A comparative study on filtering protein secondary structure prediction

Petros Kountouris, Michalis Agathocleous, Vasilis J. Promponas, Georgia Christodoulou, Simos Hadjicostas, Vassilis Vassiliades and Chris Christodoulou

Abstract—Filtering of protein secondary structure prediction aims to provide physicochemically realistic results, while it usually improves the predictive performance. We performed a comparative study on this challenging problem, utilising both machine learning techniques and empirical rules and we found that combinations of the two lead to the highest improvement.

Index Terms—protein secondary structure prediction, filtering, machine learning, structural bioinformatics, bidirectional recurrent neural networks

1 Introduction

Knowledge of the three-dimensional (3D) structure of a protein is crucial to understand its function. However, the rapid growth of the number of protein sequences has far outpaced the experimental determination of their structures. Thus, there is a growing need for a computational approach to the problem of protein structure prediction. The prediction of secondary structure, the local structure commonly defined by hydrogen bond patterns and local geometry, is a critical first step towards this end and, therefore, it has attracted a great amount of interest over the past 50 years. With respect to their secondary structure, amino acid residues in protein chains are usually assigned into three main classes, namely helix, extended and coil/loop.

Over the past 20 years, the secondary structure predictive accuracy has improved significantly through the use of machine learning techniques [1] and evolutionary information from multiple sequence alignments [2]. Several artificial neural network (ANN) architectures have been used, such as feed-forward ANNs [2], [3], bidirectional recurrent ANNs (BRNNs) [4], [5], [6] and cascade-correlation ANNs [7], whilst support vector machines (SVMs) have been proven successful over the past decade [8], [9], [10]. Other methods used hidden Markov models (HMMs) [11], [12], multiple linear regression [13], [14] and nonlinear dynamic systems [15], whereas methods like JPred [16] make consensus secondary structure prediction. More recently, knowledge-based methods, such as PROTEUS [17] and HYPROSP [18], utilised structural information, whilst the predictive accuracy was further improved through the use of remote homology information [19]. The three-state predictive accuracy (Q_3) is currently around 80%, whereas the segment overlap (SOV) [20] is around 74, recently reviewed in [21].

Several protein secondary structure prediction (PSSP) methods used a multi-step process and the final step includes filtering the predictions to improve the quality of the results. This is accomplished by removing conformations that are physicochemically unlikely. For instance, helical conformations in proteins are repetitive structures that consist of at least three, four or five residues for 3_{10}-helix, α-helix and π-helix, respectively. Since the different types of helices are usually grouped in a single category by PSSP methods, a predicted helical structure would be expected to have a minimum number of three residues in order to fulfill geometric and hydrogen-bonding requirements. Hence, predictions of single helical residues are physicochemically unrealistic, because one residue cannot make a turn in order to form a helix. To tackle this problem, both machine learning algorithms [22] and empirical rules have been used in the past [2], [7], [23]. Despite being employed widely, there is no clear indication for the most effective filtering method in PSSP and, to the best of our knowledge, no study has been carried out to find the most suitable filtering technique.

In this paper, we perform a comparative study on the challenging problem of filtering PSSP, utilising both widely used empirical smoothing rules and machine learning techniques. Using an ensemble of six BRNNs with per-residue weight updating [24], we predict the secondary structure on two non-redundant, non-homologous datasets and, subsequently, we apply a number of filtering techniques

- *C. Christodoulou is with the Department of Computer Science, University of Cyprus, P.O. Box 20537, 1678 Nicosia, Cyprus. E-mail: cchrist@cs.ucy.ac.cy*
- *P. Kountouris, M.Agathocleous, G. Christodoulou, S. Hadjicostas and V. Vassiliades are with the Department of Computer Science, University of Cyprus, Cyprus*
- *V. J. Promponas is with the Department of Biological Sciences, University of Cyprus, Cyprus.*

to smooth the predictions. Importantly, the SOV increases significantly in most cases. On the other hand, some classifiers increase the per-residue accuracy, whereas others decrease it. The Logistic function, the Multi-layer Perceptron (MLP) and the SVMs were found to be superior to the tested methods in terms of both Q_3 and SOV score. Notably, the results improve even further when we use combinations of machine learning algorithms and empirical filtering rules.

2 Materials and Methods

2.1 Dataset and preprocessing

The study was carried out using two non-redundant, non-homologous datasets. The first, denoted as CB513 throughout this paper, was compiled by Cuff and Barton [25] and contains 513 protein chains of known 3D structure, which have less than 25% sequence identity to ensure that homologous sequences are not included in the training set. CB513 has been widely utilised to compare several secondary structure prediction methods in the literature, e.g. [9], [10]. Because of its small size, this dataset was used to study the impact of various input coding schemes. The second dataset was PDB-Select25 (version October 2008) [26], a set of 4018 high quality X-ray and NMR structures with less than 25% sequence similarity. From the initial set, we removed chains for which DSSP [27] did not return valid output, which resulted in a final set of 3977 protein chains. Even though most typical PSSP methods are optimised to work with globular proteins, we decided not to remove around 90 transmembrane proteins contained in this dataset.

Secondary structure was assigned based on the experimentally determined 3D structures using the established DSSP program [27], which assigns secondary structure in eight states: H (α-helix), G (3_{10}-helix), E (extended β-strand), B (isolated β-bridge), T (turn), S (bend) and "_" (other/coil). Most of the existing methods predict secondary structure using a three-state assignment. Hence, we reduce the above representation into a three-state scheme, by assigning H, G, and I to the helix state (H), E and B to the extended state (E) and the rest to the loop state (L). This three-state representation is also followed by the EVA secondary structure prediction validation server [28].

Since their first use in PSIPRED [3], PSI-BLAST's [29] position specific scoring matrices (PSSMs) are utilised by the majority of PSSP methods. PSSMs are constructed using multiple sequence alignments and provide crucial evolutionary information about the protein structure. PSSMs consist of $N \times 20$ elements, where the N rows correspond to the length of the of amino acid sequence and the columns correspond to the 20 standard amino acids. We generated a PSSM for each chain in the dataset using the BLOSUM62 substitution matrix [30] with an e-value of 0.001 and three iterations against the NCBI non-redundant (nr) database, downloaded in February 2009. The database was filtered by *pfilt* [31] to remove low complexity regions, transmembrane spans and coiled coil regions. This filtering could be important for dealing with transmembrane proteins.

