BIOINFORMATICS ORIGINAL PAPER

Vol. 29 no. 8 2013, pages 1018–1025
doi:10.1093/bioinformatics/btt076

Structural bioinformatics

Advance Access publication February 15, 2013

Prediction of contact matrix for protein–protein interaction

Alvaro J. González, Li Liao* and Cathy H. Wu*

Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716, USA

Associate Editor: Anna Tramontano

ABSTRACT

Motivation: Prediction of protein–protein interaction has become an important part of systems biology in reverse engineering the biological networks for better understanding the molecular biology of the cell. Although significant progress has been made in terms of prediction accuracy, most computational methods only predict whether two proteins interact but not their interacting residues—the information that can be very valuable for understanding the interaction mechanisms and designing modulation of the interaction. In this work, we developed a computational method to predict the interacting residue pairs—contact matrix for interacting protein domains, whose rows and columns correspond to the residues in the two interacting domains respectively and whose values (1 or 0) indicate whether the corresponding residues (do or do not) interact.

Results: Our method is based on supervised learning using support vector machines. For each domain involved in a given domain–domain interaction (DDI), an interaction profile hidden Markov model (ipHMM) is first built for the domain family, and then each residue position for a member domain sequence is represented as a 20-dimension vector of Fisher scores, characterizing how similar it is as compared with the family profile at that position. Each element of the contact matrix for a sequence pair is now represented by a feature vector from concatenating the vectors of the two corresponding residues, and the task is to predict the element value (1 or 0) from the feature vector. A support vector machine is trained for a given DDI, using either a consensus contact matrix or contact matrices for individual sequence pairs, and is tested by leave-one-out cross validation. The performance averaged over a set of 115 DDIs collected from the 3 DID database shows significant improvement (sensitivity up to 85%, and specificity up to 85%), as compared with a multiple sequence alignment-based method (sensitivity 57%, and specificity 78%) previously reported in the literature.

Contact: lliao@cis.udel.edu or wuc@cis.udel.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on October 28, 2012; revised on December 30, 2012; accepted on February 10, 2013

1 INTRODUCTION

Protein–protein interaction (PPI) plays a central role in cellular functions, and the prediction of PPI has become an important part of systems biology in reverse engineering the biological networks for better understanding the molecular biology of the cell. The cost and time of experimental approaches to determining PPI remain high and thus have motivated development of computational methods. Despite significant progress in prediction accuracy, most computational methods only predict whether two proteins interact but do not tell which amino acids on these two proteins actually interact—the very information that can be extremely valuable for further understanding the interaction mechanisms and hence for designing modulation of the interaction via mutagenesis. It is therefore desirable to develop computational methods that can predict the contact matrix for interacting protein domains to indicate which residue pairs interact.

PPI requires some 'compatibility' between the interacting partners in terms of structure, electrostatics and other properties, which shall be conserved during evolution due to the selection pressure and ultimately manifest themselves to certain degrees in the amino acid compositions of the domain sequences that make up the interaction interface (Chothia and Janin, 1975; Jones and Thornton, 1996; Larsen *et al.*, 1998; Livingstone and Barton, 1993). It is such compatibility that most computational methods are trying to establish, by looking at features in protein sequences and structures, to predict the PPI (Tuncbag *et al.*, 2009). The conservation of interface sequences leads to the use of interacting domains as surrogate of PPI, although PPI can not always be reliably predicted just based on interacting domains—because, for instance, in the presence of multi-domains, combination of domains may block interactions that would be otherwise implicated if solely based on individual domain interactions (Moza *et al.*, 2006).

Nevertheless, domain–domain interaction (DDI) provides a useful perspective to studying PPI and is adequate for our purpose here, which is to predict the contact matrix for proteins that are known to interact via certain domains. Specifically, pairs of interacting proteins will be organized in groups of domain–domain interacting (DDI) families. Each member of such a family is an interacting protein pair whose interaction occurs through the association of a pair of domains, an interface that is common to all the members of the family, even though substantial variance might exist between the whole protein complexes to which the protein pairs belong, such as their folded state, cellular function, etc. These DDIs form a relatively large and stable interface of $\sim 2000\text{Å}^2$ on average. Identifying residues from the domain sequences that reside on the interface would provide the training data needed for developing a computational method with predictive power. Several groups have published work in organizing and standardizing known DDIs (Raghavacharil *et al.*, 2008; Finn *et al.*, 2005). The 3DID database (Stein *et al.*, 2010) is among the most successful and widely used ones. This database identifies all cases of DDIs of known three dimensional structure by first assigning Pfam domains (Finn *et al.*, 2010) to each individual protein in the Protein

*To whom correspondence should be addressed.

Data Bank (PDB) (Berman *et al.*, 2000). Next, the atomic contacts between domains in the same structure are computed, where a pair of interacting residues are defined by requiring one or more of hydrogen bonds (N–O distances $\leq$ 3.5Å), salt bridges (N–O distances $\leq$ 5.5Å) or van de Waals interactions (C–C distances $\leq$ 5Å). A DDI is called when at least five of these contacts exist between a pair of domains (Aloy and Russell, 2002). As of March 2011, 3DID collected a total of 5971 distinct DDI families, where each family has on average 26.72 different PDB complexes in which the corresponding DDI was observed.

Figure 1 shows a graphical representation of the contact matrix for a DDI family. In the top left panel, there are four protein complexes, each with a pair of interacting sequence chains. The complexes are not in general homogeneous, but in the interface region, all of them do exhibit a similar shape. This is because, for each one of the protein pairs, the interaction occurs through domain A in the sequences marked red, and through domain B in the sequences marked green. Therefore, since in the domain A region of all the red sequences there is high sequence conservation, they can be aligned in a multiple sequence alignment (MSA^A), and the same happens to the domain B region of the green sequences (MSA^B). The matrix in the bottom right panel is the contact matrix of the DDI family. The black dots in the matrix show those positions in the three dimensional complexes where physical interactions occur between the protein pairs. Note that these residue–residue interactions do not necessarily occur in all the complexes in the family. For instance, i_1 is observed only in complexes 1, 2 and 3; i_2 only happens in complex 1; whereas i_3 happens in all the four complexes.

