_j | x_j \; &\textit{is among the } k' - th \textit{ to } k'' - th \\ &\textit{nearest neighbors of } x_i, \textit{ according to } \{d_{ij}\}, \\ &j \in \{1, \cdots, i-1, i+1, \cdots, N\}\} \end{aligned} \tag{2}$$

where $k' = (p - 1) \times \kappa$, $k'' = (p - 1) \times \kappa + \kappa$, κ is the size of a sub-context, and P is the number of sub-context. In this way, we can compute the contextual dissimilarity by averaging the dissimilarity of the sub-context as

$$\begin{aligned} r_{ij} &= \frac{1}{P} \sum_p \left[\frac{1}{\kappa^2} \sum_{m \in \mathcal{N}_p(i), n \in \mathcal{N}_p(j)} d_{mn} \right] \\ &= \frac{1}{P} \sum_p d_{ij}(p) \end{aligned} \tag{3}$$

where $d_{ij}(p) = \frac{1}{\kappa^2} \sum_{m \in \mathcal{N}_p(i), n \in \mathcal{N}_p(j)} d_{mn}, p = 1, \cdots, P$, is the hierarchical sub-contextual dissimilarity. Figure 2(b) illustrates the idea of sub-contextual dissimilarity.

Intuitively, if the context of two proteins is dissimilar to each other (r_{ij} is higher than the average), they should have a higher dissimilarity value, and vice versa. We implement this by multiplying a coefficient, which is the ratio of r_{ij} to the average of all the contextual dissimilarity $\bar{r} = \frac{1}{N^2} \sum_{i,j} r_{ij}$,

$$\begin{aligned} d^*_{ij} &= d_{ij} \times \frac{r_{ij}}{\bar{r}} \\ &= d_{ij} \times \delta_{ij} \end{aligned} \tag{4}$$

Here, $\delta_{ij} = \frac{r_{ij}}{\bar{r}}$ is a regularization factor for d_{ij}, with which we can improve d_{ij} by its contextual information.

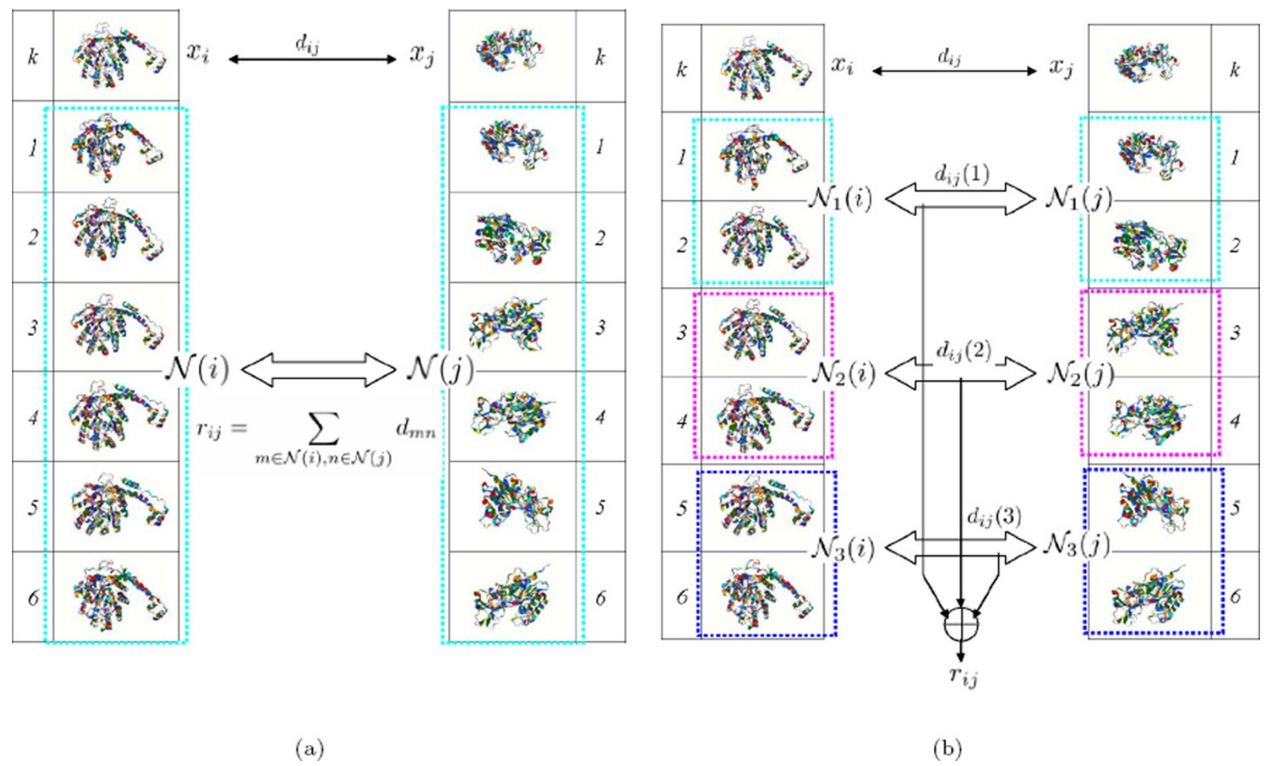

Figure 2 Illustration of context-based dissimilarity and hierarchical context-based dissimilarity. The two proteins x_i and x_j, on which the dissimilarity is to be measured, are in the first row. The nearest neighbors of these two proteins are listed below them as the context, respectively. (a) The traditional context $\mathcal{N}(i)$; (b) The proposed hierarchical context $\mathcal{N}_p(i)$, p = {1, 2, 3}.

Moreover, this procedure can be done in an iterative manner. We can use the regularized dissimilarity measure d^*_{ij} to re-define the new hierarchical context $\mathcal{N}_p(i)$. In this way, we can learn the protein-protein dissimilarity d^*_{ij} and hierarchical context $\mathcal{N}_p(i)$ coherently.

Supervised regularization factor learning

We try to utilize the label information $L = \{l_1, \dots, l_N\}$ of the database proteins to learn a better regularization factor δ_{ij}. The class information is adopted both in the intraclass and interclass dissimilarity computation to maximize the Fisher criterion [42] for protein class separability. Firstly, we can select a number of protein pairs $\{\gamma = (i, j)|i, j = 1, \dots, N\}$. For each pair, we compute the hierarchical contextual dissimilarities and organize them as a P-dimensional dissimilarity vector $\mathbf{d}_\gamma = [d_{ij}\,(1)\; d_{ij}\,(2)\; \dots\; d_{ij}\,(P)]^\top$, as shown in Figure 3. Then, inspired by the score fusion rule [43,44], using L, we further label each pair $\gamma = (i, j)$ as a relevant pair $y_\gamma = +1$ if $l_i = l_j$, or an irrelevant pair $y_\gamma = -1$ otherwise.