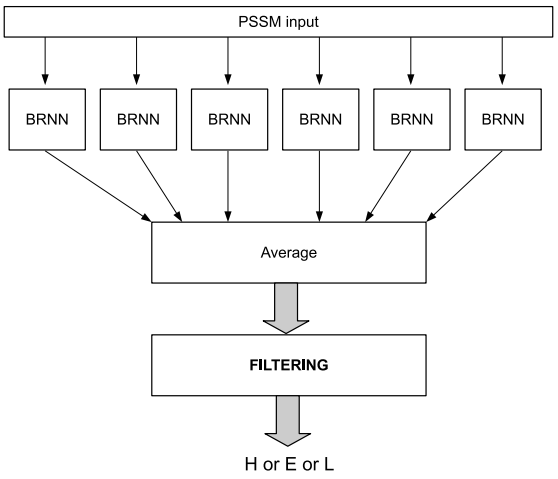

Fig. 1. The architecture of the ensemble of BRNNs, followed by the filtering of the output. The PSSM values are given as input to six BRNNs, which predict the secondary structure of each residue in the amino acid sequence. Subsequently, the outputs are averaged and are given as input to the filtering methods investigated in this study.

2.2 Ensemble of Bidirectional Recurrent Neural Networks

ANNs were first employed for PSSP in [1]. Since then, they have been widely applied in this domain under different settings [2], [3]. In 1999, Baldi and colleagues [4] implemented a BRNN architecture to predict secondary structure, which was proven one of the most successful approaches in the field. The predictive accuracy was boosted in a subsequent study through the use of an ensemble of eleven BRNNs [5]. The BRNN architecture consists of a feed forward neural network (FFNN) and two recurrent neural networks (RNNs). More specifically, there is a Forward RNN (FRNN) which processes the local information contained at the left of the local window (upstream information), whereas a Backward RNN (BWRNN) takes into account the amino acids at the right-hand side of the local window (downstream information).

In a recent study [24], we used the same BRNN architecture of Baldi et al. [4], but we proposed a modified training procedure. In brief, rather than updating network weights after the presentation of the entire protein chain (as performed by Baldi et al. [4]), we update the weights at every residue. This training procedure resulted in a significant increase

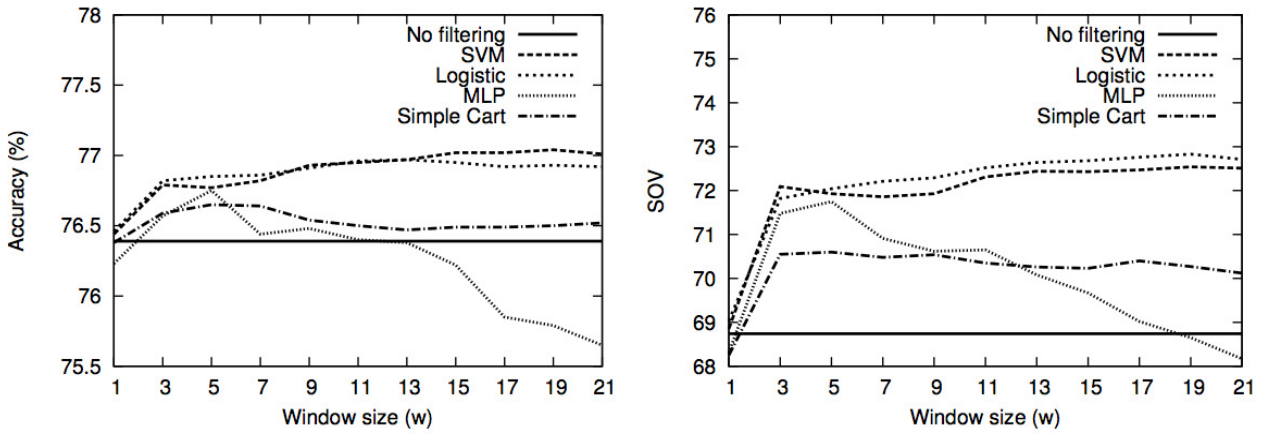

Fig. 2. Experiments with different local window sizes for filtering PSSP (see text for more information) using four machine learning algorithms on the CB513 dataset. The predictive accuracy (left) and the SOV score (right) strongly depend on the size of the local window used for filtering.

of the predictive accuracy when a single BRNN is considered. In this paper, we employ an ensemble of six BRNNs but, rather than using a single residue in the central FFNN, we utilised a local window of five residues, centred around the residue of interest. Thus, the classifier incorporates the local information contained in the neighbouring residues. Each BRNN returns three real values for the central residue of the local window, one for each secondary structure state. Subsequently, the corresponding outputs for each state are averaged and, therefore, the output of the ensemble is an array of three values for each residue. The resulting predictions are then used for filtering, which is the main focus of this paper. The overall architecture is illustrated in Figure 1.

2.3 Filtering techniques

We evaluate an array of machine learning algorithms to identify those that perform better in filtering PSSP. More specifically, we employed the WEKA software package [32] to test the following classification algorithms: Naive Bayes, Simple Cart, Radial Basis Function (RBF) network initialised by k-Means clustering ($k = 3$), IBk (nearest neighbour algorithm) with $k = 3$, MLP, Random Forest, J48 decision tree (C4.5) and Logistic function. The latter is an implementation of a multinomial logistic regression function with a ridge estimator to avoid overfitting [33].

Additionally, we employed SVMs to filter the predictions. More specifically, we used the default one-against-one multi-class SVM provided by the LibSVM software package [34]. We utilised the RBF kernel function and we set the kernel parameter, γ, at $\frac{1}{3w}$, where w is the length of the local window. Finally, the misclassification penalty parameter, C, was equal to unity.

Moreover, we used a meta-classifier to combine two or more from the above algorithms by using several voting schemes. More specifically, we employed the following voting schemes (implemented in WEKA): (i) Majority Vote, (ii) Maximum probability, (iii) Minimum probability, (iv) Product of probabilities and (v) Average of probabilities. The combinations of filtering methods are selected based on the performance of the individual learning techniques and they are discussed in Section 3. For the same purpose, we implemented a HMM using the Viterbi algorithm [35]. A detailed description of the above algorithms is beyond the scope of this article, but a comprehensive survey for many of them in the context of their potential in data mining was written by Wu and co-workers [36]. All algorithms from WEKA and LibSVM were used with the default parameters. The results presented here can be possibly improved by optimising each algorithm individually.