Such a representation of a DDI had previously been used in Brannetti *et al.* (2000) and Ferraro *et al.* (2006) to predict specificity of SH3 proteins binding to their ligands, where the domain B would correspond to the SH3 domain and the domain A to the binding ligand (peptide). As an intermediate step toward inferring the binding for a new sequence pair, each sequence will be aligned to the corresponding MSA and a consensus contact matrix will be used to determine the contacting residues. Although this method seems to be reasonable and straightforward to implement and had worked well in enhancing the inference of the SH3 domain specificity, its performance as a predictor for protein contact matrix is poor, as shown in the results when applied to the 3DID data, primarily due to its rigid reliance on the multiple sequence alignments. Weigt *et al.* (2009) proposed a method to construct the consensus contact matrix for a large number of homologue pairs without using structural information but only sequence variations, as measured by the so-called direct information, which is computed by maximum entropy with constraints of matching the observed occurrence frequencies of amino acids. Being able to differentiate direct couplings of amino acids from transitive couplings, the method shows remarkable improvement in specificity, as compared with similar method of using the mutual information based on observed occurrence frequencies. Yet, being unsupervised, the method does not address particularly how to predict the contact matrix for a give pair of sequences, other than having them as members in the MSAs and then use the consensus contact matrix as the predicted contact matrix.

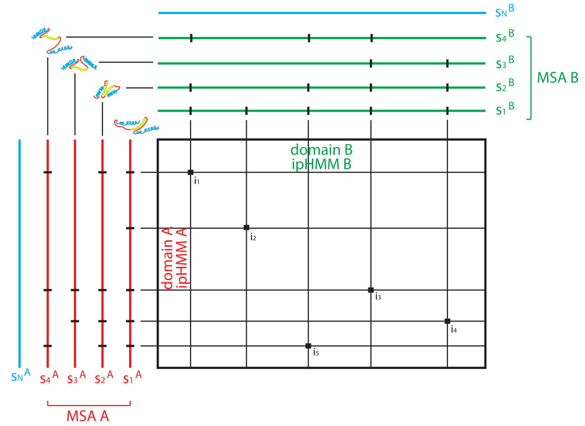

Fig. 1. Schematic diagram of the contact matrix for a pair of interacting domains

In this work, we developed a computational method to predict more accurately the contact matrix for interacting protein domains, by (i) incorporating the structural information of the interface with interaction profile hidden Markov models (ipHMMs) (Friedrich *et al.*, 2006) over MSAs, (ii) tapping into the more subtle features present in the domain sequence with the use of Fisher scores over a simple alignment and (iii) using support vector machines for a supervised learning over the use of a family consensus contact matrix, which is often dependent on the initial multiple sequence alignments and the voting mechanism used to get the consensus. The performance from the leave-one-out cross validation averaged over 115 DDIs for both methods shows significant improvement as compared with a multiple sequence alignment-based method previously reported in the literature.

2 METHOD

The method is based on supervised learning using support vector machines. The training and testing data are composed of 115 DDIs, collected from the 3DID database, whose contact matrices are known from the 3D structure of the protein complexes. As the first step, interacting domains are identified and profiled using ipHMMs, which is discussed in section 2.1. At the second step, each residue for a member domain sequence is represented as a 20-dimension vector of Fisher scores, characterizing how similar it is as compared with the family profile at that position. Each element of the contact matrix for a sequence pair is now represented by a feature vector from concatenating the vectors of the two corresponding residues. The third step is to train a support vector machine for each DDI family, and the last step is to use the trained SVM to predict the element value (1 or 0) from its feature vector for the contact matrix of a new sequence pair.

2.1 Identification and profiling of interacting domains

Since the contact matrix is referring to an interacting domain pair, it is helpful to briefly review how interacting domains are identified and profiled, and their use in predicting DDI and PPI.

A typical scenario is presented in Figure 1, where the contact matrix for two new proteins, s_N^A and s_N^B will be predicted. The first question is what is the likelihood that these two proteins can form an interaction? The DDI-based approach is to identify interacting domains in the two sequences: is there a domain d^A found in s_N^A, and a domain d^B found in s_N^B, such that d^A–d^B constitute a known interacting interface, namely, a

d^A–d^B DDI family that can be found in a database like 3DID? If yes, then s_N^A and s_N^B interact, otherwise they do not interact. This simple scheme works but with false positive and false negative predictions, even assuming the domain identification is reliable. Because of the incompleteness of DDI databases, absence of a known interacting domain cannot rule out a PPI. On the other hand, because of multi-domains, combination of domains may block interactions that are otherwise suggested by individual domains (Moza *et al.*, 2006). Some of these issues and their workarounds have been addressed in our previous work (González and Liao, 2010), and the techniques used will be applied here.

For the above scheme to work, it is essential to identify the presence of d^A in s_N^A and of d^B in s_N^B. Protein domains often contain significant sequence variations. Profile hidden Markov model (pHMM) (Eddy, 1998) is a successful method to capture the commonalities of a given domain while allowing variations. A collection of pHMMs covering many common protein domains and families is available in the Pfam database (Finn *et al.*, 2010). Not only can pHMMs provide a statistical characterization for a family of conserved protein sequences, but they can also be used for efficiently identifying presence of domains in protein sequences through the Viterbi or the posterior decoding algorithms (Rabiner, 1989). However, pHMMs are designed to model domains in general, and the special case of interacting domains must be treated with care because interaction sites impose strong constraints and therefore play a key role in the identification of these domains. To this end, Friedrich *et al.* (2006) developed a new model, called ipHMM, which modifies the ordinary pHMM by adding to the model architecture new states explicitly representing residues on the interface. As shown in Figure 2, the match states of the classical pHMM are now replaced by a non-interacting (M_{ni}) and an interacting match state (M_i). The new match state is provided with the same properties of a match state in the ordinary profile hidden Markov model architecture, i.e. these interacting match states are able to emit all amino acid symbols with probabilities, which are parameters to be fixed according to the training examples. It has been shown that this modification leads to improved accuracy in interacting domain prediction, which makes ipHMMs a better choice for our method.

