BIOINFORMATICS ORIGINAL PAPER Vol. 26 no. 15 2010, pages 1841–1848
doi:10.1093/bioinformatics/btq302

*Sequence analysis*

Advance Access publication June 6, 2010

# Applying the Naïve Bayes classifier with kernel density estimation to the prediction of protein–protein interaction sites

Yoichi Murakami* and Kenji Mizuguchi*
National Institute of Biomedical Innovation, Osaka, Japan
Associate Editor: Alfonso Valencia

**ABSTRACT**

**Motivation:** The limited availability of protein structures often restricts the functional annotation of proteins and the identification of their protein–protein interaction sites. Computational methods to identify interaction sites from protein sequences alone are, therefore, required for unraveling the functions of many proteins. This article describes a new method (PSIVER) to predict interaction sites, i.e. residues binding to other proteins, in protein sequences. Only sequence features (position-specific scoring matrix and predicted accessibility) are used for training a Naïve Bayes classifier (NBC), and conditional probabilities of each sequence feature are estimated using a kernel density estimation method (KDE).

**Results:** The leave-one out cross-validation of PSIVER achieved a Matthews correlation coefficient (MCC) of 0.151, an *F*-measure of 35.3%, a precision of 30.6% and a recall of 41.6% on a non-redundant set of 186 protein sequences extracted from 105 heterodimers in the Protein Data Bank (consisting of 36 219 residues, of which 15.2% were known interface residues). Even though the dataset used for training was highly imbalanced, a randomization test demonstrated that the proposed method managed to avoid overfitting. PSIVER was also tested on 72 sequences not used in training (consisting of 18 140 residues, of which 10.6% were known interface residues), and achieved an MCC of 0.135, an *F*-measure of 31.5%, a precision of 25.0% and a recall of 46.5%, outperforming other publicly available servers tested on the same dataset. PSIVER enables experimental biologists to identify potential interface residues in unknown proteins from sequence information alone, and to mutate those residues selectively in order to unravel protein functions.

**Availability:** Freely available on the web at http://tardis.nibio.go.jp/PSIVER/

**Contact:** yoichi@nibio.go.jp; kenji@nibio.go.jp

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

Received on March 12, 2010; revised on June 2, 2010; accepted on June 3, 2010

## 1 INTRODUCTION

Identification of protein–protein interaction sites is not only critical for understanding how proteins perform their biological functions, but also helpful in developing new drugs (Burgoyne and Jackson, 2006; Russell and Aloy, 2008). Experimentally determined protein 3D structures indeed provide important clues to identifying interaction sites and understanding protein functions (Fariselli *et al.*, 2002; Fernandez-Recio *et al.*, 2005; Jones and Thornton, 1997a, 1997b; Neuvirth *et al.*, 2004). However, in spite of significant efforts in determining many protein structures, the number of known 3D structures is still considerably smaller than that of protein sequences [as of February 2010, 514 212 entries in UniprotKB/Swissprot (release 57.13 statistics) compared with 63 093 in the Protein Data Bank (PDB; Berman *et al.*, 2000)]. The limited availability of structures often restricts the identification of interaction sites of proteins and their functional annotation. New computational methods are, therefore, needed to identify interface residues from protein sequences alone and to assist experimental studies, which mutate these residues selectively and assess their effects on interaction.

Is it possible to predict interaction sites, i.e. residues binding to other proteins, from sequence information alone? Ofran and Rost (2003) and Yan *et al.* (2004) have reported that residues involved in interactions tend to form clusters in sequences; within four neighboring residues on either side, 97–98% of interface residues have at least one additional interface residue and 70–74% have at least four additional interface residues. This analysis indicates that neighboring residues of an actual interface residue have high potential for being interface residues, and also suggests that fragments of sequences (referred to as sub-sequences hereafter) potentially have informative features to discriminate between interaction and non-interaction sites. Thus, sequence features contained in sub-sequences are expected to allow us to predict interface residues in sequences. Ofran and Rost (2007) used sub-sequences of nine consecutive residues to develop a neural network (NN)-based method using predicted structural features and evolutionary information. In addition, they improved the method by using a post-processing filter to eliminate all the isolated residues that were predicted as interface but had fewer than seven predicted interface residues in a sub-sequence of 11 residues, and achieved over 90% accuracy (ACC; the proportion of correctly predicted residues) in a 3-fold cross-validation. Yan *et al.* (2004) also used sub-sequences of nine residues to develop a two-stage classifier by combining support vector machines (SVMs) and Bayesian network classifiers trained only on surface residues extracted from protein–protein heterocomplexes, and achieved an ACC of 72% and a Matthews correlation coefficient (MCC) of 0.30. Both methods used sub-sequences of 9 consecutive residues for the prediction of protein–protein interaction sites.

Several other computational methods have been reported, to predict residues binding to other proteins in protein sequences, using machine learning techniques such as NNs, SVMs and the random forest method. Zhou and Shan (2001) and Fariselli *et al.* (2002)

*To whom correspondence should be addressed.



 

applied NNs to classify surface residues into interface and non-interface residues based on analysis of the composition of residues and their structural neighbors. Res *et al*. (2005) trained SVMs to classify whether or not a residue is involved in protein–protein interactions based on evolutionary information on the sub-sequence of nine consecutive residues, and achieved an ACC of 64% in leave-one out cross-validation (LOOCV). Wang *et al*. (2006) also applied SVMs to predict interface residues using features extracted from sequence profiles obtained from the HSSP database (Dodge *et al*., 1998) and evolutionary conservation scores based on phylogenetic trees, and trained the SVMs on a non-redundant set of heterodimeric proteins and achieved an ACC of 65.4% and an MCC of 0.297. Chen and Jeong (2009) applied a random forest-based integrative method to predict interaction sites with a number of features: physicochemical properties, evolutionary conservation scores, residue-based distance matrices and sequence profiles, from protein sequences. Sikic *et al*. (2009) attempted to identify interaction sites in protein sequences using the random forests method and nine consecutive residues in a sequence, and achieved a precision of 85% with a 26% recall and an $F$-measure of 40% in 10-fold cross-validation.