The ultimate goal of a classification algorithm is not to achieve high training accuracy, but to classify successfully previously unseen examples. Hence, we use n-fold cross-validation to estimate the generalisation error. More specifically, we divide the training set into n subsets and, sequentially, we use $n - 1$ for training and the remaining one for testing. This procedure is repeated n times, until all subsets are used once for testing. In this paper, we report the results from 10-fold cross-validation on the CB513 dataset and 5-fold cross-validation on the PDB-Select25 dataset. For both datasets, the folds have similar representation of helical, extended and loop residues. Moreover, in the case of CB513, we ensure similar distributions of small/large protein chains as well as of the four main

TABLE 1
Filtering PSSP on the CB513 dataset using the method shown in the first column, sorted by the highest predictive accuracy (Q_3). w is the local window size for filtering that maximises the SEL score for each method. In bold are the highest scores in the corresponding column.

Filtering method	w	Q_3 (%)	Q_H (%)	Q_E (%)	Q_L (%)	SOV	SOV_H	SOV_E	SOV_L	C_H	C_E	C_L	SEL
LibSVM	19	**77.04**	78.02	65.81	82.40	72.54	71.92	68.64	70.80	0.718	0.635	0.583	74.79
Logistic	19	76.93	78.69	67.33	80.76	**72.83**	72.43	68.74	71.31	0.716	0.633	0.582	**74.88**
MLP	5	76.75	77.94	66.02	81.68	71.75	70.72	68.77	70.17	0.717	0.626	0.579	74.25
Simple Cart	5	76.65	79.06	66.78	80.11	70.60	70.91	67.57	69.66	0.712	0.625	0.580	73.63
SS-filt	–	76.43	75.98	62.23	84.58	71.25	70.53	66.92	70.56	0.711	0.622	0.578	73.84
No filtering	–	76.39	77.12	63.53	82.87	68.74	68.75	66.07	69.63	0.706	0.618	0.580	72.57
WH-filt	–	76.24	74.77	62.33	**85.06**	69.43	67.31	65.90	70.35	0.710	0.616	0.577	72.84
RBF Network	1	76.23	**81.52**	69.88	75.44	69.30	71.73	68.84	66.86	0.705	0.618	0.578	72.77
Naive Bayes	3	76.10	78.68	**71.99**	76.27	71.75	71.37	70.46	68.57	0.710	0.629	0.561	73.92
Viterbi	1	75.98	77.77	63.93	81.14	69.59	69.58	65.57	67.53	0.705	0.619	0.562	72.79
J48	3	75.98	78.97	65.87	79.10	68.53	69.33	66.84	67.84	0.704	0.610	0.569	72.25
Random Forest	19	75.19	79.64	68.58	75.23	66.76	68.58	66.68	64.94	0.696	0.608	0.550	70.98
IBk (k=3)	13	72.03	78.67	62.64	71.81	62.66	67.98	62.71	60.86	0.640	0.566	0.499	67.34

SCOP classes (all-α, all-β, $\alpha+\beta$ and α/β) [37]. The subsets are available on request.

Additionally, we filtered PSSP using two empirical techniques that were previously used to filter other PSSP methods. For this purpose, we implemented a custom software, which uses regular expressions to perform the empirical filtering step. The first set of smoothing rules (denoted as SS-filt) was compiled by Salamov and Solovyev [23] and contains the following three filtering rules: (i) replace single helical residues with loop, i.e. !H H !H → !H C !H; (ii) replace single strand residues with loop, i.e. !E E !E → !E C !E; and (iii) all strands of length two surrounded with helices are replaced by helices, i.e. H E E H → H H H H. The second set of filtering rules (denoted as WH-filt) consists of ten empirical rules that have been used to filter PSSP from DESTRUCT and can be found in [7]. The above rules are based on empirical knowledge and aim to remove physicochemically unrealistic predictions.

Finally, combinations of machine learning algorithms and empirical algorithms were also used. More specifically, a machine learning algorithm was applied at first and, subsequently, an empirical rule was used to filter the outputs. This approach resulted in further improvement of the machine learning algorithms as discussed below.

2.4 Prediction accuracy assessment

To facilitate an objective comparison of the above learning methods, several measures were used to assess the performance of each filtering technique, most of them defined in the EVA server [28]. Q_3 is the three-state overall percentage of correctly predicted residues:

$$Q_3 = 100 \frac{1}{N_{res}} \sum_i M_{ii}, \tag{1}$$

where N_{res} is the total number of residues and M_{ij} is the number of residues observed in state i and predicted in state j, with i and $j \in \{H, E, L\}$ (i.e. M_{ii} is the number of residues predicted correctly in state i). Moreover, we calculate the per-state accuracy, as the percentage of correctly predicted residues in a particular state:

$$Q_i = 100 \frac{M_{ii}}{obs^i} \tag{2}$$

where obs^i is the number of residues observed in state i. Additionally, the Matthews correlation coefficient [38], C_i, provides a measure for the performance at each state:

$$C_i = \frac{p_i n_i - u_i o_i}{\sqrt{(p_i + u_i)(p_i + o_i)(n_i + u_i)(n_i + o_i)}}$$

$$\text{with } p_i = M_{ii}, \qquad n_i = \sum_{j \neq i} \sum_{k \neq i} M_{jk}, \tag{3}$$

$$o_i = \sum_{j \neq i} M_{ji} \quad \text{and} \quad u_i = \sum_{j \neq i} M_{ij}.$$

TABLE 2
Filtering PSSP on the PDB-Select25 dataset using the method shown in the first column, sorted by the highest predictive accuracy (Q_3). In bold are the highest scores in the corresponding column.