Now with the training samples as $\Gamma = \{(\mathbf{d}_\gamma, y_\gamma)\}$, $\gamma = 1, \dots, {}_N C_2$, we train a binary SVM [41] classifier to distinguish between the relevant pairs and the irrelevant pairs. The publicly available package SVMlight [45] is applied to implement the SVM on our training set Γ. This package allows us to optimize a number of parameters and offers the options to use different kernel functions to obtain the best classification performance [46]. The separating hyperplane generated by SVM model is given by

$$f(\mathbf{d}) = \mathbf{d} \cdot w + b \tag{5}$$

where w is a vector orthogonal to the hyperplane, and b is a parameter that minimizes $||w||^2$ and satisfies the following conditions:

$$y_\gamma(\mathbf{d}_\gamma \cdot w + b) \geq 1 \tag{6}$$

for all $1 \leq \gamma \leq {}_N C_2$, where ${}_N C_2$ is the total number of examples (protein pairs). An SVM model with a linear decision boundary is shown in Figure 3 to distinguish the relevant protein pairs from the irrelevant ones. Note that not all the ${}_N C_2$ possible protein pairs are necessary to be included to train the SVM model (5). For any pair of proteins (x_i, x_j), after we compute its contextual dissimilarity vector $\mathbf{d}_{ij}$, the trained SVM classifier is applied to get the distance of this point to the margin boundary of SVM as $\tilde{y}_{ij} = f(\mathbf{d}_{ij})$. Apparently, $\tilde{y}_{ij}$ is a measure of dissimilarity of the context of this pair of proteins. Thus, it can be used to form a regularization factor as

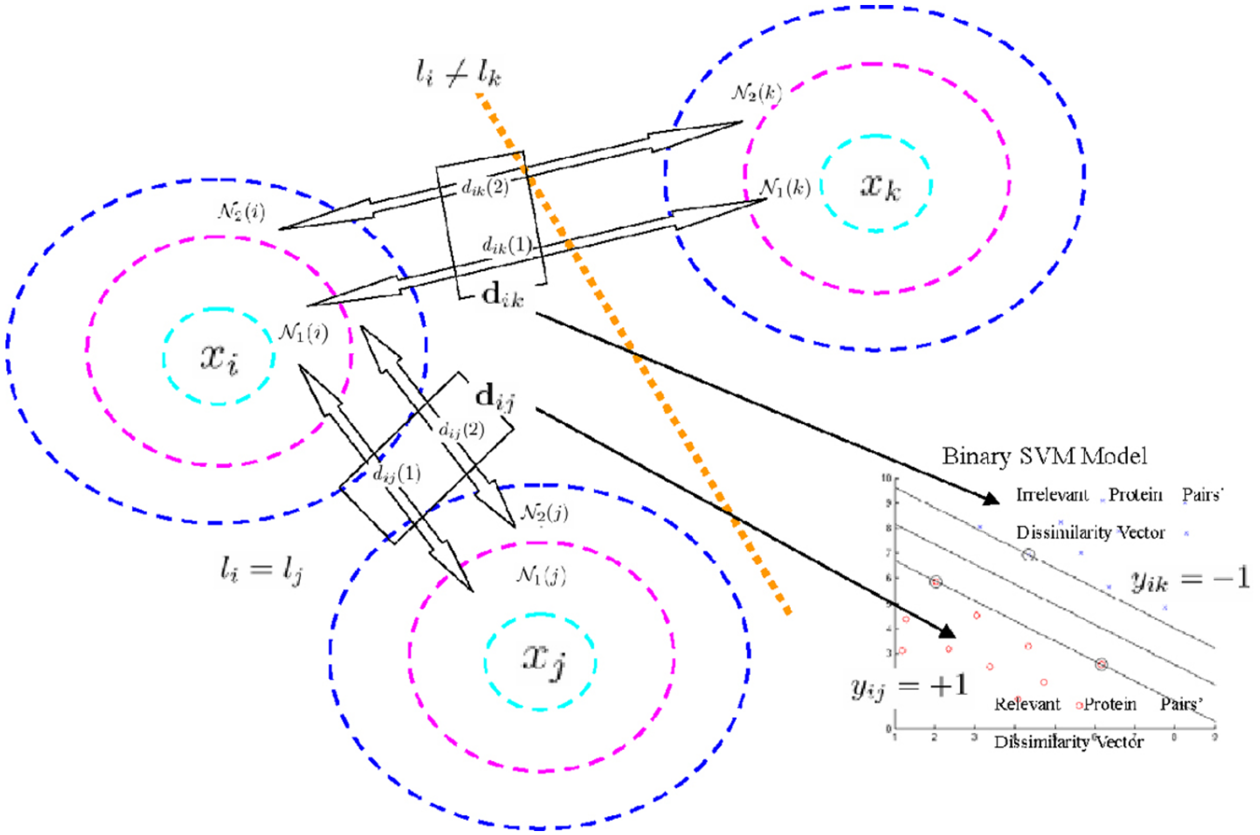

Figure 3 Differentiate relevant and irrelevant proteins by classification. (x_i, x_j) is assumed to be a relevant pair and (x_i, x_k) is assumed to be an irrelevant pair. The contextual dissimilarity vectors of both pairs are distinguished by a binary SVM model.

$$\begin{aligned}\delta'_{ij} &= exp(-\frac{\widetilde{y}_{ij}}{\sigma}) \\ &= exp\left[-\frac{(\mathbf{d}_{ij} \cdot w + b)}{\sigma}\right]\end{aligned} \tag{7}$$

where σ is a preemptor of the factor. With this regularization factor learned from the contextual proteins, we regularize the dissimilarity d_{ij} of protein pair (x_i, x_j) as

$$d^*_{ij} = d_{ij} \times \delta'_{ij} \tag{8}$$

Updating the context and dissimilarity coherently

With the learned dissimilarity measure d^*_{ij}, we can re-define the "context" of a protein x_i according to its dissimilarity to all the other proteins $d^*_{ij}, j \in \{0, \cdots, i-1, i+1, \cdots, N\}$. The new "hierarchical-context" relying on d^*_{ij} is donated as $\mathcal{N}^*_p(i), p = \{1, \cdots, P\}$. In this way, we can develop an iterative algorithm that learns d^*_{ij} and $\mathcal{N}^*_p(i), p = \{1, \cdots, P\}$ coherently. Since $\mathcal{N}^*_p(i)$ implicitly depends on d^*_{ij} through the nearest neighbors of x_i, we use a fixed-point recursion method [47] to solve d^*_{ij}. In each iteration, $\mathcal{N}^*_p(i)$ is first computed by using the previous estimation of d^*_{ij}, which is then updated by multiplying the regularization factor δ'_{ij} as in (8). The iterations are carried out for T times, as given in Algorithm 1.

With the learned dissimilarity matrix $\mathbf{D}^{(t+1)}$, we use $\mathbf{D}^{(t+1)}[0; 1, \ldots, N]$ as the dissimilarity between the query protein x_0 and the database proteins $\{x_1, \ldots, x_N\}$. Thus we can rank the database proteins in an ascending order.

Efficient implementation of ProDis-ContSHC

The proposed learning algorithm is time-consuming. Therefore, it is not suitable for realtime protein retrieval systems. Here we propose several techniques to significantly improve the efficiency of the algorithm.

- Similar to [33], in order to increase the computational efficiency, it is possible to run ProDis-ContSHC for only part of the database of the known proteins. Hence, for each query protein x_0, we first retrieve $N' \ll N$ of the most similar proteins, and perform ProDis-ContSHC to learn the dissimilarity matrix of size $(N' + 1) \times (N' + 1)$ for only those proteins. Then we calculate the new dissimilarity