In this article, we present a new machine-learning method to predict residues binding to other proteins in protein sequences using the Naïve Bays classifier (NBC) and kernel density estimation (KDE) with two features; position-specific scoring matrix (PSSM) and predicted accessibility (pA). The NBC has been so far applied to the prediction of DNA/RNA-binding residues (Yan *et al*., 2006; Terribilini *et al*., 2007) and to the prediction of protein interaction partners (Qi *et al*., 2006). Although it ignores cooperative effects of input features (unless cross-features are computed before the implementation), it has been known as an efficient machine learning method that works well for different classification tasks (Mitchell, 1997). In the case of protein–protein interactions, the degree of independence between features is not fully understood and hence an application of this method is worth attempting. The method presented here is the first to apply the NBC with KDE to the prediction of protein–protein interaction sites using PSSM and pA. The method is tested using LOOCV on a non-redundant set of 186 protein sequences extracted from 105 heterodimeric proteins with known interaction sites, and is assessed on an additional set of 72 protein sequences extracted from the protein–protein docking benchmark set version 3.0 (Hwang *et al*., 2008). We will demonstrate that the NBC with KDE contribute to building effective classifiers for imbalanced data and that PSVIER outperforms a publicly available sequence-based prediction server.

## 2 DATASETS AND METHODS

A schematic diagram of the algorithm of the new method presented here is shown in Figure 1. A query sequence enters to two different NBCs created based on PSSM and pA, respectively. Probability ratios for each targeted residue in the sequence are calculated using both NBCs and then normalized on a scale of 0 to 1 using a sigmoid function, and combined to a score for classifying residues as interface (positive class) or non-interface (negative class). Isolated residues predicted as interface are finally filtered out.

### 2.1 A training dataset of 186 protein sequences

A training dataset of protein–protein complexes was extracted from structures of known protein–protein complexes in the PDB (Berman *et al*., 2000)

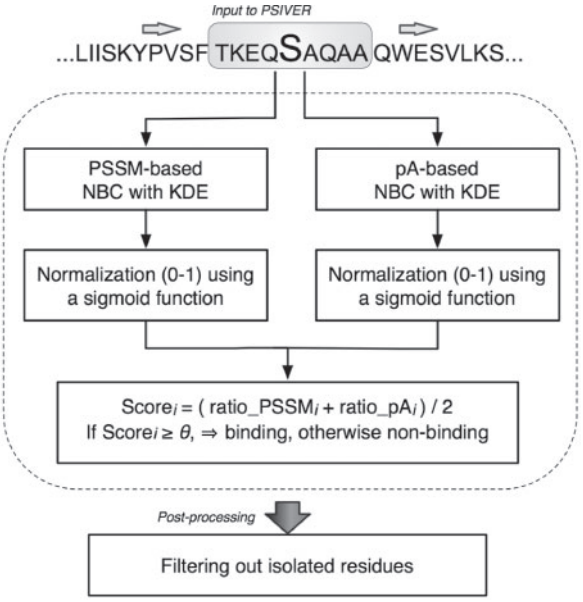


**Fig. 1.** A schematic diagram of the algorithm of our method presented in this article.

solved by X-ray crystallography with a resolution of ≤3.0 Å and with the author-provided biological unit being dimeric. In this work, we focused on (transient) heterodimeric complexes that consist of two non-identical chains. To construct a suitable dataset, we applied the following filters in this order. (i) Following the reasoning that structures with many missing residues may affect learning, any protein complexes whose chains had the missing residues ratio (=*the number of missing residues of a chain listed in REMARK465/the total number of residues of the chain* ×100) ≥ 30% were removed. (ii) UniProtKB/Swiss-Prot (UniProt_Consortium, 2008) accessions and SCCS (SCOP Concise Classification Strings; Murzin *et al*., 1995) were assigned to proteins and any complexes consisting of two chains with either the same UniProt accessions or the same SCCS, or both, were removed, but proteins to which SCCS were not yet assigned were retained. If there were multiple proteins with either the same accession or the same SCCS, or both, only the protein with the highest resolution was retained. (iii) Transmembrane proteins listed in PDBTM (Tusnady *et al*., 2004) were excluded. (iv) Some of the retained structures, determined as dimeric by the authors, may be part of larger oligomeric complexes found in other PDB entries. These structures would have additional interaction sites and might affect the prediction performance of the method. To eliminate such structures as much as possible, each of the retained sequences was scanned against the BLAST pdb database (Altschul *et al*., 1990) and entries with percentage identity (= *match/max{length_sequence1, length_sequence2}*) ≥ 95% were retrieved as related structures and their oligomeric states examined. Any sequences whose related structures were determined in an oligomeric state higher than dimeric were removed. For example, the structure of cyclin-dependent kinase-2 (CDK2) was determined as a dimer complexed with cell cycle-regulatory protein CksHs1 (PDB-ID: 1buh), but it was also determined as a trimer complexed with cycline A and P27klp1 cycline-dependent-kinase inhibitor (PDB-ID: 1jsu) and thus, CDK2 was removed. (v) Interface buried surface accessibility (iBSA) and interface polarity was calculated for each complex by using NACCESS (Hubbard, 1993). Any protein complexes with iBSA of <500 Å$^2$ or ≥2500 Å$^2$ and interface polarity of ≤25% were removed to eliminate structures, which may permanently exist in the living cells, based on the analysis reported by Nooren and Thornton (2003). (vi) The remaining sequences were clustered using BLASTClust (Altschul *et al*., 1990) into groups with ≥25% intra-cluster pair-wise sequence identity over a 90% overlap on both sequences. From each cluster, the highest resolution

protein was selected and in case of a tie, the protein with the longest sequence length was selected. After removal of redundant sequences from each cluster, we obtained a set of 186 protein sequences from 105 (likely transient) heterodimeric protein complexes with sequence identity <25% (Dset186; listed in Supplementary Table S1).