Filtering method	Q_3 (%)	Q_H (%)	Q_E (%)	Q_L (%)	SOV	SOV_H	SOV_E	SOV_L	C_H	C_E	C_L	SEL
LibSVM	**77.53**	79.60	65.98	**82.13**	**72.29**	72.46	71.44	71.20	0.724	0.647	0.589	**74.91**
MLP	77.28	79.44	69.08	79.97	72.14	72.55	73.07	70.87	0.719	0.645	0.585	74.72
Logistic	77.04	78.21	70.20	79.81	71.82	71.13	73.76	70.99	0.713	0.643	0.584	74.43
Simple Cart	77.02	80.08	68.43	79.15	70.96	71.86	72.46	70.16	0.712	0.636	0.586	73.99
J48	76.59	79.89	67.63	78.74	69.13	70.72	71.44	69.05	0.705	0.628	0.581	72.86
RBF Network	76.43	80.34	68.90	77.29	68.63	70.85	71.51	68.55	0.705	0.619	0.581	72.53
SS-filt	76.19	73.49	73.79	79.76	70.80	67.40	76.28	70.88	0.703	0.628	0.579	73.49
No filtering	76.12	74.42	74.53	78.41	67.95	65.00	76.17	69.22	0.702	0.622	0.582	72.03
Random Forest	76.00	**81.25**	70.09	74.83	65.78	69.01	70.94	64.97	0.703	0.635	0.563	70.89
Naive Bayes	75.99	77.33	**76.70**	74.48	71.36	70.74	75.73	68.81	0.709	0.633	0.563	73.68
Viterbi	75.88	75.72	74.57	76.73	70.35	67.99	73.84	68.23	0.704	0.626	0.566	73.12
WH-filt	75.74	71.67	73.62	80.31	68.76	63.69	75.38	70.57	0.696	0.618	0.577	72.25
IBk (k=3)	73.45	80.03	66.10	71.97	61.80	67.66	68.38	60.76	0.652	0.601	0.522	67.63

In addition, we report the SOV score [20], a measure that is based on the average overlap between the observed and the predicted segments instead of the average per-residue accuracy.

Finally, we define a selection criterion, denoted as SEL, which takes equally into account the achieved Q_3 and SOV scores, the most established assessment measures. Therefore, the SEL score is calculated as follows:

$$SEL = \frac{Q_3 + SOV}{2} \quad (4)$$

2.5 Finding the best local window

A window of neighbouring residues is often used in secondary structure prediction to capture additional information about local interactions [39] and, hence, we investigate the use of a local window, w, for filtering, centred around the residue to be predicted. More specifically, the ensemble of BRNNs has three output values for each residue, one for each secondary structure state. Therefore, a local window of size w will result in an input vector of $3 \times w$ attributes for the filtering classifier. Due to different design and capabilities, the size of the local window that maximises the predictive accuracy or the SOV is different for each classifier employed in this study. Using the CB513 dataset, we tested different input coding schemes for each method to find the best local window in each case. Figure 2 shows how the predictive accuracy and the SOV measure changes by varying the size of the local window. The selected window size was the one that maximised the SEL score (Equation 4) for each machine learning technique tested, thus taking into account both the Q_3 and the SOV score. The drop of accuracy in the case of MLP for $w > 5$ is most probably due to overfitting. After optimising the local window sizes for each method on the CB513 dataset, we utlilised them for filtering PSSP on the PDB-Select25 dataset.

3 Results and Discussion

Table 1 shows the performance of each filtering method sorted by the highest accuracy, after finding the local window size, w, that maximises the predictive accuracy (Q_3) on the CB513 dataset. The SVM achieved the highest predictive accuracy of 77.04%, an absolute improvement of 0.65% compared to the unfiltered performance, while the SOV score increased by 3.8, reaching the value of 72.54. However, it is the Logistic function that achieved the highest SOV score of 72.83, an increase of 4.09, whereas its predictive accuracy of 76.93% is ranked second to the tested methods. Notably, the Logistic function has higher SEL score than the SVM, while it is also a faster classifier for this problem. In addition, the MLP and the Simple Cart also achieve improved accuracies and SOV scores higher than 70. Despite that only half of the machine learning algorithms increase their accuracy after filtering, the majority of them increase the SOV significantly.

Table 2 shows the performance of each technique on the PDB-Select25 dataset, for which the window sizes

TABLE 3
Filtering PSSP using combinations of machine learning algorithms and empirical rules. Firstly, a machine learning algorithm is employed for filtering (shown in the first column) and, subsequently, the output is filtered by empirical rules (SS-filt or WH-filt) to further improve PSSP. In bold are the highest Q_3, SOV and SEL scores.

	CB513 dataset						**PDB-Select25 dataset**					
	SS-filt			WH-filt			SS-filt			WH-filt		
Classifier	Q_3 (%)	SOV	SEL	Q_3 (%)	SOV	SEL	Q_3 (%)	SOV	SEL	Q_3 (%)	SOV	SEL
LibSVM	**77.02**	72.94	74.98	76.85	72.21	74.53	**77.50**	72.66	75.08	77.33	71.98	74.65
Logistic	76.92	**73.42**	**75.18**	76.76	72.52	74.64	77.04	72.64	74.84	76.89	71.78	74.34
MLP	76.74	72.50	74.62	76.58	71.48	74.03	77.27	**72.91**	**75.09**	77.15	72.15	74.65
Simple Cart	76.67	72.64	74.65	76.51	71.14	73.82	77.06	72.70	74.88	76.89	71.53	74.21
RBF Network	76.54	72.52	74.53	76.39	70.61	73.50	76.66	71.54	74.10	76.44	69.91	73.17
J48	76.21	71.42	73.81	75.85	69.34	72.60	76.75	71.73	74.24	76.54	70.14	73.34
Naive Bayes	76.11	72.10	74.10	75.96	71.50	73.73	76.01	71.79	73.90	75.82	71.11	73.47
Viterbi	75.98	68.59	72.29	75.88	69.15	72.52	75.88	70.35	73.12	75.81	70.01	72.91
Random Forest	75.71	71.58	73.64	75.49	69.12	72.30	76.47	71.27	73.87	76.16	68.42	72.29
IBk (k=3)	73.33	68.73	71.03	73.01	65.45	69.23	74.55	68.30	71.42	74.14	65.14	69.64

were optimised in the CB513 dataset (see Table 1). It is worth mentioning that the overall performance of the unfiltered BRNN ensemble ("No Filtering" row) is slightly lower for this larger dataset. This observation is consistent when all three overall performance measures are considered, i.e. Q_3, SOV and SEL. Nonetheless, the majority of the applied methods improve the predictive performance and a higher increase is observed compared to the results on the CB513 dataset. The SVM is the most accurate filtering method based on all three basic measures (Q_3, SOV and SEL), showing an improvement of around 1.4%, 4.4 and 2.9, respectively, while the Logistic function and the MLP are amongst the most accurate techniques. In fact, the MLP performs particularly well on this dataset and is ranked second to the tested methods based on the achieved Q_3, SOV and SEL. Apart from the MLP, the RBF network and the J48 decision tree perform better on the PDB-Select25 dataset than on the CB513 dataset.