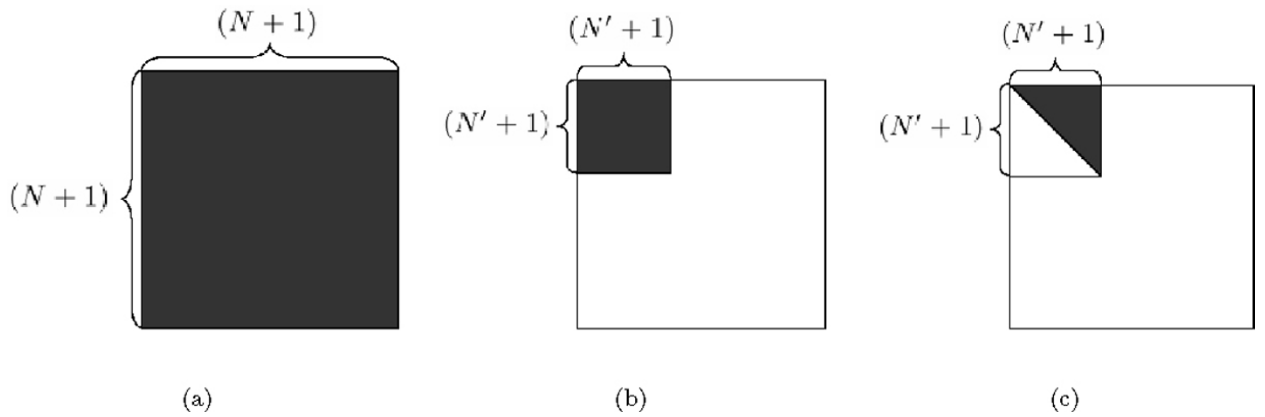

Figure 4 Efficient implementation of ProDis-ContSHC. (a) Performing ProDis-ContSHC on the original matrix of size (N + 1) × (N + 1) from the entire dataset; (b) Performing ProDis-ContSHC on a subset of the database proteins, i.e., a dissimilarity matrix of size (N' + 1) × (N' + 1); (c) Using the symmetry property of the dissimilarity matrix to reduce the training time.

measure $D'_{(N'+1)\times(N'+1)}$ for only those (N' + 1) proteins. Here, we assume that all the relevant proteins will be among the top N' most similar proteins. This strategy is illustrated in Figure 4(a) and 4(b).

- Most of the dissimilarity and similarity measures are symmetric ones, i.e., $d_{ij} = d_{ji}$. As can be observed in (13), the regularization of d_{ij} is also symmetric. Therefore, it is possible to develop an efficient learning algorithm by using this property. In the algorithm, all the computation results of (i, j) (such as $\mathbf{d}_{ij}$ and δ_{ij}) can be used directly by (j, i). In this way, we can save almost half of the computational time, as shown in Figure 4(c).
- A bottleneck of ProDis-ContSHC may be the training procedure for the SVM model in each iteration. For a database of N proteins belonging to C classes, there are ${}_N C_2$ protein pairs, in which $\sum_{c=1}^{C} {}_{N_c}C_2$ are relevant pairs, while $\sum_{c=1}^{C} \sum_{c' \neq c} N_c \times N_{c'}$ are irrelevant pairs, where C is the number C of the protein classes and N_c is the number of proteins in the c-th class $\left(\sum_{c=1}^{C} N_c = N\right)$. There might be a huge number of protein pairs available for the SVM training. However, it is not necessary to include all of them in the training process. One can select a small but equal number of the relevant and the irrelevant pairs to train the SVM classifier. This is an effective way to reduce the training time of SVM.

Algorithm 1 ProDis-ContSHC: **S**upervised Learning of **Pro**tein **Dis**similarity and Updating **H**ierarchical **Con**text **C**oherently.

Require: Input $\mathbf{D} = [d_{ij}]_{(N+1)\times(N+1)}$: matrix of size ($N$+1)×($N$+1) of pairwise protein feature distances, where x_0 is the query protein and $\{x_1, \ldots, x_N\}$ are the database proteins;

Require: Input κ: size of the hierarchical sub-context;

Require: Input P: number of the hierarchical context;

Initialize dissimilarity matrix: $\mathbf{D}^{(1)} = \mathbf{D}$;

for $t = 1, \ldots, T$ **do**

Update the hierarchical context for each protein $x_i : \mathcal{N}_p^{(t)}(i), (p = 1, \cdots, P)$,

$$\begin{aligned}\mathcal{N}_p^{(t)}(i) = \{x_j | x_j \textit{ is among the } k' - th \textit{ to } k'' - th \\ \textit{nearest neighbors of } x_i, \textit{ according to} \\ \mathbf{D}^{(t)}(i; 1, \cdots, N)\}\end{aligned} \tag{9}$$

where $k' = (p - 1) \times \kappa$, $k'' = (p - 1) \times \kappa + \kappa$, and $\mathbf{D}^{(t)}(i; 0, \cdots, N) = [d_{i0}^{(t)}, \cdots, d_{iN}^{(t)}]$.

Compute the contextual proteins dissimilarity vector $\mathbf{d}_{ij}^{(t)}$ for each pair of proteins (i, j), i, j ⌊ $\{0, \ldots, N\}$:

$$\mathbf{d}_{ij}^{(t)} = [d_{ij}^{(t)}(1)\ d_{ij}^{(t)}(2)\ \cdots\ d_{ij}^{(t)}(P)]^\top \tag{10}$$

where $d_{ij}^{(t)}(p) = \frac{1}{k^2} \sum_{m \in \mathcal{N}_p^{(t)}(i), n \in \mathcal{N}_p^{(t)}(j)} d_{mn}^{(t)}$.

Select relevant and irrelevant protein pairs and label them as $y_\gamma = +1$ and $y_\gamma = -1$ respectively, train an SVM model for their contextual dissimilarity vectors $\mathbf{d}_\gamma^{(t)}$ as

$$f^{(t)}(\mathbf{d}) = w^{(t)} \cdot \mathbf{d} + b^{(t)} \tag{11}$$

Compute the distance to the SVM margin boundary for the contextual dissimilarity vector $\mathbf{d}_{ij}^{(t)}$ of each

```
    pair of proteins as $\tilde{y}_{ij}^{(t)} = f^{(t)}(\mathbf{d}_{ij}^{(t)})$, and set a regularization factor for this pair of proteins:
    $$\delta_{ij}^{(t)} = exp(-\frac{\tilde{y}_{ij}^{(t)}}{\sigma}) \quad (12)$$
    Update the pairwise protein dissimilarity measures:
    for $i = 0, 1, \ldots, N$ do
        for $j = 0, 1, \ldots, N$ do
            $$d_{ij}^{(t+1)} = d_{ij}^{(t)} \times \delta_{ij}^{(t)} \quad (13)$$
        end for
    end for
  $\mathbf{D}^{(t+1)} = [d_{ij}^{(t+1)}]_{(N+1)\times(N+1)}$.
  end for
  Output the dissimilarity matrix: $\mathbf{D}^{(t+1)}$.
```

Benchmark sets

To evaluate the proposed ProDis-ContSHC algorithm, we conduct experiments on two different benchmark sets, i.e., the ones used in [21] and [26] respectively.

ASTRAL 1.73 protein domain dataset

Following [26], we use the following database and queries as our first benchmark set:

Database The ASTRAL 1.73 [48] 95% sequence-identity non-redundant data set is used as the protein database. We generate our index database from the tableau data set published by Stivala et al. [49], which contains 15,169 entries.

Queries A query data set containing 200 randomly selected protein domains is used in our experiment. For each query, a list that contains all the proteins in the respective index database is returned with the ranking scores.

We generate a vector of features x for a given protein based on its tableau representation [49].

FSSP/DALI protein dataset

To evaluate the performance of the proposed methods, a portion of the FSSP database [50] is selected as in [21]. This dataset has 3,736 proteins classified into 30 classes. It's constructed according to the DALI algorithm [51,52]. The protein numbers in different classes varies 2 to 561. For protein feature representation, the following two features are extracted from the 3D structure and the sequence of a protein as in [20,21]:

- The Polar-Fourier transform, resulting in the FT_{02} features;
- Krawtchouk moments, resulting in the $Kraw_{00}$ features.

The descriptor vectors are weighted and an integrated descriptor vector is produced as x, which will be used for the protein retrieval tasks.