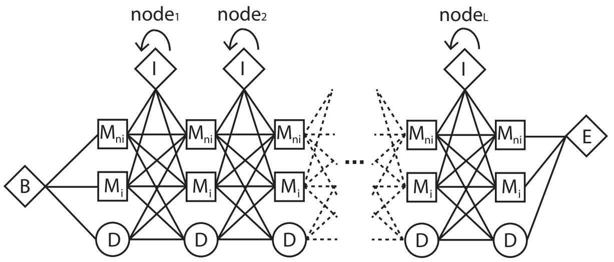

Fig. 2. State diagram of the ipHMM. The usual match states are now split between non-interacting (M_{ni}) and interacting (M_i) match states

Algorithm 1 Average Fisher vectors

```
/* $P$ is the number of sequences in the MSAs at training */
/* $FisherVector()$ is a function that uses Equation (2) to calculate a residue's Fisher vector */
/* $FamilyContactMatrix$ is the contact matrix of Figure 1 where all the interactions in the family (from any sequence pair) are accounted for */
/* $concat()$ is a function that concatenates two vectors to form one new vector */
/* Training */
for each interacting sequence pair $s_m^A-s_m^B$, $1 \le m \le P$ do
    Align $s_m^A$ to $ipHMM^A$, $s_m^B$ to $ipHMM^B$.
    for each amino acid $s_m^A(i)$, $1 \le i \le L^A$ do
        Calculate $FisherVector(s_m^A(i))$ and save it.
    end for
    for each amino acid $s_m^B(j)$, $1 \le j \le L^B$ do
        Calculate $FisherVector(s_m^B(j))$ and save it.
    end for
end for
PO = empty array;/* Positive dataset */
NE = empty array;/* Negative dataset */
for each cell $(i,j)$ in the contact matrix do
    $AvFisher^A \leftarrow average\{FisherVector(s_m^A(i))\}|_{1\le m\le P}$
    $AvFisher^B \leftarrow average\{FisherVector(s_m^B(j))\}|_{1\le m\le P}$
    if $FamilyContactMatrix(i,j)$ is interacting then
        Add $concat(AvFisher^A, AvFisher^B)$ to PO.
    else
        Add $concat(AvFisher^A, AvFisher^B)$ to NE.
    end if
end for
Sample negative dataset so that size(PO) = size(NE).
Train support vector machine (SVM) using PO and NE.
/* Testing */
Align $s_N^A$ to $ipHMM^A$, $s_N^B$ to $ipHMM^B$.
for each cell $(i,j)$ in the contact matrix do
    Calculate $TestFisher^A \leftarrow FisherVector(s_N^A(i))$.
    Calculate $TestFisher^B \leftarrow FisherVector(s_N^B(j))$.
    Evaluate $concat(TestFisher^A, TestFisher^B)$ in the SVM and predict
        +1 (interacting) or − 1 (not interacting).
end for
```

Algorithm 2 All Fisher vectors in the family (no average)

```
/* $P$ is the number of sequences in the MSAs at training */
/* $FisherVector()$ is a function that uses Equation (2) to calculate a residue's Fisher vector */
/* $ContactMatrix^m$ is the contact matrix where only the interactions from sequence pair $m$ are accounted for */
/* $concat()$ is a function that concatenates two vectors to form one new vector */
/* Training */
PO = empty array;/* Positive dataset */
NE = empty array;/* Negative dataset */
for each interacting sequence pair $s_m^A-s_m^B$, $1 \le m \le P$ do
    Align $s_m^A$ to $ipHMM^A$, $s_m^B$ to $ipHMM^B$.
    for each cell $(i,j)$ in the contact matrix do
        Calculate $TrainFisher^A \leftarrow FisherVector(s_m^A(i))$.
        Calculate $TrainFisher^B \leftarrow FisherVector(s_m^B(j))$.
        if $ContactMatrix^m(i,j)$ is interacting then
            Add $concat(TrainFisher^A, TrainFisher^B)$ to PO.
        else
            Add $concat(TrainFisher^A, TrainFisher^B)$ to NE.
        end if
    end for
end for
Sample negative dataset so that size(PO) = size(NE).
Train support vector machine (SVM) using PO and NE.
/* Testing: the same as in Algorithm 1 */
Align $s_N^A$ to $ipHMM^A$, $s_N^B$ to $ipHMM^B$.
for each cell $(i,j)$ in the contact matrix do
    Calculate $TestFisher^A \leftarrow FisherVector(s_N^A(i))$.
    Calculate $TestFisher^B \leftarrow FisherVector(s_N^B(j))$.
    Evaluate $concat(TestFisher^A, TestFisher^B)$ in the SVM and predict
        +1 (interacting) or − 1 (not interacting).
end for
```

With the two interacting domains each being modelled as ipHMMs, the contact matrix can be used to predict potential interacting contacts in a new pair of sequences, s_N^A and s_N^B, that contain d^A and d^B, in the following way: (i) align s_N^A to $ipHMM^A$ and s_N^B to $ipHMM^B$, using Viterbi or posterior decoding; (ii) project the interacting contacts i_1 through i_5 in the contact matrix horizontally onto s_N^A and vertically onto s_N^B; and (iii) predict the residues marked by these projections as interacting contacts in the two new sequences. The rationale for this simplistic approach, which will serve as a baseline in this study, is that the domain–domain interaction characterized by the contact matrix is assumed to behave uniformly across all the complexes in which the interface is found. An important strength of this baseline method—and of the algorithms we will propose in subsection 2.3—is that for aligning the sequences to the domain families in step (i) it is not required a given sequence similarity between the sequence and its corresponding family: even at low levels of sequence homology, the alignment can be calculated via Viterbi or posterior decoding, and this is all that is needed to construct feature vectors and make contact predictions. We will show that even at low similarity levels, our predictions can be highly accurate (Fig. 6).