3.1 Prediction accuracy per state

Notably, some machine learning techniques perform particularly well in the prediction of individual states. More specifically, the RBF network and the Random Forest achieve the highest per-state accuracy for helical residues on both datasets, even though their overall Q_3 score is lower than that of the best performing methods. In fact, the RBF Network increases the Q_H accuracy by 4.4% on the CB513 dataset, which is more than 2.5% higher than the Q_H score of the two best performing classifiers. On the PDB-Select25 dataset, the Random Forest achieves a remarkable improvement of the Q_H score by 6.8%.

Similarly, the Naive Bayes classifier is very accurate in the prediction of extended residues achieving Q_E of 71.99% on the CB513 dataset, which is more than 4% higher than the Q_E achieved by the Logistic function, whereas it is more that 6% higher than the Q_E of the SVM. Importantly, its achieved Q_E score of 76.7% on the PDB-Select25 dataset is around 10.7% higher than that of the SVM. In contrast, all three algorithms perform relatively poor in the prediction of loop residues, resulting in a low overall per-residue accuracy.

Interestingly, the ensemble of BRNNs overpredicts extended residues with the utilisation of the PDB-Select25 dataset (see Table 2), but the application of filtering techniques significantly affects the predictive performance. Some classifiers, such as the Naive Bayes and the Viterbi algorithm, perform well for the prediction of extended residues, while others, such as the LibSVM, decrease the Q_E score. The explanation can be derived from the size of secondary structure elements. While α-helices are usually long repetitive structures with an average length of about ten residues, most extended structures in proteins are shorter than eight residues. Therefore, using a long local window may improve the prediction of longer structures (helix and coil), but it may also decrease the predictive performance of short extended structures. In fact, the Q_E scores shown in Table 2 are usually higher for classifiers that use short local windows (from one to three residues) and lower for classifiers that use long windows, such as 19 residues. In addition, the dataset size seems to be important

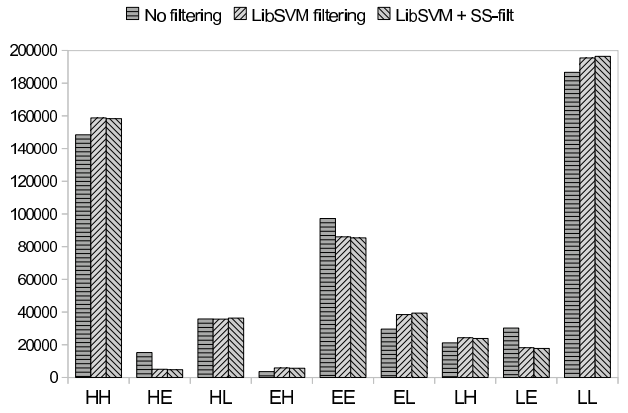

Fig. 3. Per state prediction after the application of filtering techniques on the PDB-Select25 dataset. The y-axis corresponds to the number of residues and the x-axis to the combinations of observed and predicted state. For instance, HE corresponds to residues that are observed as helices (H) but are predicted as extended (E). The three columns at each state show the number of residues for the unfiltered classifier (ensemble of BRNNs), the LibSVM filtering and the combination of LibSVM and SS-filt, respectively.

for the prediction of the extended class, which underrepresented in experimentally determined structures and this is an additional reason it is more difficult to predict. Thus, increasing the size of the dataset can provide crucial additional information about the extended state.

3.2 Combining machine learning and empirical techniques

Interestingly, the empirical filtering does not increase the Q_3 score dramatically, but it improves the SOV. The smoothing rules used in SS-filt are more effective than those in WH-filt in terms of both Q_3 and SOV. The empirical rules have better performance than some of the machine learning algorithms shown in Tables 1 and 2, but they come short when they are compared with the best performing techniques, such as the SVM, the Logistic function and the MLP.

Nevertheless, a combination of machine learning techniques and empirical rules can lead to a generally improved filtering in secondary structure prediction. As shown in Table 3, the quality of the prediction is improved when we apply empirical rules after filtering by a machine learning algorithm. The empirical rules are manually created by scientists and, hence, they provide physicochemically realistic information, which sometimes cannot be captured by a learning algorithm. On both datasets, the predictive accuracy is not improved for the best performing classifiers, but it is higher for algorithms that achieved lower accuracies without the employment of the empirical rules (compare with Tables 1 and 2), such as the

```
A  1AHB_A, 180-246
   PriStr    VPSLATISLENSWSGLSKQIQLAQGN...
   Real SS   LLLHHHHHHHHHHHHHHHHHHLLL...
   No-Filt   LLLLLEEEHLHLHLHHLHHHEHHLLL...
   LibSVM    LLLHHHEEHHHLLLHHLHHHHHLLL...
   SS-Filt   LLLHHHHHHHHLLLHHLHHHHHLLL...

B  1HIW_S, 7-109
   PriStr    ...QTGSEELRSLYNTIAVLYCVHQRIDVKDTKEA...
   Real SS   ...LLELHHHHHHHHHHHHHHHHLLLLLLEHHHH...
   No-Filt   ...LLLLHHHHHHHLHEEEEHELLLEHHHELHHHH...
   LibSVM    ...LLLLHHHHHHHHHHEEHHLLLLHHLLLHHHH...
   SS-Filt   ...LLLLHHHHHHHHHHHHHHLLLLHHLLLHHHH...

C  2DA7_A
   PriStr    ...LPQEFVKEWFEQRKVYQYSNSRSGPSSG
   Real SS   ...LLHHHHHHHHHHHHHHHLLLLLLLLLL
   No-Filt   ...LLLLEEEEHHHHHHHHHHLLLLLLLLL
   LibSVM    ...LLHHHEEHHHHHHHHHHLLLLLLLLL
   SS-Filt   ...LLHHHHHHHHHHHHHHHLLLLLLLLL

D  1MSO_B
   PriStr    FVNQHLCGSHLVEALYLVCGERGFFYTPKT
   Real SS   LLLELLLLHHHHHHHHHHHHHLEEELLLL
   No-Filt   LLLLLHLLHHHEEEEEEHLLLLLEEELLLL
   LibSVM    LLLLLHLLHHHHHHHHHHLLLLLEEELLLL
   SS-Filt   LLLLLLLLHHHHHHHHHHLLLLLEEELLLL

E  2ELN_A
   PriStr    GSSGSSGILLKCPTDGCDYSTPDKYKLQAHLKVHTALD
   Real SS   LLLLLLLLLEELLLLLLLEELLHHHHHHHHHHLLLL
   No-Filt   LLLLLLLEEELLLLLLLLLLLLEHEEEEEEEELLHLL
   LibSVM    LLLLLLLLLLLLLLLLLLLLLLHHHHHHEHHHLHLL
   SS-Filt   LLLLLLLLLLLLLLLLLLLLLLHHHHHHLHHHLLLL
```