## 2.2 Definition of interface residues

To determine surface residues, the relative solvent accessibility (rSA) was calculated for each residue using NACCESS (Hubbard, 1993), which is an implementation of the Lee and Richards (1971) algorithm. A residue was considered surface if its rSA was <5% (Jones and Thornton, 1997a, 1997b). Using this definition, 76.4% (27 670 residues) of a total of 36 219 residues in Dset186 were defined as surface residues. Furthermore, an interface residue was defined as a surface residue that lost absolute solvent accessibility (SA) of $<1.0$ Å$^2$ on complex formation (Jones and Thornton, 1997a, 1997b). As a result, we defined 15.2% (5517 residues) of the residues in Dset186 as interface.

## 2.3 A test dataset of 72 protein sequences

To construct an independent test set, we used the protein–protein docking benchmark set version 3.0 (Hwang *et al*., 2008), the previous version of which, 2.0 (Mintseris *et al*., 2005) has been used for an assessment of structure-based protein–protein interface prediction servers (Zhou and Qin, 2007). Any sequences showing ≥25% sequence identity over a 90% overlap with any of the sequences in Dset186, using BLASTClust, were removed from the benchmark set. After the removal, 72 protein sequences from 36 protein complexes were obtained (Dtestset72; listed in Supplementary Table S2). Proteins that are part of larger oligomeric complexes are not removed from Dtestset72, because the oligomeric states of unannotated proteins are often unknown in the real prediction problem. According to the definition of surface and interface residues described above, we defined 72.8% (13 213 residues) of a total of 18 140 residues in Dtestset72 as surface residues and 10.6% (1923 residues) as interface.

## 2.4 Naïve Bayes classifier

The proposed method uses a NBC to distinguish between interface and non-interface residues. The NBC is generally known as a simple probabilistic classifier and assumes the independence of features given a class. This assumption can greatly reduce the complexity of the development of the classifier. The sequence features of an $n$-residue sub-sequence (also called a window), with the target residue being described in the centre, were used for the input $X=(x_1\,x_2\ldots x_i\ldots x_n)$ to the NBC. For each target residue, our NBC produced a binary class $C\in\{0,1\}$ where 1 denotes that the target residue was predicted as interface and 0 denotes non-interface. The NBC was trained using a set of labeled training dataset $(X,C)$. In the binary classification, the class for the target residue was determined by comparing two posteriors as in Equation (1)

$$\frac{P(C=1|X=x_1x_2\ldots x_i\ldots x_n)}{P(C=0|X=x_1x_2\ldots x_i\ldots x_n)}=\frac{P(C=1)\prod_{i=1}^{n}P_i(x_i|C=1)}{P(C=0)\prod_{i=1}^{n}P_i(x_i|C=0)} \quad (1)$$

and by taking the logarithm as in Equation (2).

$$\log\frac{P(C=1|X=x_1x_2\ldots x_i\ldots x_n)}{P(C=0|X=x_1x_2\ldots x_i\ldots x_n)}=\log\frac{P(C=1)}{P(C=0)}+\sum_{i=1}^{n}\frac{P_i(x_i|C=1)}{P_i(x_i|C=0)} \quad (2)$$

The target residue of the input $X$ was classified as 1 (interface residues) if

$$\log\frac{P(C=1|X=x_1x_2\ldots x_i\ldots x_n)}{P(C=0|X=x_1x_2\ldots x_i\ldots x_n)}\geq\theta \quad (3)$$

and 0 (non-interface residues) otherwise. The classification threshold $\theta$ determines the trade-off between sensitivity and specificity, and was trained on the training dataset to maximize the prediction performance.

## 2.5 Sequence features

Two sequence features were used as the input to the NBC.

(1) PSSM was obtained using PSI-BLAST (Altschul *et al*., 1997) with an $E$-value threshold of 0.001, for three iterations against the BLAST non-redundant protein sequence database (using BLAST options; −j 3 −d nr −h 0.001). The PSSM describes the evolutionary conservation of the residue positions, and its scores are typically in the range ±7. The input $X$ to the NBC was constructed by concatenating the $n$ rows of the PSSM for each target residue, covering a sub-sequence; $X=(p_{(1,1)}\cdots p_{(1,20)}p_{(2,1)}\cdots p_{(2,20)}\cdots p_{(i,1)}\cdots p_{(i,20)}\cdots p_{(n,20)})$.

(2) Predicted accessibility (pA) of a residue was obtained using SABLE (version 2.0; Wagner *et al*., 2005). The pA represents the rSA of each residues and is expressed on a scale of 0 (fully buried) to 100 (fully exposed). SABLE has been reported to achieve overall correlation coefficients of about 0.66 between actual rSA and pA in independent test sets (Adamczak *et al*., 2005), and it was compared with several other rSA prediction methods and shown to predict most consistently the observed rSA in protein complexes (Porollo and Meller, 2007). The input $X$ to the NBC was constructed by concatenating the pA for each residue in a sub-sequence; $X=(a_1\,a_2..a_i\ldots a_n)$.

Both features were expressed in integers.

## 2.6 Kernel density estimation

Conditional probabilities: $P_i(x_i|C=c)$, the probability that the feature value in the $i$-th position is equal to $x_i$ given class $c$, were estimated using KDE from a set of labeled training data $(X,C)$. KDE is a non-parametric way of estimating the probability density function population (Parzen, 1962). The probability $P_i(x_i|C=c)$ was estimated using Equation (4).

$$P_i(x_i|C=c)=\frac{1}{N_c h}\sum_{j=1}^{N_c}K(x_i,x_{j|i|c}) \quad K(a,b)=\frac{1}{\sqrt{2\pi}}e^{\frac{(a-b)^2}{2h^2}} \quad (4)$$

where K is a Gaussian function kernel with mean zero and variance 1, $N_c$ is the number of the input data $X$ belonging to class $c$, $x_{j|i|c}$ is the feature value in the $i$-th position of the $j$-th input $X=(x_1\,x_2\ldots x_i\ldots x_n)$ in class c, and $h$ is a bandwidth, or a smoothing parameter. To optimally estimate the conditional probabilities, $h$ was optimized on the training dataset.