Results and discussion

Results on ASTRAL 1.73 dataset

To compare a query protein x_0 to a protein x_i in the ASTRAL 1.73 dataset, we compute the cosine similarity [27] as the baseline similarity measure as in [26]. Cosine similarity [27] simply calculates the cosine of the angle between the two vectors x_i and x_j.

$$s_{ij} = C(x_i, x_j) = \frac{x_i \cdot x_j}{||x_i||\ ||x_j||} \quad (14)$$

A higher cosine similarity score implies a smaller angle between the two vectors. Although ProDis-ContSHC is proposed to learn protein-protein dissimilarity d_{ij}, it can be extended easily to learn similarity s_{ij} as well. The only difference is to set the regularization factor as $\delta'_{ij} = exp(\frac{\tilde{y}_{ij}}{\sigma})$ instead of $\delta'_{ij} = exp(-\frac{\tilde{y}_{ij}}{\sigma})$ in (7).

ROC curve and precision-recall curve performance

SCOP [53] fold classification is used as the ground truth to evaluate the performance of the different methods. To fairly compare the accuracy, we use the receiver operating characteristic (ROC) curve [54], the area under this ROC curve (AUC) [54], and the precision-recall curve [55]. Given a query protein x_0 which belongs to the SCOP fold l_0, the top k proteins returned by the search algorithms are considered as the *hits*. The remaining proteins are considered as the *misses*. For the i-th protein x_i belonging to the SCOP fold l_i, if $l_i = l_0$ and $i \leq k$, the protein x_i is defined as a true positive (TP). On the other hand, if $l_i \neq l_0$ and $i \leq k$, x_i is defined as a false positive (FP). If $l_i \neq l_0$ and $i > k$, x_i is defined as a true negative (TN). Otherwise, x_i is a false negative (FN). Using these definitions, we can then compute the true positive rate (TPR or recall), the false positive rate (FPR), recall and precision as follows:

$$\begin{aligned} TPR &= \frac{TP}{P} = \frac{TP}{TP+FN} \\ FPR &= \frac{FP}{N} = \frac{FP}{FP+TN} \end{aligned} \quad (15)$$

$$\begin{aligned} Recall &= \frac{TP}{TP+FN} \\ Precision &= \frac{TP}{TP+FP} \end{aligned} \quad (16)$$

TPR_k, FPR_k, $Recall_k$, and $Precision_k$ are calculated for all $1 \leq k \leq N$, where N is the size of the database. The ROC defines a curve of points with FPR_k as the abscissa and TPR_k as the ordinate. Precision-recall defines a curve with $recall_k$ and $precision_k$ as abscissa and ordinate respectively. We use the area under the ROC curve (AUC) as a single-figure measurement for the quality of a ROC curve [54], and use the averaged AUC over all the queries to evaluate the performance of the method.

To demonstrate the contribution of the supervised learning idea, we also compare ProDis-ContSHC with its unsupervised counterpart, i.e., contextual dissimilarity algorithm based on the unsupervised learning, i.e., ProDis-ContHC. ProDis-ContHC is also applied to improve the cosine similarity. We also compare with the widely-used contextual dissimilarity measure [30,31] (CDM), which tries to take into account the local distribution of the vectors and iteratively estimates distance update terms in the spirit of Sinkhorns scaling algorithm, thereby modifying the neighborhood structures.

The performance of different methods are compared, as shown in Figure 5. Figure 5(a) shows the ROC curves of the original cosine similarity and its improved versions by three contextual similarity learning algorithms on the ASTRAL 1.73 [48] 95% dataset, with different numbers of proteins returned to each query. It can be seen from Figure 5(a) that the TPR of all the methods increases as the FPR grows. The reason is due to the fact that, provided the number of queries is fixed, when the number *k* of returned proteins to each query is very small, the returned proteins are not enough to "represent" the class features of the query, which then causes the low TPR. Meanwhile, in this situation, most of the returned proteins are highly confident of belonging to the same class as the query, resulting in a low FPR. Moreover, the TPR is almost 100% when the FPR>50%. It is clear that the ROC curve of ProDis-ContSHC completely embodies the ROC curves of the other three methods, which implies ProDis-ContSHC is the best method among the four. That also means that supervised learning is better than unsupervised learning for this purpose. ProDis-ContHC, on the other hand, is the second best method among these four, which demonstrates the contribution of the hierarchical sub-context idea to the traditional contextual dissimilarity measures. The overall AUC results are listed in Table 1, from which similar conclusions can be drawn. It is noticeable that the AUC for ProDis-ContSHC is very close to 1, which means ProDis-ContSHC works almost perfectly on this dataset. We further compare these four methods by the precision-recall curves, which are shown in Figure 5(b). It can be seen that the proposed contextual similarity learning algorithms significantly outperform the traditional methods. ProDis-ContSHC, again, is consistently the best method among the four.

Regarding the efficiency of the method, in this experiment, the learning time of the ProDis-ContSHC is longer than that of the ProDis-ContHC and CDM. This is because in each iteration of the learning algorithm, a quadratic programming problem with many training

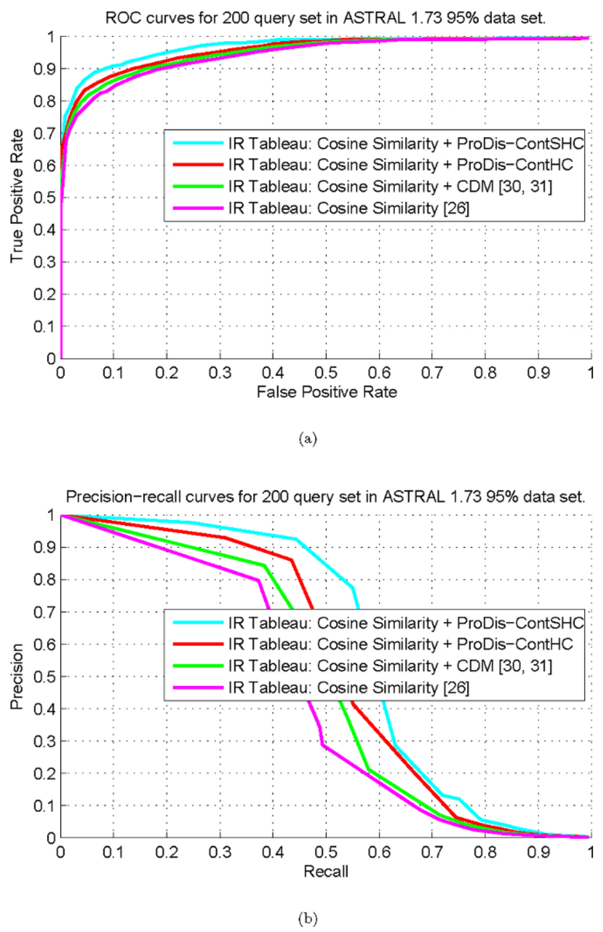

Figure 5 Performance of similarity measures on the ASTRAL 1.73 90% dataset. (a) The ROC curves of the original similarity measure, and the improved measures by ProDis-ContSHC, ProDis-ContHC, and CDM, respectively. (b) The precision-recall curves of the original similarity measure, and the improved measures by ProDis-ContSHC, ProDis-ContHC, and CDM, respectively.