Fig. 4. Five examples that show the effect of filtering on PSSP. The first line in each case shows the PDB ID and the Chain ID. Sequences A and B are taken from CB513 and the remaining sequences from PDB-Select25. The mispredictions are shown in shadow. "PriStr" is the amino acid sequence; "Real SS" is the observed secondary structure; "No-Filt" is the PSSP from the ensemble of BRNNs; "LibSVM is the PSSP filtered with LibSVM and "SS-Filt" is the application of the SS-Filt empirical rules on the output of LibSVM filtering. Secondary structure states are reported using the reduced three-state scheme (see Section 2.1).

RBF Network and the Random Forest. Most importantly, combinations of machine leaning techniques and empirical rules give a major boost to the SOV score leading to an improvement of 2% in most cases, demonstrating the crucial information provided by the empirical rules. On CB513, the Logistic function achieves the highest SOV score of 73.42 when we apply the SS-filt rules, while the SVM has the highest predictive accuracy. On PDB-Select25, the SVM remains the most accurate method in terms of Q_3, whilst the MLP achieves the highest SOV (72.91). In both datasets, the best performing methods have SOV score greater than 72.5. As discussed above, the SS-filt rules are more effective than the WH-filt rules and this

TABLE 4
Filtering PSSP using combinations of machine learning algorithms with different voting schemes on the CB513 dataset. The last three columns show the results after using the SS-filt empirical rules. In bold are the highest scores in the corresponding column.

Classifiers	Voting	w	Machine learning only			SS-filt		
			Q_3 (%)	SOV	SEL	Q_3 (%)	SOV	SEL
Logistic + RBF + Random Forest	Prod	3	76.71	71.74	74.23	76.74	72.76	74.75
Logistic + Simple Cart + RBF	Prod	5	76.54	70.50	73.52	76.53	72.02	74.27
Logistic + Naive Bayes	Prod	3	76.30	71.93	74.12	76.31	72.37	74.34
Logistic + Naive Bayes	Avg	3	76.28	71.92	74.10	76.28	72.37	74.33
Logistic + Simple Cart + MLP	Prod	9	77.11	72.25	74.68	77.08	73.43	75.26
Logistic + Simple Cart + MLP	Min	5	76.93	72.17	74.54	76.92	73.11	75.01
Logistic + Simple Cart + MLP	Max	11	76.95	72.16	74.55	76.93	73.22	75.07
Logistic + Simple Cart + MLP	Avg	9	**77.12**	**72.36**	**74.73**	**77.09**	**73.46**	**75.27**
Logistic + Simple Cart + MLP	Maj	11	76.90	71.75	74.33	76.89	72.62	74.76

is also observed in the results of Table 3.

Figure 3 illustrates how the number of correct predictions or mispredictions changes after filtering PSSP with LibSVM and the subsequent application of the SS-filt empirical rules on the PDB-Select25 dataset. As stated above, the ensemble of BRNNs overpredicts extended residues and this is reflected on the high percentage of correctly predicted extended residues, but also on the large number of mispredicted helical and loop residues as extended (HE and LE states). This behaviour is smoothed after filtering with LibSVM, which decreases the number of both correct predictions and mispredictions to extended state, while improving the performance of helix and loop prediction. The application of the empirical rules does not have a significant effect on the number of correctly predicted residues, but, as discussed above, it improves the SOV score. The analysis of mispredictions based on their dihedral angles, ϕ and ψ, did not reveal any particular trend since the mispredicted residues are distributed all over the Ramachandran plot (data not shown).

Figure 4 shows five examples which demonstrate the possible effect of filtering on the overall PSSP. Importantly, the use of the SVM filtering improves PSSP noticeably and in some cases, such as examples D and E, leads in a significant improvement of the predictive performance. Subsequently, the application of the SS-Filt rules is a final refinement step, which smoothes the LibSVM predictions using rules that are well-defined a priori (Section 2.3), but its effect is not always significant for the overall predictive performance. Importantly, the filtering step (LibSVM with SS-Filt) should be used as a post-processing step which will refine the results of a PSSP method. The success of any filtering technique strongly depends on the success of the initial prediction method. Filtering can be highly beneficial for state-of-the-art PSSP methods because the output of the initial method is fed as input to the filtering algorithm and, thus, this information must be as accurate as possible. This is certainly a challenging task which is outside the scope of this article, where we consider our intitial prediction method satisfactory given its results.

3.3 Combining machine learning techniques

Based on the results shown in Tables 1 and 2, we tested various combinations of machine learning techniques using different voting schemes implemented in WEKA (see Section 2.3) and the results are shown in Table 4 for the CB513 dataset. More specifically, a number of machine learning algorithms are initially used for filtering and their output is fed into a voting function, which decides for the final prediction. The voting schemes based on the average probabilities and the product of probabilities achieve the highest accuracies. The predictions are then filtered by the SS-filt empirical rules to further improve the predictive performance. Given the extensive computational needs of the SVM, we did not use it for the combinations presented in Table 4. Instead, the Logistic function was employed in combination with other machine learning techniques. The voting based on the average probability using the Logistic function, the Simple Cart and the MLP slightly improved the Q_3 and SOV scores. However, the improvement is insignificant compared to the performance of the Logistic function alone. Moreover, combining the Logistic function with classifiers that performed particularly well in the prediction of helical and extended residues, such as the RBF Network and the Naive Bayes, did not have positive impact on the overall performance. Due to the computational requirements and given

the insignificant improvement of predictive accuracy on the CB513 dataset, we did not apply the above combinations of machine learning techniques on the PDB-Select25 dataset.