For comparison with KDE, conditional probabilities $P_i(x_i|C=c)$ were also estimated using Laplace (or add-one) smoothing (LS), which is a simple conventional smoothing method to avoid zero probability as in Equation (5).

$$P_i(x_i|C=c)=\frac{m_{x_i|c}+1}{n_c+k} \quad (5)$$

where $m_{xi|c}$ is the number of the input $X$ that belong to class $c$ and the $i$-th position of which is equal to $x_i$, $n_c$ is the total number of input $X$ in class $c$, and $k$ is the number of possible unique feature values for input $X$.

## 2.7 Sigmoid function

Probability ratios calculated from the two different NBCs, based on PSSM and pA, respectively, were normalized from 0 to 1 using a sigmoid function as in Equation (6).

$$\sigma(\text{ratio})=\frac{1}{1+e^{-(\text{ratio}_i-s)}} \quad (6)$$

where $g$ (gain or slope parameter) is the steepness factor and $s$ sets the mid point of the sigmoid curve along the horizontal axis. The final $Score_i$ for target residue $i$ was determined by combining the two normalized ratios as in Equation (7).

$$\text{Score}_i=\frac{\sigma(\text{ratio\_PSSM}_\text{i})+\sigma(\text{ratio\_pA}_\text{i})}{2.0} \quad (7)$$

### 2.8 Evaluation measures and validation

The following six measures were calculated to assess the NBC performance, using counts of true positives (TP; residues correctly predicted as interface), false positives (FP; residues incorrectly predicted as interface), true negatives (TN; residues correctly predicted as non-interface) and false negatives (FN; residues incorrectly predicted as non-interface).

- Recall, or sensitivity, measures the proportion of the known interface residues that are correctly predicted as interface residues and is defined as $TP/(TP+FN)$.
- Precision measures the proportion of the residues predicted as interface that are known interface residues and is defined as $TP/(TP+FP)$.
- Specificity (SP) measures the proportion of the known non-interface residues that are correctly predicted as non-interface residues and is defined as $TN/(TN+FP)$.
- Accuracy (ACC) is the proportion of the known residues that are correctly predicted in all predictions and is defined as $(TP+TN)/(TP+FN+TN+FP)$.
- MCC indicates the degree of the correlation between the actual and predicted classes of the residues (Matthews, 1975). MCC values range between 1, where all the predictions are correct, and −1 where none are correct. MCC is defined as $((\mathrm{TP}\times\mathrm{TN})-(\mathrm{FP}\times\mathrm{FN}))/\sqrt{(\mathrm{TP}+\mathrm{FP})\times(\mathrm{TP}+\mathrm{FN})\times(\mathrm{TN}+\mathrm{FP})\times(\mathrm{TN}+\mathrm{FN})}$.
- $F$-measure (Hripcsak and Rothschild, 2005) combines precision and recall into their harmonic mean, and is defined as $2\times(\text{precision}\times\text{recall})/(\text{precision}+\text{recall})$.

ACC is known to be inappropriate for an imbalanced dataset since it becomes high when residues in the majority class are favorably predicted. On the other hand, the MCC score is considered to be the best evaluation measure for the overall performance of a method (Baldi *et al*., 2000).

*LOOCV*: the performance of the NBC models trained on Dset186 was assessed using LOOCV. One protein sequence was taken out of the 186 protein sequences and was used as test data, and the remaining sequences were used as training data. This process was repeated 186 times, and the final performance results were averaged over all the test results. To find the best threshold that can optimally classify each residue as interface or non-interface, predictions were made for each test data at a given threshold and the averaged performance measures calculated over the 186 iterations. A range of probability ratios and the final scores from 0 to 1 were examined, in increments of 0.01. In addition, an area under the receiver operator characteristic curve (AUC) (Bradly, 1997) was calculated for each test data, and then the final AUC was averaged over the 186 iterations. To optimize the NBC models, all possible combinations of KDE smoothing parameter $h$, ranging from 0.01 to 0.2 in increments of 0.01, and various lengths of sub-sequences covering 3–21 residues (*wsize*), were examined by LOOCV. Furthermore, to optimally normalize and combine the probability ratios, all possible combinations of sigmoid parameters $g$, ranging from 1 to 10 in increments of 1, and $s$, ranging from −1 to 1 in increments of 0.1, were examined.

*Randomization test*: a randomization test was also carried out to assess whether or not the method was sensitive to the training dataset (Salzberg, 1997). In the current work, the original class labels of the training dataset were replaced with randomly determined class labels, while preserving the class ratio between the number of positive examples and that of negative examples. The performance of NBC trained on the randomized Dset186 using the same method was then compared with the model trained on the original Dset186 to assess over-fitting.

*True test*: the best model, trained on Dset186 and validated by LOOCV, was tested on Dtestset72, which was independent of Dset186. The performance measures were calculated for each protein sequence in the test set, and the averaged performance over the 72 protein sequences was calculated using the best threshold trained on Dset186.

## 3 RESULTS

### 3.1 Performance of NBC with KDE

The NBCs were trained on a set of 186 non-redundant protein sequences (Dset186) extracted from 105 (likely transient) heterodimeric complexes to distinguish between interface and non-interface residues. The dataset contains 5517 interface residues (15.2%, the positive class) and 30 702 non-interface residues (84.8%, the negative class). The proposed prediction method was built on two different NBCs based on PSSM and pA, respectively. The conditional probabilities that the feature value in the $i$-th position is equal to $x_i$ given class $c$ were estimated using KDE. To train the NBCs, we used all possible combinations of KDE smoothing parameter values 0.01–0.2 and window sizes of length 3–21. All possible combinations of $g$ parameter values 1–10 and $s$ parameter values −1.0 to 1.0 in a sigmoid function were used to optimally combine the two probability ratios. The probability ratios calculated from the two different NBCs were then combined using the sigmoid function. The NBCs were evaluated in LOOCV experiments.