Table 1 Performance of different retrieval methods on the ASTRAL 1

Method	AUC
IR Tableau: Cosine Similarity + ProDis-ContSHC	0.973
IR Tableau: Cosine Similarity + ProDis-ContHC	0.961
IR Tableau: Cosine Similarity + CDM [30,31]	0.954
IR Tableau: Cosine Similarity [26]	0.948
Tableau Search [56]	0.871
QP Tableau [49]	0.925
Yakusa [57]	0.950
SHEBA [58]	0.941
VAST [59,60]	0.890
TOPS [61,62]	0.871

AUC results for QP Tableau [49], SHEBA [58] and VAST [59,60] are taken from [49], which used exactly the same query set and the same dataset as our experiments.

protein pairs have to be solved to train the SVM. In addition, the computation of the regularization factor of supervised similarity learning algorithm needs more function evaluations.

We also compare the proposed algorithms with seven other protein retrieval methods, i.e., tableau search [56], QP tableau [49], Yakusa [57], SHEBA [58], VAST [59,60], and TOPS [61,62]. The overall AUC values are shown in Table 1. It can be concluded that the tableau feature based methods do not always achieve better performance than other methods, such as tableau search. Among the existing tableau feature based methods, IR tableau outperforms the others. Yakusa and SHEBA also have comparable performance. As seen in Table 1, the AUC of the proposed algorithms is clearly better than all the other methods.

Improving different similarity measures via contextual dissimilarity learning algorithms

To further evaluate the robustness of our method, we test the behavior of ProDis-ContSHC and other contextual similarity learning algorithms on different similarity measures. A group of experiments are conducted on the ASTRAL 1.73 95% dataset with the following similarity measures:

- The cosine similarity [27] as introduced in the previous section.
- The Jaccard index [28]: it is defined as the size of the intersection divided by the size of the union of two sets, i.e.,

$$J(x_i,\ x_j) = \frac{|x_i \bigcap x_j|}{|x_i \bigcup x_j|} \tag{17}$$

- The Tanimoto coefficient [29]: it is a generalization of the Jaccard index, defined as

$$J(x_i,\ x_j) = \frac{x_i \cdot x_j}{||x_i||^2 + ||x_j||^2 - x_i \cdot x_j} \tag{18}$$

- Squared Euclidean distance [22]: it is another means of measuring similarity of proteins.

$$d_{ij} = \sqrt{(x_i - x_j)^\top (x_i - x_j)} = \sqrt{\sum_m (x_i(m) - x_j(m))^2} \tag{19}$$

where $x_i(m)$ is the m-th element of vector x_i.

ProDis-ContSHC, ProDis-ContHC, and the CDM algorithms are applied to improve each of these similarity measures, respectively. The AUC values of the corresponding retrieval systems are plotted in Figure 6. In general, improving the original similarity measure by ProDis-ContSHC leads to the largest improvement. The only exception is for Tanimoto coefficient, on which ProDis-ContSHC has slightly lower AUC than ProDis-ContHC, but comparable AUC to the CDM. One possible reason is that the supervised classifier fail to capture the real distribution of the contextual similarity. ProDis-ContHC, on the other hand, also performs better than the CDM algorithm and the original similarity measures. This strongly suggests that our previous conclusions are valid and consistent. That is, hierarchical sub-contextual information can remarkably improve the traditional context-based similarity measures, whereas supervised learning can further improve the accuracy for most of the input similarity measures.

Results on FSSP/DALI dataset

Unlike the similarity measure used in the last experiment, here we use the Euclidean distance [22] to compare a pair of proteins as the baseline dissimilarity measure as in [20,21]. In this way, we have an idea about how our algorithms work with both similarity and dissimilarity measures. For a query protein x_0, the pairwise Euclidean distances, d_{0i}, $i = 1, 2, \dots , N$, are ranked. The top k proteins are returned as the retrieval results. To evaluate the performance of the proposed algorithms, we test them on both the protein retrieval and the protein classification tasks, following [20,21].

Performance on protein retrieval

The efficiency of the proposed dissimilarity learning algorithm is first evaluated in terms of the performance on the protein retrieval task. In this case, each protein $x_i \in X$ of the dataset is used as a query x_0 and the retrieved proteins are ranked according to the shape dissimilarity $d_{0j}$to the query, where $j = 1, 2, \dots , i - 1, i + 1, \dots , N$. We also use the precision-recall curve to demonstrate the performance of the proposed methods, where precision is the proportion of the retrieved proteins that are relevant to the query and recall is the proportion of the relevant proteins in the entire dataset that are retrieved as the results.

To test the robustness and consistency of our methods, we apply our methods to three different protein descriptor vectors, i.e., Daras et al.'s FT_{02}, $Kraw_{00}$, and FT_{02}&$Kraw_{00}$ [20,21] geometric descriptor vectors. We also apply the unsupervised version of our algorithm, ProDis-ContHC, and the CDM algorithm to the same dissimilarity measure and the same descriptor vectors to compare with ProDis-ContSHC. Figure 7 shows the precision-recall curves for different algorithms on different protein descriptor vectors. As mentioned in [20,21], there is always a tradeoff between the precision and recall values. This is clearly shown in Figure 7(a), (b), and 7(c), in which the algorithms reach their peak precision values at the smallest recall values. It can be seen that ProDis-ContSHC has a clearly better performance than any other method, whereas ProDis-ContHC is the second

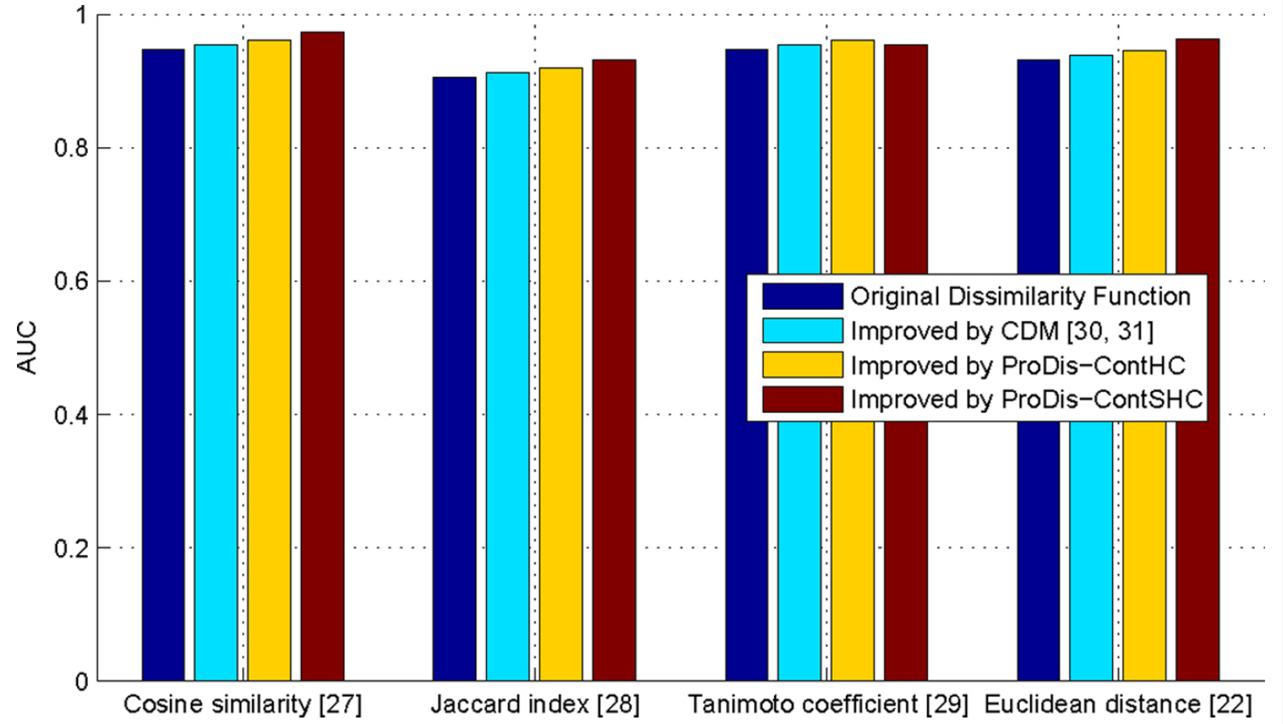

Figure 6 Performance of similarity measures on different base measures on the ASTRAL 1.73 90% dataset. Performance of similarity measures on different base measures on the ASTRAL 1.73 90% dataset. The four base measures being tested are cosine similarity [27], the Jaccard index [28], the Tanimoto coefficient [29], and the Euclidean distance [22].

best one. This is quite consistent with what is observed in the last experiment, in which a similarity measure is used. Therefore, our algorithms can consistently improve any similarity/dissimilarity measure. Among the three protein descriptor vectors, ProDis-ContSHC performs the best on the combined vector, i.e., $Kraw_{00}$ &FT_{02}. This is because this vector not only employs the context, but also their relevant information to predict the relationship between the query and the database proteins.