Brannetti *et al.* (2000) and later Ferraro *et al.* (2006) have previously used this alignment-based approach to study SH3 domain specificity. While their goal is to predict whether a protein containing an SH3 domain interacts with a ligand, their method aligns the query protein and ligand, respectively, with the training proteins (in an MSA) and ligands (in another MSA) whose contact matrices are known. From the alignment, the putative contact points for the query protein–ligand pair are identified, and the amino acid pairs at these contact points are compared with the distribution of amino acid pairs at these contact positions for the training protein–ligand pairs to infer how likely the query protein and ligand interact. It is worth noting that our baseline approach is already a slight refinement to the method of Ferraro by using ipHMMs, which take into account the structural information about interacting residues at the model construction stage. The ipHMMs allow for more accurate alignment of a query sequence to the existing sequences in the model by using the Viterbi algorithm than a progressive approach does, as the one used in multiple sequence alignments. Still, the assumption that all the protein complexes in a given DDI family share the same contact matrix does not hold in general because of sequence and structural variations, even within the same family. Consequently, the prediction would suffer; a false negative may be caused from missing the interacting sites that are new to s_N^A and s_N^B and are not recorded in the contact matrix, and a false positive may occur when the interacting sites annotated in the contact matrix are not conserved in s_N^A and s_N^B.

2.2 Representing residues with Fisher scores

To overcome the shortcomings of the alignment-based approach that we discussed above, we will have to allow a pair of residues to be considered as a contact point even when they are not aligned to any contact point in the known contact matrix in the DDI family (false negative) and also allow the pair to be considered as a non-contact point when they are aligned to a known contact point in the DDI family (false positive). To achieve these goals, we propose to (i) characterize each residue in the sequence in a way that captures how it contributes to the alignment of the sequence with the whole family as an ipHMM, and (ii) determine for each residue pair its candidacy as a contact point, not based on a rigid alignment to the existing contact matrix, but by comparing it with the known contacting pairs and non-contacting pairs, all expressed in a representation from step (i), in a supervised learning approach. We address the first part in this subsection and the supervised learning part in the next subsection.

The representation that we adopted for characterizing the residues is the Fisher scores derived from aligning the sequence to the ipHMM. The use of Fisher vectors to represent protein sequences was first proposed by Jaakkola *et al.* (1999) in the context of detection of remote protein homologues, and was later adopted for other applications in bioinformatics, including discriminating signal peptides from transmembrane proteins (Kahsay *et al.*, 2005) and PPI prediction (González and Liao, 2009; González and Liao, 2010). Given an ipHMM for the family of domain d^A, the probability of s_1^A in Figure 1 being a member, i.e. containing domain d^A, can be calculated as $P(s_1^A|\theta^A)$, where θ^A are the parameters (emission and transition probabilities) of $ipHMM^A$. The Fisher score of the alignment with respect to parameter θ_0^A is defined as

$$\frac{\partial}{\partial \theta_0^A}\log P\left(s_1^A|\theta^A\right) \qquad (1)$$

In this work, we use a vector of Fisher scores to represent not the protein sequences in the multiple sequence alignments, but rather each one of their residues. Take the sequence s_1^A and $ipHMM^A$ in Figure 2, and let domain d^A be of length L^A, i.e. $ipHMM^A$ has L^A match states (see Fig. 2). The alignment of s_1^A to $ipHMM^A$ will also be of length L^A, although some amino acids in the original sequence when its length is longer than L^A will not appear in the alignment, as they would align to insert states. Let us thus denote this alignment as the amino acid chain $\left\{s_1^A(1), s_1^A(2), \ldots, s_1^A(L^A)\right\}$. Each residue in the sequence can be characterized by a set of Fisher scores defined as follows:

$$\left\langle \frac{\partial}{\partial e_{M_i}^{AA_1}}\log P\left(s_1^A|\theta^A\right), \frac{\partial}{\partial e_{M_i}^{AA_2}}\log P\left(s_1^A|\theta^A\right), \ldots, \frac{\partial}{\partial e_{M_i}^{AA_{20}}}\log P\left(s_1^A|\theta^A\right)\right\rangle \qquad (2)$$

where $e_{M_i}^{AA_k}$, $1 \le k \le 20$ is the emission probability of amino acid k at match state M_i. Note that these match states can be either interacting or non-interacting (see Fig. 2); both types will be used in this study and a comparison of the influence on the prediction performance will be made.

The same can be done for all the residues in the alignment of s_1^B to $ipHMM^B$. The element (i,j) in the contact matrix for the sequence pair can now be represented by the concatenation of the Fisher vector for $s_1^A(i)$ and the Fisher vector for $s_1^B(j)$. The value of element (i,j), as the ground truth, is used to label the concatenated vector with $+1$ for interacting or -1 for non-interacting correspondingly. In the next section, we propose to use a support vector machine to learn and predict the interacting condition between pairs of residues coming from two unseen protein sequences, for instance s_N^A and s_N^B in Figure 1.

2.3 Supervised learning with SVMs

We propose two algorithms to conduct the supervised learning with support vector machines. These two algorithms differ in how the training examples are prepared.

The first algorithm is based on the consensus contact matrix for the sequence pairs in the training set. The idea is illustrated in Figure 3, and the pseudocode is in panel Algorithm 1. The following is done for all the sequence pairs in the training data, for which we know their residue contacts information from the corresponding three dimensional

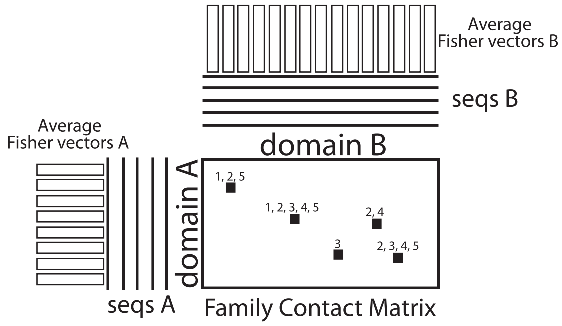

Fig. 3. Schematic diagram of Algorithm 1 (Average Fisher vectors)