To assess over-fitting of the proposed method to the highly imbalanced training dataset, randomization tests were performed on the randomized Dset186 using both the PSSM-based and pA-based NBCs with KDE (Table 1, Model I). Compared with the models trained on the original Dset186 (Table 1, Model II), the models trained on the randomized Dset186, where the class labels of the original dataset were randomly replaced while preserving the original class ratio, performed much worse (in terms of MCC, $F$-measure and AUC). This indicates that the proposed method based on the NBC with KDE was insensitive to the highly imbalanced dataset.

Table 1, Model II shows the best performance of the two NBCs based on PSSM and pA, respectively, and also shows, for comparison purposes, the best performance of two other models whose conditional probabilities were calculated using LS. The PSSM-based NBC with KDE gave slightly higher MCC (0.111) and $F$-measure (33.1%) values than those with LS, although they achieved the same AUC of 0.61. The pA-based NBC with KDE gave significantly higher MCC (0.130) and $F$-measure (34.9%) values than those with LS, and also achieved slightly higher AUC of 0.61. The window size that achieved the highest performance was 9 in all the models. Although the changes in AUC were moderate, the MCC was greatly improved in both models with KDE, especially in the pA-based model. The best NBCs based on PSSM and pA, respectively were combined by optimally integrating two probability ratios for each targeted residues calculated by those NBCs using a sigmoid function. The performance of the integrated models is shown in Table 1, Model III. With two different NBCs integrated, the MCC improved significantly with both KDE and LS. The PSSM+pA model with KDE achieved an MCC of 0.14, a significantly higher value than that with LS.

### 3.2 Reducing isolated positive predictions

To see whether we can improve the performance by eliminating isolated positive predictions, we investigated the sequence neighborhood of the actual interface residues observed in the protein sequences in Dset186. Table 2 shows the distribution of the number of interface residues in a window of 3–11 consecutive residues centered on an interface residue. About 79%, 90% and 96% of

**Table 1.** The prediction performance evaluated on Dset186 using different NBC models (I–IV)

| Model | MCC | Precision (%) | Recall (%) | SP (%) | ACC (%) | AUC | *F*-measure (%) | Window size | Threshold |
|---|---|---|---|---|---|---|---|---|---|
| (I) Models trained on randomized Dset186 (Randomization Test) | | | | | | | | | |
| PSSM with KDE | 0.018 | 18.3 | 2.8 | 98.1 | 83.3 | 0.51 | 4.9 | 7 | −0.50 |
| PSSM with LS | 0.023 | 19.0 | 7.1 | 94.4 | 80.9 | 0.51 | 10.3 | 1 | −0.57 |
| pA with KDE | 0.010 | 15.9 | 33.3 | 68.2 | 62.8 | 0.50 | 21.5 | 3 | −0.73 |
| pA with LS | 0.025 | 17.5 | 18.6 | 84.0 | 73.8 | 0.50 | 18.0 | 1 | −0.71 |
| (II) Models trained on original Dset186 | | | | | | | | | |
| PSSM with KDE | 0.111 | 25.4 | 47.4 | 65.4 | 62.6 | 0.61 | 33.1 | 9 | −0.39 |
| PSSM with LS | 0.101 | 24.7 | 46.7 | 63.7 | 61.2 | 0.61 | 32.3 | 9 | −0.80 |
| pA with KDE | 0.130 | 25.6 | 54.9 | 61.0 | 58.8 | 0.61 | 34.9 | 9 | −0.72 |
| pA with LS | 0.099 | 22.7 | 62.0 | 49.9 | 51.9 | 0.59 | 33.2 | 9 | −0.80 |
| (III) Models created by integrating the best PSSM-based NBC with the best pA-based NBC using a sigmoid function | | | | | | | | | |
| PSSM + pA with KDE | 0.140 | 27.2 | 48.2 | 68.2 | 64.2 | 0.62 | 34.8 | – | 0.37 |
| PSSM + pA with LS | 0.124 | 23.7 | 65.7 | 49.2 | 52.4 | 0.61 | 34.8 | – | 0.38 |
| (IV) The best performance of models (III) after filtering | | | | | | | | | |
| PSSM + pA with KDE | 0.151 | 30.6 | 41.6 | 74.3 | 67.3 | – | 35.3 | – | 0.37 |
| PSSM + pA with LS | 0.130 | 24.1 | 65.9 | 50.3 | 53.2 | – | 35.3 | – | 0.43 |

(I) The best performance of the PSSM-based and pA-based models trained on randomized Dset186 either with KDE or with LS (without KDE) in LOOCV. (II) The best performance of the PSSM-based and pA-based models trained on original Dset186 either with KDE or with LS in LOOCV. (III) The best performance of the models created by integrating the best PSSM-based model with the best pA-based model using a sigmoid function. (IV) The best performance after reducing false positives using filtering, which eliminates isolated positive predictions with $\leq N$ additional interface residues in a sub-sequence. For the PSSM + pA based NBC with KDE, *wsize* (a length of a sub-sequence) = 11, $N = 2$ and threshold = 0.37 achieved the best performance. For the PSSM + pA-based NBC with LS, *wsize* = 7, $N = 1$ and threshold = 0.43 achieved 0.43.

**Table 2.** The ratio of interface residues observed within a sub-sequence with an interface residue being described in the centre

| | Number of interface residues observed within a sub-sequence belonging to the positive class (%) | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Window size | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 3 | 21.35 | 47.22 | 31.48 | | | | | | | | |
| 5 | 10.42 | 24.72 | 28.55 | 23.60 | 12.80 | | | | | | |
| 7 | 4.42 | 13.56 | 20.81 | 25.09 | 19.32 | 11.55 | 5.38 | | | | |
| 9 | 2.23 | 7.94 | 13.90 | 19.27 | 21.32 | 16.77 | 11.09 | 5.18 | 2.47 | | |
| 11 | 1.87 | 6.00 | 10.80 | 15.35 | 17.20 | 17.91 | 13.88 | 8.50 | 4.82 | 2.65 | 1.21 |

the observed interface residues had at least one additional interface residue in a sub-sequence of 3, 5 and 7 residues, respectively. Furthermore, about 98% of the observed interface residues had at least one additional interface residue and about 76% had at least four interface residues in a window of nine consecutive residues (within four residues on either side). These results are similar to those reported by Ofran and Rost (2003) and Yan *et al*. (2004), and clearly indicate that interface residues tend to form clusters in sequences. A filtering methods has been previously proposed to eliminate isolated positive predictions with $\leq N$ additional interface residues in a sub-sequence (Ofran and Rost, 2003; Res and Lichtarge, 2005).