4 Conclusion

The aim of this work was to compare the performance of a variety of filtering methods to the problem of PSSP, which has not been studied systematically, although it is utilised by a plethora of prediction methods. We employed both machine learning algorithms and empirical methods and, using two non-redundant, non-homologous sets of 513 and 3977 protein chains, respectively, we showed that the SVM, the Logistic function and the MLP are the most suitable learning techniques to tackle this problem. More importantly, combinations of machine learning techniques and empirical smoothing rules can improve the quality of the predictions even further, particularly the SOV score.

Based on the results presented in this article, we suggest the utilisation of the Logistic function or the MLP followed by the application of the SS-filt empirical rules to filter PSSP. Despite achieving slightly lower predictive accuracy than the SVM, these classifiers are much faster compared to the SVM and can lead to reliable filtering of the predictions.

Our findings are based on initial (sequence-to-structure) secondary structure predictions obtained by a BRNN with per-residue weight updating. Different approaches at this starting step are expected to yield different results, thus the impact of filtering methods on alternative initial data, or their combinations, should be further investigated.

We are currently conducting a similar study to evaluate learning algorithms used for ANN ensembles, instead of just averaging the outputs of a number of ANNs. Finally, since filtering is a common step in many protein structure prediction problems, such as β-turn prediction [40], this comparative study can be useful for other research fields in structural bioinformatics.

Acknowledgments

The authors would like to thank the Cyprus Research Promotion Foundation for grant TPE/ORIZO/0308(FR)/05.

References

[1] N. Qian and T. J. Sejnowski, "Predicting the secondary structure of globular proteins using neural network models." *J Mol Biol*, vol. 202, no. 4, pp. 865–884, 1988.

[2] B. Rost and C. Sander, "Prediction of protein secondary structure at better than 70% accuracy." *J Mol Biol*, vol. 232, no. 2, pp. 584–599, 1993.

[3] D. T. Jones, "Protein secondary structure prediction based on position-specific scoring matrices." *J Mol Biol*, vol. 292, no. 2, pp. 195–202, 1999.

[4] P. Baldi, S. Brunak, P. Frasconi, G. Soda, and G. Pollastri, "Exploiting the past and the future in protein secondary structure prediction." *Bioinformatics*, vol. 15, no. 11, pp. 937–946, 1999.

[5] G. Pollastri, D. Przybylski, B. Rost, and P. Baldi, "Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles." *Proteins*, vol. 47, no. 2, pp. 228–235, 2002.

[6] G. Pollastri and A. McLysaght, "Porter: a new, accurate server for protein secondary structure prediction." *Bioinformatics*, vol. 21, no. 8, pp. 1719–1720, 2005.

[7] M. J. Wood and J. D. Hirst, "Protein secondary structure prediction with dihedral angles." *Proteins*, vol. 59, no. 3, pp. 476–481, 2005.

[8] S. Hua and Z. Sun, "A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach." *J Mol Biol*, vol. 308, no. 2, pp. 397–407, 2001.

[9] G. Karypis, "YASSPP: Better kernels and coding schemes lead to improvements in protein secondary structure prediction." *Proteins*, vol. 64, no. 3, pp. 575–586, 2006.

[10] P. Kountouris and J. D. Hirst, "Prediction of backbone dihedral angles and protein secondary structure using support vector machines." *BMC Bioinformatics*, vol. 10, no. 1, p. 437, 2009.

[11] K. Karplus, C. Barrett, M. Cline, M. Diekhans, L. Grate, and R. Hughey, "Predicting protein structure using only sequence information." *Proteins*, vol. Suppl 3, pp. 121–125, 1999.

[12] K. Lin, V. A. Simossis, W. R. Taylor, and J. Heringa, "A simple and fast secondary structure prediction method using hidden neural networks." *Bioinformatics*, vol. 21, no. 2, pp. 152–159, 2005.

[13] X. M. Pan, "Multiple linear regression for protein secondary structure prediction." *Proteins*, vol. 43, no. 3, pp. 256–259, 2001.

[14] S. Qin, Y. He, and X.-M. Pan, "Predicting protein secondary structure and solvent accessibility with an improved multiple linear regression method." *Proteins*, vol. 61, no. 3, pp. 473–480, 2005.

[15] J. R. Green, M. J. Korenberg, and M. O. Aboul-Magd, "PCI-SS: MISO dynamic nonlinear protein secondary structure prediction." *BMC Bioinformatics*, vol. 10, p. 222, 2009.

[16] J. A. Cuff, M. E. Clamp, A. S. Siddiqui, M. Finlay, and G. J. Barton, "JPred: a consensus secondary structure prediction server." *Bioinformatics*, vol. 14, no. 10, pp. 892–893, 1998.

[17] S. Montgomerie, S. Sundararaj, W. J. Gallin, and D. S. Wishart, "Improving the accuracy of protein secondary structure prediction using structural alignment." *BMC Bioinformatics*, vol. 7, p. 301, 2006.

[18] K. P. Wu, H. N. Lin, J. M. Chang, T. Y. Sung, and W. L. Hsu, "HYPROSP: a hybrid protein secondary structure prediction algorithm–a knowledge-based approach." *Nucleic Acids Res*, vol. 32, no. 17, pp. 5059–5065, 2004.

[19] C. Mooney and G. Pollastri, "Beyond the twilight zone: automated prediction of structural properties of proteins by recursive neural networks and remote homology information." *Proteins*, vol. 77, no. 1, pp. 181–190, 2009.

[20] A. Zemla, C. Venclovas, K. Fidelis, and B. Rost, "A modified definition of sov, a segment-based measure for protein secondary structure prediction assessment." *Proteins*, vol. 34, no. 2, pp. 220–223, 1999.

[21] H. Zhang, T. Zhang, K. Chen, K. D. Kedarisetti, M. J. Mizianty, Q. Bao, W. Stach, and L. Kurgan, "Critical assessment of high-throughput standalone methods for secondary structure prediction." *Brief Bioinform*, 2011. [Online]. Available: http://dx.doi.org/10.1093/bib/bbq088

[22] J. Chen and N. Chaudhari, "Cascaded bidirectional recurrent neural networks for protein secondary structure prediction." *IEEE/ACM Trans Comput Biol Bioinform*, vol. 4, no. 4, pp. 572–582, 2007.