structures: the two sequences are aligned to their ipHMMs, and their residues in the alignment are represented as Fisher vectors. Once this is done for all the sequence pairs in the two MSAs, each element (*i*,*j*) in the ($L^A \times L^B$)-sized contact matrix will be represented by the concatenation of two Fisher vectors: one is the average of all the Fisher vectors from the training sequences in MSA^A at position *i*, and the other is the average of all the Fisher vectors from the training sequences in MSA^B at position *j*. The vector for this element will be labeled +1 if any of the training sequence pairs presents an interaction at this position; otherwise it will be labeled −1. In this way, we construct a dataset with ($L^A \times L^B$) example vectors, where the features of each vector encapsulate characteristics of the family through the averages of Fisher vectors across all the training sequence pairs. These vectors belong to one of two categories: positive examples or negative examples, according to contacts in the matrix. The dataset is then learned with a SVM, and evaluated on a test set of ($L^A \times L^B$) examples coming from two protein sequences whose three dimensional structure is also known but preserved for testing. These test examples are constructed in the same way that train examples are built, except that they are not averaged across the family. Notice that in training, all the positive examples are used, but not all the negatives are included in the training set: we randomly choose negative examples so that the positive and the negative sets are the same size. This is done to keep the convergence time of the SVM manageable and to avoid a minority class problem. Furthermore, our experiments showed that the learning ability of the SVM is robust towards this decrease in the number of negative train examples. The same is done in testing.

The second algorithm is illustrated in Figure 4, and its pseudocode is in panel Algorithm 2. For a pair of interacting sequences in training set, they are aligned to the ipHMMs and each of their residues is represented as Fisher vectors. The contact matrix for this specific pair of sequences, which may be different from the contact matrix of the other sequence pairs in the training set, is used to label the concatenated Fisher vectors at each position as +1 or −1. The same is done for all the training sequence pairs. This means that the positive and negative datasets are prepared for each sequence pair in the training set individually, not relying on a consensus contact matrix and fisher vectors averaged across the DDI family. This scheme of using all fisher vectors, of course, results in a significantly increased number of training examples—the number is roughly *P* times more than that in the first algorithm, where *P* is the number of sequence pairs in the training set. It is expected that by taking the residue pairs out of the constraints of a rigid alignment, the method can better handle the sequence and structural variations in the interacting domains and contact points. The sampling of negative examples done in Algorithm 1 is also done here.

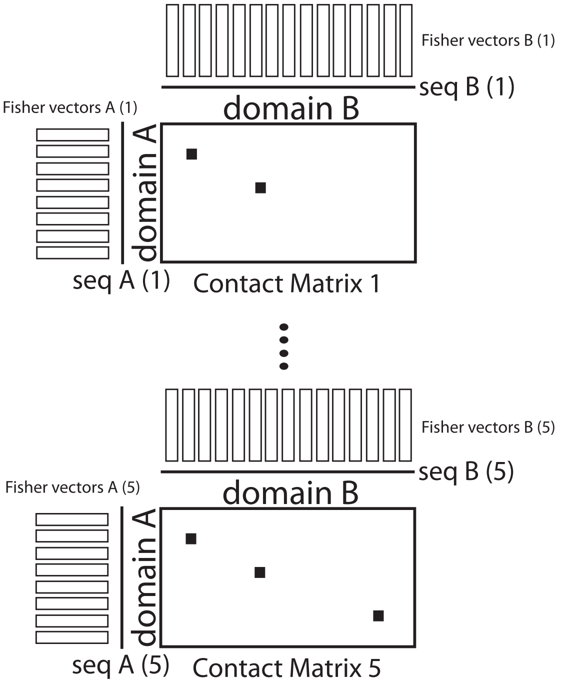

Fig. 4. Schematic diagram of Algorithm 2 (All Fisher vectors)

3 RESULTS

The method was trained and tested on a dataset collected from the 3DID database with close to 6000 DDI families. Of those, we have chosen a subset of 115 DDIs where for each family, (i) the two interacting domains are no longer than 550 amino acids, and (ii) the number of different PDB structures in the family is between 3 and 300. In this way, we are testing our methods on a rich dataset, while avoiding big families that could take prohibitive times to train and cross-validate.

For each DDI family, we assess the predictive power of the two proposed algorithms using leave-one-out cross validation. If there are *P* protein pairs in a family (i.e. *P* PDB complexes for the family), we set aside one of the pairs for testing, and use the remaining ($P - 1$) examples for training the models. This is repeated such that each protein pair in the family is used once as the test example. It is common to find DDI families in 3DID contain highly similar sequence pairs. To avoid trivial test cases (when the test pair has an almost identical counterpart in the train set), we represent each sequence pair by its two concatenated sequences, and group these artificial sequences in clusters, within which any two sequences have a ≥90% BLAST sequence similarity. Then in training and testing, we use only one sequence pair from each cluster. In this way, we tested a total of 473 protein pairs. As a baseline, we present prediction results when only the alignment of the two sequences to the ipHMMs and the family contact matrix are used to predict contacts in the test sequence pair. In this case, to assign a real number to each predicted residue pair, we use the number of contacts that were observed at that point in the family contact matrix; then we will predict interaction when this number is greater than zero, and when it is zero we predict no interaction. When algorithms 1 and 2 are used, each element in the contact matrix of the test pair is assigned a prediction score in the form of a real number, which is the signed distance from the vector representation of the element to the hyperplane of the trained SVM. We predict an element to be interacting when the distance is greater than zero, and no interaction otherwise. For all three methods (baseline, ‘Average’ and ‘All vectors’), we measure the sensitivity and specificity of the predictions as defined as follows:

$$sensitivity = TP/(TP + FN)$$
$$specificity = TP/(TP + FP)$$

where TP: True Positives, FN: False Negatives, FP: False Positives. Since we have real-valued predictions for each point in the test contact matrix, we also measured the performance by the area under the ROC curve (AUC), which does not rely on a preset threshold for making predictions and therefore can more reliably assess a method’s power to differentiate positive

Table 1. Comparison between proposed methods and baseline, according to sensitivity, specificity and AUC

Method	Sensitivity (%)	Specificity (%)	AUC (%)
Average (Algorithm 1)	M_{ni}: 84.46 M_i: 71.75	M_{ni}: 69.54 M_i: 84.81	M_{ni}: 81.01 M_i: 89.35
All vectors (Algorithm 2)	M_{ni}: 84.10 M_i: 59.90	M_{ni}: 66.53 M_i: 82.62	M_{ni}: 83.33 M_i: 91.20
Baseline	56.92	78.22	78.16

examples from negative examples. Table 1 lists the results averaged over the 115 tested DDI families, and Figure 5 shows the histograms of these averages: in these curves, for any given value of the figure of merit in the horizontal axis (sensitivity, specificity or AUC), the vertical axis shows the number of families that attained a performance greater than this value. The table shows a generalized performance analysis over all the families, whereas the histograms allow for a more per-family based analysis. This is an important analysis because the models we propose are built independently for each family.