In the current work, we implemented a similar filtering method. To maximize the ability of the filtering, all possible combinations of *wsize* ranging from 3 to 11 and the cutoff $N$ ranging from 1 to (*wsize* + 1)/2 were examined. As a result, when we converted all isolated positive predictions with ≤2 other positive residues in a window of 11 residues, the MCC of the PSSM+pA-based NBC with KDE and that with LS increased to 0.151 and 0.130, respectively, and the *F*-measure also increased from 34.8% to 35.3% in both models (Table 1, Model, IV). This filtering method may convert some of the true positives to false negatives, yet overall, it was shown to produce the class labels better correlated with the actual class labels (Table 1). Therefore, the PSSM+pA based NBC with KDE that achieved the highest MCC of 0.151 was selected as the final model (and named PSIVER, Protein–protein interaction SItes prediction serVER).

### 3.3 Predicting binding residues in Dtestset72

A test that makes predictions for protein sequences not related to those used in training enables us to compare the performance of our method directly to that of other previously published methods, if their implementations are publicly available. For this purpose, we created an independent dataset (Dtestset72) from the dockingbenchmark set v3.0 (Hwang *et al*., 2008), and made predictions using the best model trained on Dset186 (PSIVER). For comparison, we chose ISIS (Ofran and Rost, 2007) and SPPIDER (Porollo and Meller, 2007, the sequence-based method was introduced as an experimental function on 25 November 2008).

These servers have been trained on datasets different from Dset186; ISIS was trained on a set of 1134 protein chains extracted from 333 transient protein complexes, and SPPIDER used a set of 436 chains including 262 from heterocomplexes and 173 from homocomplexes. The interface residues were differently defined in these three servers; ISIS used ≤6 Å distance constraints for interatomic contacts between any atoms of the target chain and any atoms of the interacting chains (Ofran and Rost, 2007), and SPPIDER defined the interface residues based on >4% rSA change and >5 Å$^2$ SA change on complex formation of the exposed surface residues, which were defined if their rSA was >5% (Porollo and Meller, 2007). According to the definitions of ISIS and SPPIDER, 15.4% (2791 residues) and 9.0% (1629 residues) of the residues in Dtestset72 were defined as interface residues, respectively. In our definition shown in section 2.2, 10.6% (1923 residues) were defined as interface residues. The ISIS interface definition produces a larger number of interface residues, and this could give a higher chance for predictors to correctly predict interface residues. Thus, these three serves were assessed on Dtestset72 using the three different interface definitions with default parameters. The default threshold for PSIVER was 0.37, which gave the best performance in the LOOCV. The prediction results were evaluated based on MCC and *F*-measure (Table 3).

**Table 3.** The prediction performance of PSIVER, ISIS (Ofran and Rost, 2007) and SPPIDER (Porollo and Meller, 2007) tested on Dtestset72 with different interface definitions

| Method | MCC | Precision (%) | Recall (%) | SP (%) | ACC (%) | F-measure (%) |
|---|---|---|---|---|---|---|
| PSIVER interface definition (>1.0 Å$^2$ SA change) | | | | | | |
| PSIVER | 0.135 | 25.0 | 46.5 | 69.3 | 66.1 | 32.5 |
| ISIS | 0.091 | 21.0 | 35.0 | 76.2 | 70.9 | 26.3 |
| SPPIDER | 0.081 | 20.4 | 45.4 | 64.7 | 61.7 | 24.6 |
| ISIS interface definition (<6 Å interatomic distance) | | | | | | |
| PSIVER | 0.129 | 30.6 | 44.3 | 69.5 | 64.7 | 31.4 |
| ISIS | 0.091 | 26.9 | 33.3 | 76.6 | 68.9 | 27.9 |
| SPPIDER | 0.072 | 26.0 | 43.1 | 63.9 | 60.6 | 27.1 |
| SPPIDER interface definition (>4% rSA and >5 Å$^2$ SA change) | | | | | | |
| PSIVER | 0.130 | 22.2 | 47.0 | 69.0 | 66.4 | 25.6 |
| ISIS | 0.097 | 18.9 | 36.6 | 76.1 | 71.9 | 23.2 |
| SPPIDER | 0.077 | 17.8 | 45.6 | 63.6 | 62.0 | 21.9 |

As shown in Table 3, the performance values of the three servers changed little with different interface definitions. Although ISIS produced a higher ACC score than PSIVER and SPPIDER, ACC is generally considered to be an inapropriate evaluation measure for imbalanced datasets (Baldi *et al*., 2000); for example, if all residues in Dtestset72 were predicted as non-interface, ACC would be 84.8%. The MCC score represents how well predictions correlate with observed class labels, and is considered to be the best evaluation measure for the overall performance of a method (Baldi *et al*., 2000). Based on MCC and also *F*-measure values, PSIVER outperformed ISIS and SPPIDER as shown in Table 3. The performance of each protein sequence in Dtestset72 is shown in Table S2. We further benchmarked PSIVER against ISIS and SPPIDER using subsets of Dtestset72. As highlighted by Ezkurdia *et al.* (2009), the performance of predictors depends to a certain extent on the dataset used for testing. Thus, we assessed the performance of the three servers on five non-overlapping test sets of 30 or 50 protein sequences, randomly chosen from Dtestset72. As shown in Table 4, the performance of the servers tested on different subsets did change indeed, but PSIVER always outperformed the other servers.