Performance on protein classification

The performance of the method is also evaluated in terms of the overall classification accuracy [20,21]. To be more specific, for each protein x_i in the database, a dissimilarity measure is applied after removing that protein from the database ("leave-one-out" experiment [63]). A class label l_0 is then assigned to the query x_0 according to the label of the nearest database protein. The overall classification accuracy is given by:

$$\textit{Overall Classification Accuracy} = \frac{\textit{Number of correctly predicted proteins}}{\textit{Total number of proteins in the database}} \qquad (20)$$

We again conduct this experiment with the three descriptor vectors, i.e., FT_{02}, $Kraw_{00}$, and FT_{02}&$Kraw_{00}$. The overall classification accuracy is shown in Table 2. It can be seen that ProDis-ContSHC has a consistently higher than 99% accuracy on all the three descriptor vectors. Each dissimilarity measure achieves its highest accuracy on $Kraw_{00}$ &FT_{02}. Among the four dissimilarity measures, ProDis-ContSHC has the highest accuracy, whereas ProDis-ContHC is the second best one. Therefore, this conclusion has been demonstrated on both similarity and dissimilarity measures on different datasets with different descriptor vectors.

Conclusions

We have introduced in this paper a novel contextual dissimilarity learning algorithm for protein-protein comparison in protein database retrieval tasks. Its strength resides in the use of the hierarchical context between a pair of proteins and their class label information. By extensive experiments, this novel algorithm has been demonstrated to outperform the traditional context-based methods and their unsupervised version.

We formulate the protein dissimilarity learning problem as a context-based classification problem. Under such a formulation, we try to regularize the protein pairwise dissimilarity in a supervised way rather than the traditional unsupervised way. To the best of our knowledge, this is the first study on supervised contextual dissimilarity learning. We propose a novel algorithm, ProDis-ContSHC, which updates a protein's hierarchical sub-context and the dissimilarity measure coherently. The regularization

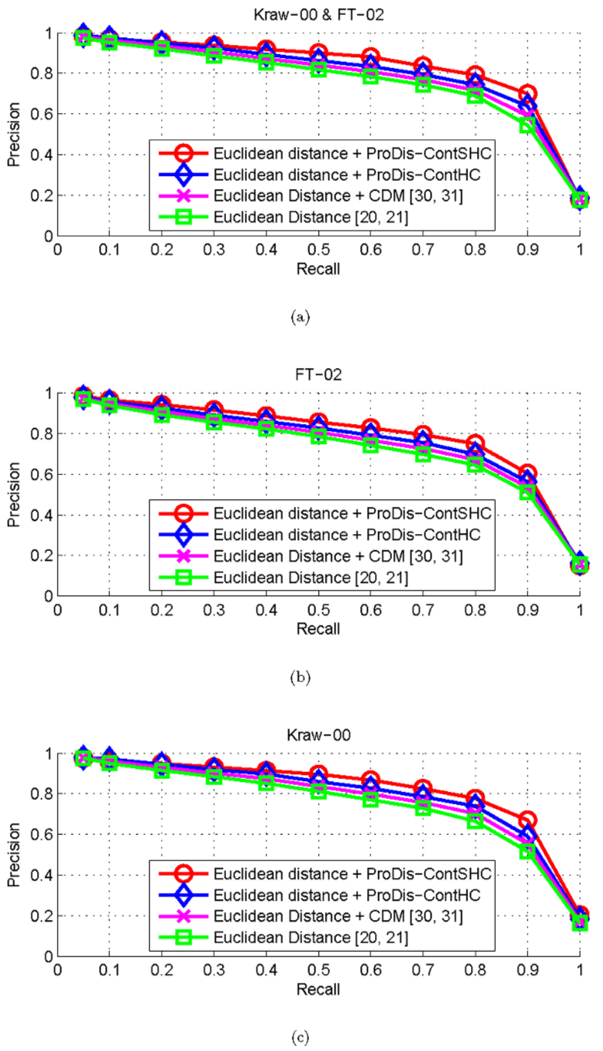

Figure 7 Performance of dissimilarity measures on the FSSP/DALI dataset. (a) The precision-recall curves of the original dissimilarity measure, and the improved measures by ProDis-ContSHC, ProDis-ContHC, and CDM, respectively, with the descriptor vector FT_{02}&$Kraw_{00}$. (b) The precision-recall curves with the descriptor vector FT_{02}. (c) The precision-recall curves with the descriptor vector $Kraw_{00}$.

factors are learned based on the classification of the relevant and the irrelevant protein pairs. The algorithm works in an iterative manner.

Table 2 Overall classification accuracy using different protein descriptors and the Euclidean distance measure

Dissimilarity measure	Descriptors		
	FT_{02}	$Kraw_{00}$	$Kraw_{00}$ &FT_{02}
Euclidean Distance + ProDis-ContSHC	0.9925	0.9954	0.9971
Euclidean Distance + ProDis-ContHC	0.9890	0.9917	0.9928
Euclidean Distance + CDM [30,31]	0.9869	0.9895	0.9909
Euclidean Distance [20,21]	0.9850	0.9879	0.9890

Experimental results demonstrate that supervised methods are almost always better than their unsupervised counterparts on all the databases with all the feature vectors. The proposed method, even though mainly presented for protein database retrieval tasks, can be easily extended to other tasks, such as RNA sequence-structure pattern indexing [64], retrieval of high throughput phenotype data [65], and retrieval of genomic annotation from large genomic position datasets [66]. The approach may also be extended to the database retrieval and pattern classification problems in other domains, such as medical image retrieval [67-69], speech recognition, and texture classification [70].

Acknowledgements

The study was supported by grants from Shanghai Key Laboratory of Intelligent Information Processing, China (Grant No. IIPL-2011-003), Key Laboratory of High Performance Computing and Stochastic Information Processing, Ministry of Education of China (Grant No. HS201107), National Grand Fundamental Research (973) Program of China (Grant No. 2010CB834303 and 2011CB911102), National Natural Science Foundation of China (Grant No. 60973154), Hubei Provincial Science Foundation, China (Grant No. 2010CDA006 and 2010CD06601), and a start-up grant from King Abdullah University of Science and Technology.

This article has been published as part of *BMC Bioinformatics* Volume 13 Supplement 7, 2012: Advanced intelligent computing theories and their applications in bioinformatics. Proceedings of the 2011 International Conference on Intelligent Computing (ICIC 2011). The full contents of the supplement are available online at http://www.biomedcentral.com/bmcbioinformatics/supplements/13/S7.