In Equation (2) we had shown how the Fisher scores can be calculated with respect to non-interacting (M_{ni}) or interacting (M_i) match states. In Table 1 and Figure 5, we show results for each of these cases. In terms of AUC, which measures the learning ability of the algorithm, the best results are obtained, expectedly, with the M_i vectors, because these encapsulate information more directly relevant to the interaction. This is true for both algorithms ('Average' and 'All vectors'). Also, according to the AUC, the 'All vectors' algorithm achieves more learning power than the 'Average' algorithm, and the two of them perform considerably better than the baseline. These conclusions can be drawn from both, Table 1 (AUC column) and Figure 5 (bottom). Even though more computationally expensive, the 'All vectors' approach, by treating equally each residue pair in the family, independent of what sequence pair the two residues come from, can better learn the correlation between the evolutionary and the structural features of the residue pair and its likelihood of being a contact. To assess the performance in terms of sensitivity and specificity, a thresholding scheme needs to be adopted. Our thresholding for the baseline method is to predict as positive points in the matrix where training sequences have contacts. For the supervised learning algorithms, we predict as positive the residue pairs with a positive distance to the hyperplane. Even though this thresholding scheme is arbitrary, the results show that, for both algorithms, the non-interacting match states have higher sensitivity, whereas the interacting match states have higher specificity.

It is well known that sequence similarity is a factor that can influence the prediction accuracy in a significant way for many methods that use sequence information. To investigate how our method is affected, we divided the 115 families into four groups based on sequence similarity: 20–40%, 40–60%, 60–80% and 80–90%. Here, we treat all protein pairs in a family as concatenated pseudo-sequences, and calculate the sequence similarity between all pairs of pseudo-sequences. The average family similarity is the mean of all the pairwise comparisons. The box plots in Figure 6 show how the prediction performance, measured as AUC, varies across these four groups. It can be seen that, while the average prediction accuracy in these groups

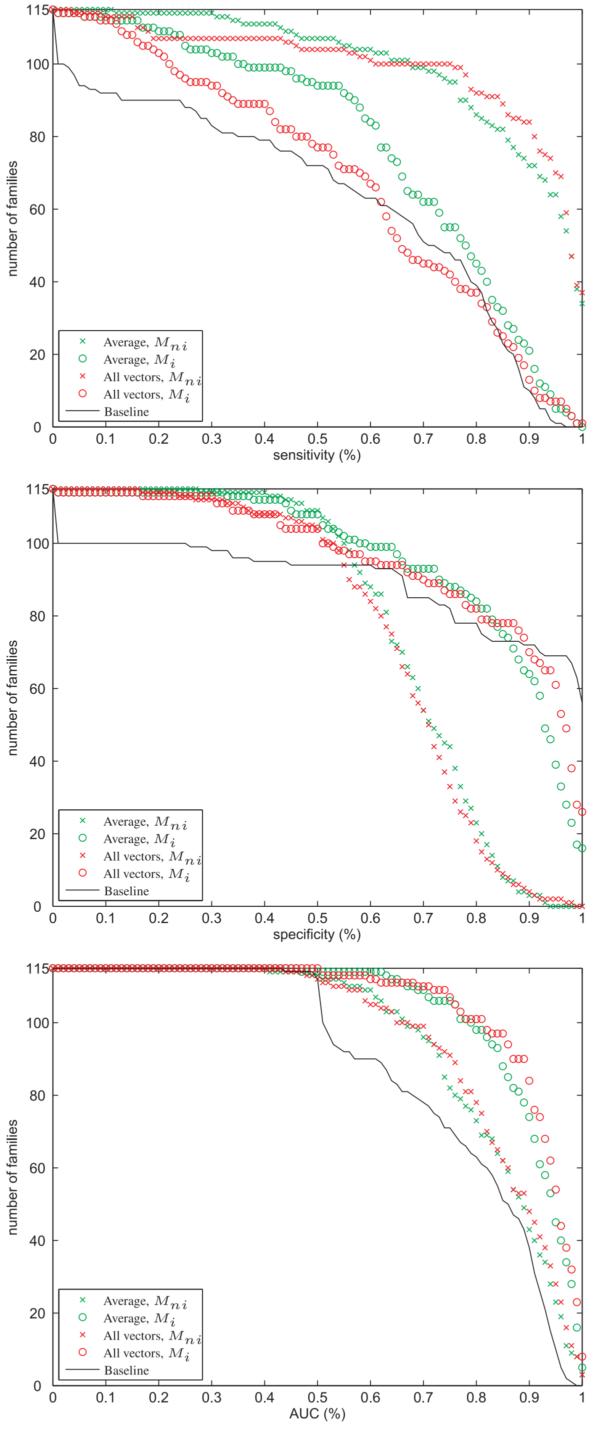

Fig. 5. Results in terms of sensitivity (top), specificity (middle) and AUC (bottom). In each histogram, a point in the curve shows in the vertical axis the number of DDI families that have a performance greater or equal than the position in the horizontal axis