## 4 DISCUSSION AND CONCLUSION

This article presents a new machine learning method (PSIVER) for the prediction of protein-protein interaction sites, using the NBC and KDE with two sequence features, PSSM and pA. The proposed method was assessed using LOOCV, which is generally known to be an almost unbiased (but expensive) validation method, on a dataset of 186 non-redundant protein sequences obtained from 105 (likely transient) heterodimers (Dset186) and by prediction on a dataset of 72 protein sequences not related to those used in training (Dtestset72). From Dset186, we excluded permanent homocomplexes (Jones and Thornton, 1997a, 1997b; Nooren and Thornton, 2003), because protein–protein interactions regulating a variety of functions in the cells tend to be transient. Both datasets were highly imbalanced ones; the proportion of the residues known to be interface in Dset186 and Dtestset72 was 15.2% and 10.6%, respectively. These figures were considerably smaller than those in the datasets used by other published methods (Ofran and Rost, 2007; Res *et al*., 2005; Sikic *et al*., 2009; Wang *et al*., 2006; Yan *et al*., 2004). Imbalanced datasets are generally considered to cause over-fitting to the majority class and affect the performance.

To deal with these imbalanced datasets, we applied, in the current work, the NBC and KDE to the prediction of protein–protein

**Table 4.** Prediction results for five subsets of 30 or 50 non-overlapping protein sequences randomly chosen from Dtestset72

| Method | Subset1 | Subset2 | Subset3 | Subset4 | Subset5 |
|---|---|---|---|---|---|
| Five subsets of 30 non-overlapping sequences | | | | | |
| PSIVER | 0.125 (26.5) | 0.118 (26.9) | 0.138 (27.8) | 0.136 (26.6) | 0.144 (28.8) |
| ISIS | 0.102 (24.7) | 0.086 (24.6) | 0.075 (23.8) | 0.118 (24.9) | 0.106 (26.6) |
| SPPIDER | 0.059 (22.5) | 0.068 (24.4) | 0.044 (20.6) | 0.054 (22.2) | 0.080 (24.6) |
| Five subsets of 50 non-overlapping sequences | | | | | |
| PSIVER | 0.140 (27.7) | 0.136 (27.2) | 0.150 (29.3) | 0.136 (27.1) | 0.145 (29.1) |
| ISIS | 0.098 (24.6) | 0.082 (23.7) | 0.090 (24.8) | 0.091 (24.4) | 0.099 (26.1) |
| SPPIDER | 0.077 (23.5) | 0.091 (24.0) | 0.081 (24.5) | 0.083 (23.4) | 0.076 (23.9) |

The MCC score and *F*-measure (in brackets, %) are shown.

interaction sites. The NBC assumes that the sequence features of each residue in a sub-sequence of proteins are independent of each other. Although this assumption is not always true, it does dramatically reduce the complexity of the model development and classification task, since the estimation of high-dimensional probabilities is reduced to that of one-dimensional conditional probabilities. Furthermore, the conditional probabilities for each feature in each position of a sub-sequence were estimated using KDE, which can effectively extrapolate a probability density function from a collection of feature values. These two elements used in the proposed method contributed to dealing with the imbalanced data effectively, as demonstrated clearly in the randomization test.

Although we included only those protein structures that were annotated as dimeric by the authors and we also eliminated any protein whose related structures were determined in an oligomeric state higher than dimeric, some of the proteins in Dset186 and Dtestset72 might still have additional protein–protein interaction sites and thus might have affected the performance of the method. Nevertheless, the method had to be assessed based on the class labels of residues observed in available protein complexes. As a result of the assessment, the highest performing model used both features over a window of nine residues, and achieved an MCC of 0.140, an $F$-measure of 34.8% and an AUC of 0.62 in LOOCV. The calculation time required for LOOCV was 14 min 23 s (about 4.7 s for one testing) on the system of a single 3GHz dual core CPU and 2 GB memory. The calculation time is likely to be 1000 times faster than that required for an equivalent operation using SVMs (data not shown). When filtered out all isolated positive predictions with $\leq$2 additional interface residues in a window of 11 residues, the model achieved an MCC of 0.151 and an $F$-measure of 35.3%.

An objective comparison with previously published methods is an important issue in the development of almost all computational methods. Ofran and Rost (2007) and Sikic *et al*. (2009) pointed out that direct comparison with the performance reported in the literature is nearly impossible, since different datasets, different definition of interaction sites, and different cross-validation methods were used, and also since not all method implementations were publicly available. Even when the details of the training datasets were available, those we have examined contained a number of highly similar sequences ($\geq$25% sequence identity over a 90% overlap of both sequences using BLASTClust) or were inconsistent in some other ways, for example, the dataset of 99 polypeptide chains used by Chen and Jeong (2009), upon inspection, was found to contain a number of highly similar sequences [e.g. PDB-IDs; 1got (chain B) and 1mct (chain A) were the same proteins as PDB-IDs; 1gg2 (chain B) and 1avw (chain A), respectively], and therefore, we had to abandon the use of these datasets, as a reliable validation and comparison would have been impossible. Instead, we devised a true test that makes predictions for protein sequences not related to those used in training. Such a test enables us to compare the performance of our method directly to that of other previously published methods, if their implementations are publicly available. We compared the performance of PSIVER on Dtestset72 with that of the ISIS server (Ofran and Rost, 2007) and the sequence-based version of SPPIDER (Porollo and Meller, 2007) with default parameters. This test showed that PSIVER outperformed ISIS and SPPIDER, with an MCC of 0.135 and an $F$-measure of 32.5% (PSIVER interface definition; Table 3), and the results of the individual proteins (Supplementary Table S2) suggested no bias from the training using transient heterodimeric complexes alone, since PSIVER outperformed the other servers that had been trained on different types of oligomeric complexes.

The proposed method, therefore, enables experimental biologists to identify potential interface residues in unknown proteins from sequences alone, and to design targeted mutations in order to unravel protein functions. Furthermore, incorporating PSIVER predictions with structural information (if available) will give further valuable insights into the identification of protein–protein interaction sites and the functional annotation of unknown proteins. PSIVER is freely available at http://tardis.nibio.go.jp/PSIVER/.