Author details

[1]King Abdullah University of Science and Technology (KAUST), Mathematical and Computer Sciences and Engineering Division, Thuwal, 23955-6900, Saudi Arabia. [2]Shanghai Institute of Applied Physics, Chinese Academy of Sciences, 2019 Jialuo Road, Jiading District, Shanghai 201800, China. [3]Shanghai Key Laboratory of Intelligent Information Processing, School of Computer Science, Fudan University, Shanghai 200433, China.

Authors' contributions

JW: designed the algorithm, carried out the experiments, analyzed the results, and wrote the manuscript. XG: designed the algorithm and the experiments, improved the manuscript. QW: carried out the experiments, analyzed the results, improved the manuscript. YL: improved the manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Published: 8 May 2012

References

1. Chen SA, Lee TY, Ou YY: **Incorporating significant amino acid pairs to identify O-linked glycosylation sites on transmembrane proteins and non-transmembrane proteins.** *BMC Bioinformatics* 2010, **11**:536.
2. Sobolev B, Filimonov D, Lagunin A, Zakharov A, Koborova O, Kel A, Poroikov V: **Functional classification of proteins based on projection of amino acid sequences: application for prediction of protein kinase substrates.** *BMC Bioinformatics* 2010, **11**:313.
3. Albayrak A, Otu HH, Sezerman UO: **Clustering of protein families into functional subtypes using Relative Complexity Measure with reduced amino acid alphabets.** *BMC Bioinformatics* 2010, **11**:428.
4. Ezkurdia L, Bartoli L, Fariselli P, Casadio R, Valencia A, Tress ML: **Progress and challenges in predicting protein-protein interaction sites.** *Brief Bioinform* 2009, **10(3)**:233-246.

5. Cook T, Sutton R, Buckley K: **Automated flexion crease identification using internal image seams.** *Pattern Recognition* 2010, **43(3)**:630-635.
6. Ofran Y, Rost B: **Protein-protein interaction hotspots carved into sequences.** *PLoS Comput Biol* 2007, **3(7)**:e119.
7. Yhou ZH, Lei YK, Gui J, Huang DS, Zhou X: **Using manifold embedding for assessing and predicting protein interactions from high-throughput experimental data.** *Bioinformatics* 2010, **26(21)**:2744-2751.
8. Xia JF, Zhao XM, Song J, Huang DS: **APIS: accurate prediction of hot spots in protein interfaces by combining protrusion index with solvent accessibility.** *BMC Bioinformatics* 2010, **11**:174.
9. Yhou ZH, Yin Z, Han K, Huang DS, Zhou X: **A semi-supervised learning approach to predict synthetic genetic interactions by combining functional and topological properties of functional gene network.** *BMC Bioinformatics* 2010, **11**:343.
10. Xia JF, Zhao XM, Huang DS: **Predicting protein-protein interactions from protein sequences using meta predictor.** *Amino Acids* 2010, **39(5)**:1595-1599.
11. Shi MG, Xia JF, Li XL, Huang DS: **Predicting protein-protein interactions from sequence using correlation coefficient and high-quality interaction dataset.** *Amino Acids* 2010, **38(3)**:891-899.
12. Huang DS, Zhao XM, Huang GB, Cheung YM: **Classifying protein sequences using hydropathy blocks.** *Pattern Recognition* 2006, **39(12)**:2293-2300.
13. Li JJ, Huang DS, Wang B, Chen P: **Identifying protein-protein interfacial residues in heterocomplexes using residue conservation scores.** *Int J Biol Macromol* 2006, **38**:241-247.
14. Wang B, Chen P, Huang DS, Li JJ, Lok TM, Lyu MR: **Predicting protein interaction sites from residue spatial sequence profile and evolution rate.** *FEBS Lett* 2006, **580(2)**:380-384.
15. Wang J, Li Y, Zhang Y, Tang N, Wang C: **Class conditional distance metric for 3D protein structure classification.** *2011 5th International Conference on Bioinformatics and Biomedical Engineering, (iCBBE).* 2011, 1-4.
16. Chi PH, Scott G, Shyu CR: **A fast protein structure retrieval system using image-based distance matrices and multidimensional index.** *International Journal of Software Engineering and Knowledge Engineering* 2005, **15(3)**:527-545.
17. Marsolo K, Parthasarathy S: **On the use of structure and sequence-based features for protein classification and retrieval.** *Knowledge and Information Systems* 2008, **14**:59-80.
18. Aung Z, Tan K: **Rapid 3D protein structure database searching using information retrieval techniques.** *Bioinformatics* 2004, **20(7)**:1045-1052.
19. Zhang W, Yoshida T, Tang X: **A comparative study of TF*IDF, LSI and multi-words for text classification.** *Expert Syst Appl* 2011, **38(3)**:2758-2765.
20. Daras P, Zarpalas D, Tzovaras D, Strintzis M: **3D shape-based techniques for protein classification.** *IEEE International Conference on Image Processing, 2005. ICIP 2005.* 2005, 1130-1133.
21. Daras P, Zarpalas D, Axenopoulos A, Tzovaras D, Strintzis MG: **Three-dimensional shape-structure comparison method for protein classification.** *IEEE/ACM Trans Comput Biol Bioinform* 2006, **3(3)**:193-207.
22. Oscamou M, McDonald D, Yap VB, Huttley GA, Lladser ME, Knight R: **Comparison of methods for estimating the nucleotide substitution matrix.** *BMC Bioinformatics* 2008, **9**:511.
23. Marsolo K, Parthasarathy S: **On the use of structure and sequence-based features for protein classification and retrieval.** *Proceedings of the Sixth International Conference on Data Mining, 2006. ICDM '06.* 2006, 394-403.
24. Sael L, Li B, La D, Fang Y, Ramani K, Rustamov R, Kihara D: **Fast protein tertiary structure retrieval based on global surface shape similarity.** *Proteins* 2008, **72**:1259-1273.
25. Mittelmann H, Peng J: **Estimating bounds for quadratic assignment problems associated with Hamming and Manhattan distance matrices based on semidefinite programming.** *SIAM J Optim* 2010, **20(6)**:3408-3426.
26. Zhang L, Bailey J, Konagurthu AS, Ramamohanarao K: **A fast indexing approach for protein structure comparison.** *BMC Bioinformatics* 2010, **11(Suppl 1)**:S46.
27. Lee B, Lee D: **Protein comparison at the domain architecture level.** *BMC Bioinformatics* 2009, **10(Suppl 15)**:S5.
28. Rahman M, Hassan MR, Buyya R: **Jaccard index based availability prediction in enterprise grids.** *International Conference on Computer Science, ICCS 2010.* 2010, 2701-2710.
29. Garavaglia S: **Statistical analysis of the Tanimoto coefficient self-organizing map (TCSOM) applied to health behavioral survey data.** *International Joint Conference on Neural Networks, 2001. IJCNN '01.* 2001, 2483-2488.
30. Jegou H, Harzallah H, Schmid C: **A contextual dissimilarity measure for accurate and efficient image search.** *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07.* 2007, 1-8.
31. Jegou H, Schmid C, Harzallah H, Verbeek J: **Accurate image search using the contextual dissimilarity measure.** *IEEE Trans Pattern Anal Mach Intell* 2010, **32(1)**:2-11.
32. Yang X, Bai X, Latecki LJ, Tu Z: **Improving shape retrieval by learning graph transduction.** *10th European Conference on Computer Vision. ECCV 2008.* 2008, 788-801.
33. Bai X, Yang X, Latecki LJ, Liu W, Tu Z: **Learning context-sensitive shape similarity by graph transduction.** *IEEE Trans Pattern Anal Mach Intell* 2010, **32(5)**:861-874.
34. Bai X, Wang B, Wang X, Liu W, Tu Z: **Co-transduction for shape retrieval.** *11th European Conference on Computer Vision. ECCV 2010.* 2010, 328-341.
35. Sinkhorn R: **A relationship between arbitrary positive matrices and doubly stochastic matrices.** *Ann Math Statist* 1964, **35(2)**:876-879.
36. Wang J, Li Y, Bai X, Zhang Y, Wang C, Tang N: **Learning context-sensitive similarity by shortest path propagation.** *Pattern Recognition* 2011, **44(10-11)**:2367-2374.
37. Kuang R, Weston J, Noble W, Leslie C: **Motif-based protein ranking by network propagation.** *Bioinformatics* 2005, **21(19)**:3711-3718.
38. Weston J, Kuang R, Leslie C, Noble WS: **Protein ranking by semi-supervised network propagation.** *BMC Bioinformatics* 2006, **7(Suppl 1)**:S10.
39. Sahbi H, Audibert JY, Rabarisoa J, Keriven R: **Object recognition and retrieval by context dependent similarity kernels.** *International Workshop on Content-Based Multimedia Indexing, 2008. CBMI 2008.* 2008, 216-223.
40. Sahbi H, Audibert J, Keriven R: **Context-dependent kernels for object classification.** *IEEE Trans Pattern Anal Mach Intell* 2011, **33(4)**:699-708.
41. Ding J, Zhou S, Guan J: **MiRenSVM: towards better prediction of microRNA precursors using an ensemble SVM classifier with multi-loop features.** *BMC Bioinformatics* 2010, **11(Suppl 11)**:S11.
42. González AJ, Liao L: **Predicting domain-domain interaction based on domain profiles with feature selection and support vector machines.** *BMC Bioinformatics* 2010, **11**:537.
43. Wang J, Li Y, Liang P, Zhang G, Ao X: **An effective multi-biometrics solution for embedded device.** *IEEE International Conference on Systems, Man and Cybernetics, 2009. SMC 2009.* 2009, 917-922.
44. Wang J, Li Y, Ao X, Wang C, Zhou J: **Multi-modal biometric authentication fusing iris and palmprint based on GMM.** *IEEE/SP 15th Workshop on Statistical Signal Processing, 2009. SSP '09.* 2009, 349-352.
45. Shih-Wen Ke G, Oakes MP, Palomino MA, Xu Y: **Comparison between SVM-Light, a search engine-based approach and the mediamill baselines for assigning concepts to video shot annotations.** *International Workshop on Content-Based Multimedia Indexing, 2008. CBMI 2008.* 2008, 381-387.
46. Ramana J, Gupta D: **LipocalinPred: a SVM-based method for prediction of lipocalins.** *BMC Bioinformatics* 2009, **10**:445.
47. Ey K, Poetzsche C: **Asymptotic behavior of recursions via fixed point theory.** *Journal of Mathematical Analysis and Applications* 2008, **337(2)**:1125-1141.
48. Brenner S, Koehl P, Levitt R: **The ASTRAL compendium for protein structure and sequence analysis.** *Nucleic Acids Res* 2000, **28(1)**:254-256.
49. Stivala A, Wirth A, Stuckey PJ: **Tableau-based protein substructure search using quadratic programming.** *BMC Bioinformatics* 2009, **10**:153.
50. **FSSP/DALI Database.** [http://ekhidna.biocenter.helsinki.fi/dali/start].
51. Holm L, Sander C: **The FSSP database: fold classification based on structure-structure alignment of proteins.** *Nucleic Acids Res* 1996, **24(1)**:206-209.
52. Holm L, Sander C: **The FSSP database of structurally aligned protein fold families.** *Nucleic Acids Res* 1994, **22**:3600-3609.
53. Murzin AG, Brenner SE, Hubbard T, Chothia C: **SCOP: a structural classification of proteins database for the investigation of sequences and structures.** *J Mol Biol* 1995, **247(4)**:536-540.
54. Robin X, Turck N, Hainard A, Tiberti N, Lisacek F, Sanchez JC, Müller M: **pROC: an open-source package for R and S+ to analyze and compare ROC curves.** *BMC Bioinformatics* 2011, **12**:77.
55. Tsai RT, Lai PT: **Dynamic programming re-ranking for PPI interactor and pair extraction in full-text articles.** *BMC Bioinformatics* 2011, **12**:60.