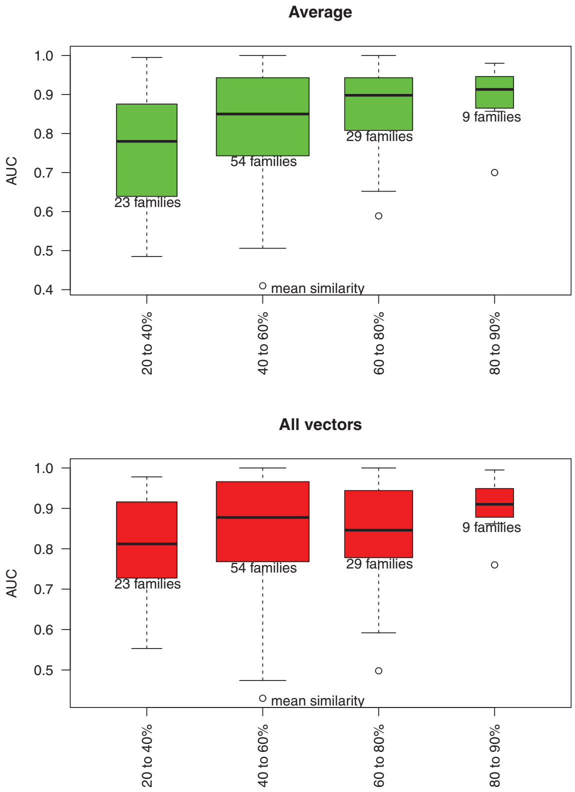

Fig. 6. The 115 DDI families being studied have been categorized into four groups according to their average within family similarity. That is, we treat all protein pairs in a family as concatenated pseudo-sequences, and calculate the sequence similarity between all pairs of pseudo-sequences. The average family similarity is the mean of all the pairwise comparisons. The box plots show a general trend that the performance is improved, as expected, as the similarity increases. It is worth noting that the methods achieve high accuracy even at low sequence similarity, which may be attributed to the incorporation of structural information

indeed increase as the sequence similarity increases, the method achieves high accuracy (median AUC > 0.8) even when sequence similarity is very low, in the so-called twilight zone. This is likely due to our method's incorporation of structural information, such as contact points, which tend to be more conserved during evolution than sequence similarity for remote homologues.

It is worth noting that in this work, we assume the interacting domains are already identified and our method addresses the question: where are the contact points/residues for the pair of interacting domains? Domain identification is a challenging task by itself and the performance varies depending on the methods being used and input data available. The ipHMM, on which our method is based, was originally developed to identify interacting domains, and it has reported overall 0.64 and 0.70 sensitivity and specificity respectively, and as high as 0.81 sensitivity and 0.97 specificity for Ext. Ser/Thr-type protein kinase (Friedrich *et al.*, 2006). As future work, it is possible that we integrate our method (as a second step) with interacting domain identification (as first step), namely, by feeding back the prediction of contact points to verify the domain identification, and then using the improved domain identification to enhance contact points prediction, iteratively.

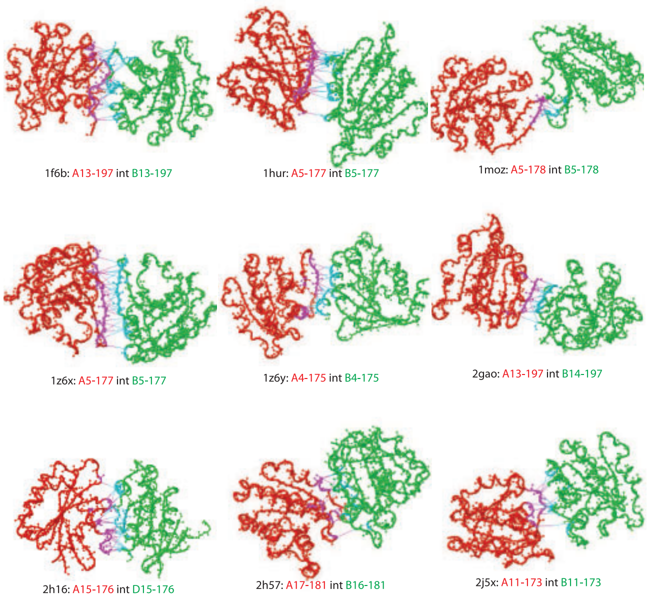
1f6b: A13-197 int B13-197
1hur: A5-177 int B5-177
1moz: A5-178 int B5-178
1z6x: A5-177 int B5-177
1z6y: A4-175 int B4-175
2gao: A13-197 int B14-197
2h16: A15-176 int D15-176
2h57: A17-181 int B16-181
2j5x: A11-173 int B11-173

Fig. 7. Nine different protein complexes from the 'ARF interacting with ARF' DDI family. In each complex, we show a pair of interacting proteins. Lines connect interacting residues in the interface. The domain d^A in each PPI is coloured red, and the domain d^B is green. Interacting residues in d^A are coloured magenta, and interacting residues in d^B are coloured cyan. Each PPI is labelled with the PDB code of the complex, the chain name of the sequence containing d^A and the one containing d^B, and also the coordinates of the two domains within the corresponding sequences

We present a case study that illustrates the benefits of using the Fisher vectors followed by supervised learning to produce a robust model for the contact matrix. Let us analyse the DDI family that involves ARF–ARF domain interaction. The ARF (ADP Ribosylation Factor) domain family (Pfam accession number PF00025) constitutes a family of GTP-binding proteins, all belonging to the Ras superfamily. They primarily function as regulators of vesicular traffic and actin remodelling, and are found interacting with membrane proteins (Pasqualato *et al.*, 2002). Figure 7 shows the molecular structures of nine protein pairs that interact through the 'ARF interacting with ARF' DDI interface. Although the nine structures share high structural similarity, the way the interaction occurs in each example exhibit significant variations. For instance, PDB 1f6b shows a complex interaction with as many as 39 interacting residue pairs. This is reflected in the two dimensional contact matrix for this pair (Supplementary Fig. S1, first row, first column), where the 39 contact points are represented as black dots. A similar (but slightly lower) complexity can be seen in PDB 1hur (first row, second column), with 25 contacts, and in PDB 1z6x (second row, first column), with 26 contacts. PDB 2h16 (third row, first column) is an interesting example with a high number of contacts (40) spread across the interface. At the other side of the spectrum, we have PDB 1moz (first row, third column) and PDB 2gao (second row, third column), with a looser interaction (notice how the two proteins appear more separated from each other), which is reflected in a lower number of contact residue pairs (9 and 16, respectively). For illustration, let us predict the contact points of PDB 1moz using remaining eight structures for training. By using the baseline method, the family contact matrix would include all the contacts observed in the training structures. We show this consensus matrix with black dots in Supplementary Figure S2 in which the test contact matrix (PDB 1moz) is shown with red dots. In this case, the consensus contact matrix would fail to predict each of the real contacts (no true positives, sensitivity 0%, specificity 0%). However, when the training is deferred to the Fisher vectors on interacting match states (M_i) and the 'All vectors' algorithm is used, the 34 596 residue pairs in the test contact matrix can be evaluated against the learned SVM. Inspecting the sorted list of distances to the hyperplane from higher to lower, we would find the first two real contacts in positions 24 and 37. In the first 500 predictions, six contacts will be found, and the ninth (last) contact is found at position 2607 in the list. This ordering produces an AUC of 98.10%, whereas the baseline produces an AUC of 44.40%.