## ACKNOWLEDGEMENTS

We thank Shandar Ahmad for carefully reading the article and helpful comments.

*Funding*: Industrial Technology Research Grant Program in 2007 from New Energy and Industrial Technology Development Organization (NEDO) of Japan in part.

*Conflict of Interest*: none declared.

## REFERENCES

Adamczak,R. *et al*. (2005) Combining prediction of secondary structure and solvent accessibility in proteins. *Proteins*, **59**, 467–475.

Altschul,S.F. *et al*. (1997) Gapped BLAST and PSI- BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Baldi,P. *et al*. (2000) Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, **16**, 412–424.

Berman,H.M. *et al*. (2000) The Protein Data Bank. *Nucleic Acids Res.*, **28**, 235–242.

Burgoyne,N.J. and Jackson,R.M. (2006) Predicting protein interaction sites: binding hot-spots in protein-protein and protein-ligand interfaces. *Bioinformatics*, **22**, 1335–1342.

Chen,X.W. and Jeong,J.C. (2009) Sequence-based prediction of protein interaction sites with an integrative method. *Bioinformatics*, **25**, 585–591.

Dodge,C. *et al*. (1998) The HSSP database of protein structure-sequence alignments and family profiles. *Nucleic Acids Res.*, **26**, 313–315.

Fariselli,P. *et al*. (2002) Prediction of protein–protein interaction sites in heterocomplexes with neural networks. *Eur. J. Biochem.*, **269**, 1356–1361.

Ezkurdia,I. *et al*. (2009) Progress and challenges in predicting protein-protein interaction sites. *Brief. Bioinform.*, **10**, 233–246.

Fernandez-Recio,J. *et al*. (2005) Optimal docking area: a new method for predicting protein-protein interaction sites. *Proteins*, **58**, 134–143.

Hripcsak,G. and Rothschild,A.S. (2005) Agreement, the f-measure, and reliability in information retrieval. *J. Am. Med. Inform. Assoc.*, **12**, 296–298.

Hubbard,S.J and Thornton,J.M. (1993) *'NACCESS', Computer Program*. Department Molecular Biology. University College, London.

Hwang,H. *et al*. (2008) Protein-protein docking benchmark version 3.0. *Proteins*, **73**, 705–709.

Jones,S. and Thornton,J.M. (1997a) Prediction of protein-protein interaction sites using patch analysis. *J. Mol. Biol.*, **272**, 133–143.

Jones,S. and Thornton,J.M. (1997b) Analysis of protein-protein interaction sites using surface patches. *J. Mol. Biol.*, **272**, 121–132.

Lee,B. and Richards,F.M. (1971) The interpretation of protein structures: estimation of static accessibility. *J. Mol. Biol.*, **55**, 379–400.

Matthews,B.W. (1975) Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta*, **405**, 442–451.

Mintseris,J. *et al*. (2005) Protein-Protein Docking Benchmark 2.0: an update. *Proteins*, **60**, 214–216.

Mitchell,T. (1997) *Machine Learning*. McGraw Hill Companies. Inc., New York.

Murzin,A.G. *et al*. (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, **247**, 536–540.

Neuvirth,H. *et al*. (2004) ProMate: a structure based prediction program to identify the location of protein-protein binding sites. *J. Mol. Biol.*, **338**, 181–199.

Nooren,I.M. and Thornton,J.M. (2003) Structural characterisation and functional significance of transient protein-protein interactions. *J. Mol. Biol.*, **325**, 991–1018.
Ofran,Y. and Rost,B. (2003) Predicted protein-protein interaction sites from local sequence information. *FEBS Lett.*, **544**, 236–239.
Ofran,Y. and Rost,B. (2007) ISIS: interaction sites identified from sequence. *Bioinformatics*, **23**, e13–e16.
Parzen,E. (1962) On estimation of a probability density function and mode. *Ann. Math. Stat.*, **33**, 1065–1076.
Porollo,A. and Meller,J. (2007) Prediction-based fingerprints of protein-protein interactions. *Proteins*, **66**, 630–645.
Qi,Y. *et al*. (2006) Evaluation of different biological data and computational classification methods for use in protein interaction prediction. *Proteins*, **63**, 490–500.
Res,I. *et al*. (2005) An evolution based classifier for prediction of protein interfaces without using protein structures. *Bioinformatics*, **21**, 2496–2501.
Russell,R.B. and Aloy,P. (2008) Targeting and tinkering with interaction networks. *Nat. Chem. Biol.*, **4**, 666–673.
Salzberg,S.L. (1997) On comparing classifiers: pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery*, **1**, 317–328.
Sikic,M. *et al*. (2009) Prediction of protein-protein interaction sites in sequences and 3D structures by random forests. *PLoS Comput. Biol.*, **5**, e1000278.
Terribilini,M. *et al*. (2007) RNABindR: a server for analyzing and predicting RNA-binding sites in proteins, *Nucleic Acids Res.*, **35**, W578–W584.
Tusnady,G.E. *et al*. (2004) Transmembrane proteins in the Protein Data Bank: identification and classification. *Bioinformatics*, **20**, 2964–2972.
Wagner,M. *et al*. (2005) Linear regression models for solvent accessibility prediction in proteins. *J. Comput. Biol.*, **12**, 355–369.
Wang,B. *et al*. (2006) Predicting protein interaction sites from residue spatial sequence profile and evolution rate. *FEBS Lett.*, **580**, 380–384.
Yan,C. *et al*. (2004) A two-stage classifier for identification of protein-protein interface residues. *Bioinformatics*, **20** (Suppl. 1), i371–i378.
Yan,C. *et al*. (2006) Predicting DNA-binding sites of proteins from amino acid sequence. *BMC Bioinformatics*, **7**, 262.
Zhou,H.X. and Qin,S. (2007) Interaction-site prediction for protein complexes: a critical assessment. *Bioinformatics*, **23**, 2203–2209.
Zhou,H.X. and Shan,Y. (2001) Prediction of protein interaction sites from sequence profile and residue neighbor list, *Proteins.* **44**, 336–343.