56. Konagurthu AS, Stuckey PJ, Lesk AM: **Structural search and retrieval using a tableau representation of protein folding patterns.** *Bioinformatics* 2008, **24(5)**:645-651.
57. Carpentier M, Brouillet S, Pothier J: **YAKUSA: a fast structural database scanning method.** *Proteins* 2005, **61(1)**:137-151.
58. Jung J, Lee B: **Protein structure alignment using environmental profiles.** *Protein Eng* 2000, **13(8)**:535-543.
59. Madej T, Gibrat JF, Bryant SH: **Threading a database of protein cores.** *Proteins* 1995, **23(3)**:356-369.
60. Gibrat JF, Madej T, Bryant SH: **Surprising similarities in structure comparison.** *Curr Opin Struct Biol* 1996, **6(3)**:377-385.
61. Gilbert D, Westhead D, Nagano N, Thornton J: **Motif-based searching in TOPS protein topology databases.** *Bioinformatics* 1999, **15(4)**:317-326.
62. Torrance G, Gilbert D, Michalopoulos I, Westhead D: **Protein structure topological comparison, discovery and matching service.** *Bioinformatics* 2005, **21(10)**:2537-2538.
63. Zhang W, Sun F, Jiang R: **Integrating multiple protein-protein interaction networks to prioritize disease genes: a Bayesian regression approach.** *BMC Bioinformatics* 2011, **12(Suppl 1)**:S11.
64. Meyer F, Kurtz S, Backofen R, Will S, Beckstette M: **Structator: fast index-based search for RNA sequence-structure patterns.** *BMC Bioinformatics* 2011, **12**:214.
65. Chang WE, Sarver K, Higgs BW, Read TD, Nolan NM, Chapman CE, Bishop-Lilly KA, Sozhamannan S: **PheMaDB: a solution for storage, retrieval, and analysis of high throughput phenotype data.** *BMC Bioinformatics* 2011, **12**:109.
66. Krebs A, Frontini M, Tora L: **GPAT: retrieval of genomic annotation from large genomic position datasets.** *BMC Bioinformatics* 2008, **9**:533.
67. Wang J, Li Y, Zhang Y, Wang C, Xie H, Chen G, Gao X: **Bag-of-features based medical image retrieval via multiple assignment and visual words weighting.** *IEEE Trans Med Imaging* 2011, **30(11)**:1996-2011.
68. Wang J, Li Y, Zhang Y, Xie H, Wang C: **Boosted learning of visual word weighting factors for bag-of-features based medical image retrieval.** *2011 Sixth International Conference on Image and Graphics (ICIG).* 2011, 1035-1040.
69. Wang J, Li Y, Zhang Y, Xie H, Wang C: **Bag-of-features based classification of breast parenchymal tissue in the mammogram via jointly selecting and weighting visual words.** *2011 Sixth International Conference on Image and Graphics (ICIG).* 2011, 622-627.
70. Liu Z, Wang J, Li Y, Zhang Y, Wang C: **Quantized image patches co-occurrence matrix: a new statistical approach for texture classification using image patch exemplars.** *Proceedings of SPIE 8009.* 2011, 80092P.

doi:10.1186/1471-2105-13-S7-S2
Cite this article as: Wang *et al.*: ProDis-ContSHC: learning protein dissimilarity measures and hierarchical context coherently for protein-protein comparison in protein database retrieval. *BMC Bioinformatics* 2012 **13**(Suppl 7):S2.