4 CONCLUSION

We have shown in this work that the contact matrix for a pair of interacting proteins can be predicted with high accuracy by combining sequence, structural and evolutionary information in an integrative manner, as captured in the ipHMMs for the two protein domain families of these interacting proteins. By devising Fisher score vectors to represent amino acids at the interacting domains, our method is capable of extracting characteristic features without constraining the residues in the rigid multiple sequence alignments used in the previous methods. This enables our method to handle residues corresponding to delete and insert states, and allows for a supervised learning on individual contact points, eliminating the need of a consensus contact matrix for the domain families, which has been a main source for false predictions. While designed for predicting contact points between interacting protein domains, the method may be useful as a module in protein folding and docking, in which a recent study (Marks *et al*., 2011) has shown that the co-evolution couplings, distinguished from the noise set of observed correlation using maximum entropy, provide an excellent indicator for residue–residue proximity in folded structures. As future work, we will investigate how to adapt the maximum entropy model into the framework of ipHMM-Fisher-SVM for further improvement in accuracy.

Funding: Delaware INBRE program, with grants from the National Center for Research Resources—NCRR (5P20RR016472-12) and the National Institute of General Medical Sciences—NIGMS (8 P20 GM103446-12) from the National Institutes of Health.

Conflict of Interest: none declared.

REFERENCES

Aloy,P. and Russell,R. (2002) Interrogating protein interaction networks through structural biology. *Proc. Natl Acad. Sci. U S A*, **99**, 5896–5901.

Berman,H. *et al*. (2000) The protein data bank. *Nucleic Acids Res*., **28**, 235–242.

Brannetti,B. *et al*. (2000) SH3-SPOT: an algorithm to predict preferred ligands to different members of the SH3 gene family. *J. Mol. Biol*., **298**, 313–328.

Chothia,C. and Janin,J. (1975) Principles of protein-protein recognition. *Nature*, **256**, 705–708.

Eddy,S. (1998) Profile hidden Markov models. *Bioinformatics*, **14**, 755–763.

Ferraro,E. *et al*. (2006) A novel structure-based encoding for machine-learning applied to the inference of SH3 domain specificity. *Bioinformatics*, **22**, 2333–2339.

Finn,R. *et al*. (2005) iPfam: visualization of protein-protein interactions in PDB at domain and amino acid resolutions. *Bioinformatics*, **21**, 410–412.

Finn,R. *et al*. (2010) The Pfam protein families database. *Nucleic Acids Res. (Database Issue)*, **38**, D211–D222.

Friedrich,T. *et al*. (2006) Modeling interaction sites in protein domains with interaction profile hidden Markov models. *Bioinformatics*, **22**, 2851–2857.

González,A. and Liao,L. (2009) Constrained Fisher scores derived from interaction profile hidden Markov models improve protein to protein interaction prediction. In: *Proceedings of the First International Conference on Bioinformatics and Computational Biology (BICoB)*. International Society for Computers and their Applications (ISCA), New Orleans, LA, pp. 236–247.

González,A. and Liao,L. (2010) Predicting domain-domain interaction based on domain profiles with feature selection and support vector machines. *BMC Bioinformatics*, **11**, 537.

Jaakkola,T. *et al*. (1999) A discriminative framework for detecting remote protein homologies. *J. Computat. Biol*, **7**, 95–114.

Jones,S. and Thornton,J. (1996) Principles of protein-protein interactions. *Proc. Natl Acad. Sci. U S A*, **93**, 13–20.

Kahsay,R. *et al*. (2005) Discriminating transmembrane proteins from signal peptides using SVM-Fisher approach. In: *The Proceedings of the Fourth International Conference on Machine Learning and Applications (ICMLA'05)*. Los Angeles, CA, pp. 151–155.

Larsen,T. *et al*. (1998) Morphology of protein-protein interfaces. *Structure*, **6**, 421–427.

Livingstone,C. and Barton,G. (1993) Protein sequence alignments: a strategy for the hierarchical analysis of residue conservation. *Comput. Appl. Biosci*., **9**, 745–756.

Marks,D. *et al*. (2011) Protein 3d structure computed from evolutionary sequence variation. *PLoS ONE*, **6**, e28766.

Moza,B. *et al*. (2006) Long-range cooperative binding effects in a T cell receptor variable domain. *Proc. Natl Acad. Sci. U S A*, **103**, 9867–9872.

Pasqualato,S. *et al*. (2002) Arf, arl, arp and sar proteins: a family of gtp-binding proteins with a structural device for 'front-back' communication. *EMBO Rep*., **3**, 1035–1041.

Rabiner,L. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, **77**, 257–286.

Raghavacharil,B. *et al*. (2008) DOMINE: a database of protein domain interactions. *Nucleic Acids Res*, **36**, D656–D661.

Stein,A. *et al*. (2010) 3DID: identification and classification of domain-based interactions of known three-dimensional structure. *Nucleic Acids Res*, **39**, D718–D723.

Tuncbag,N. *et al*. (2009) A survey of available tools and web servers for analysis of protein-protein interactions and interfaces. *Brief. Bioinform*., **10**, 217–232.

Weigt,M. *et al*. (2009) Identification of direct residue contacts in protein-protein interaction by message passing. *PNAS*, **106**, 67–72.