[23] A. A. Salamov and V. V. Solovyev, "Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignments." *J Mol Biol*, vol. 247, no. 1, pp. 11–15, 1995.

[24] M. Agathocleous, G. Christodoulou, V. Promponas, C. Christodoulou, V. Vassiliades, and A. Antoniou, "Protein secondary structure prediction with bidirectional recurrent neural nets: Can weight updating for each residue enhance performance?" in *Artificial Intelligence Applications*

and Innovations, ser. IFIP Advances in Information and Communication Technology, H. Papadopoulos, A. Andreou, and M. Bramer, Eds. Springer Boston, vol. 339, pp. 128–137, 2010.
[25] J. A. Cuff and G. J. Barton, "Evaluation and improvement of multiple sequence methods for protein secondary structure prediction." *Proteins*, vol. 34, no. 4, pp. 508–519, 1999.
[26] U. Hobohm, M. Scharf, R. Schneider, and C. Sander, "Selection of representative protein data sets." *Protein Sci*, vol. 1, no. 3, pp. 409–417, 1992.
[27] W. Kabsch and C. Sander, "Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features." *Biopolymers*, vol. 22, no. 12, pp. 2577–2637, 1983.
[28] B. Rost and V. A. Eyrich, "EVA: large-scale analysis of secondary structure prediction." *Proteins*, vol. 5, pp. 192–199, 2001.
[29] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs." *Nucleic Acids Res*, vol. 25, no. 17, pp. 3389–3402, 1997.
[30] S. Henikoff and J. G. Henikoff, "Amino acid substitution matrices from protein blocks." *Proc Natl Acad Sci USA*, vol. 89, no. 22, pp. 10 915–10 919, 1992.
[31] D. T. Jones and M. B. Swindells, "Getting the most from PSI-BLAST." *Trends Biochem Sci*, vol. 27, no. 3, pp. 161–164, 2002.
[32] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*. Morgan Kaufmann, 2005.
[33] S. le Cessie and J. van Houwelingen, "Ridge estimators in logistic regression," *Applied Statistics*, vol. 41, no. 1, pp. 191–201, 1992.
[34] C. C. Chang and C. J. Lin, *LIBSVM: a library for support vector machines*, 2001, Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm. Accessed in Oct 2011.
[35] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm" *IEEE Trans Inf Theory*, vol. 13, no. 2, pp. 260 – 269, 1967.
[36] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data mining," *Knowl Inf Syst*, vol. 14, pp. 1–37, 2007.
[37] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, "SCOP: a structural classification of proteins database for the investigation of sequences and structures." *J Mol Biol*, vol. 247, pp. 536–540, 1995.
[38] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme." *Biochim Biophys Acta*, vol. 405, no. 2, pp. 442–451, 1975.
[39] B. Rost, "Review: protein secondary structure prediction continues to rise." *J Struct Biol*, vol. 134, no. 2-3, pp. 204–218, 2001.
[40] A. J. Shepherd, D. Gorse, and J. M. Thornton, "Prediction of the location and type of β-turns in proteins using neural networks." *Protein Sci*, vol. 8, no. 5, pp. 1045–1055, 1999.

Petros Kountouris initially obtained a B.Sc. in Computer Engineering and Informatics from the University of Patras, Greece, in 2006. He later moved to the UK where he received a Ph.D. in Chemistry/Bioinformatics from the University of Nottingham in 2010. He is currently working as a post-doctoral researcher at the Department of Computer Science in the University of Cyprus, Cyprus. He has continuous interest for structural computational biology and on possible applications of computational intelligence and machine learning to bioinformatics.

Michalis Agathocleous received a B.Sc. degree (with distinction) in Computer Science from the University of Cyprus, Nicosia, Cyprus, in 2009, and a M.Sc. degree in Machine Learning, from the University College London (UCL), London, U.K., in 2010. He is currently pursuing a Ph.D. degree at the Department of Computer Science, University of Cyprus. His current research interests include bioinformatics, neuroscience, machine learning and computational intelligence.

Vasilis J. Promponas holds a B.Sc. in Physics and a Ph.D. in Biological Sciences both from the University of Athens, Greece. He is currently heading the Bioinformatics Research Laboratory in the Department of Biological Sciences at the University of Cyprus, where he is a Lecturer in Bioinformatics. His research interests revolve around developing and applying computational tools, statistical and machine learning methods for protein sequence analysis, prediction of protein structure and function, comparative genomics, biodiversity informatics, genome and protein evolution.

Georgia Christodoulou received a B.Sc. degree in Computer Science from the University of Cyprus, Nicosia, Cyprus, in 2010, and a M.Sc. degree in Bioinformatics and Theoretical Systems Biology from Imperial College, London, U.K., in 2011. Her current research interests include computational biology and bioinformatics.

Simos Hadjicostas received a B.Sc. degree in Computer Science from the University of Cyprus, Nicosia, Cyprus, in 2011. His current research interests include bioinformatics and computational intelligence.

Vassilis Vassiliades received a B.Sc. degree in Computer Science from the University of Cyprus, Nicosia, Cyprus, in 2007, and a M.Sc. degree (with distinction) in Intelligent Systems Engineering, from the University of Birmingham, Birmingham, U.K., in 2008. He is currently pursuing a Ph.D. degree at the Department of Computer Science, University of Cyprus. His current research interests include reinforcement learning, neuroevolution, multiagent systems, and computational intelligence.

Chris Christodoulou received a B.Eng. degree in Electronic Engineering from Queen Mary and Westfield College, University of London, London, U.K., and a Ph.D. degree in Neural Networks/Computational Neuroscience from Kings College, University of London. He also holds a B.A. degree in German from Birkbeck College, University of London. He was a Postgraduate Research Assistant from 1991 to 1995 and a Post-Doctoral Research Associate from 1995 to 1997 at the Centre for Neural Networks, Kings College, University of London. He joined Birkbeck College as a Lecturer in 1997, where he worked until 2005, and was also a Visiting Research Fellow at Kings College from 1997 to 2001. He is currently an Associate Professor at the University of Cyprus after joining in 2005. Since 2005, he is also a Visiting Research Fellow at Birkbeck College. His current research interests include computational and cognitive neuroscience, and neural networks.