KB8024/8025

Project in Molecular Life Sciences

MSc Molecular Techniques in Life Science

Stockholms Universitet - VT17

Report

Protein Secondary Structure Predictor based on DSSP 3-State Classifications

Dimitri A. Wirjowerdojo

Introduction

The benefits of understanding how a protein functions are extensive, especially in the field of drug development and disease therapies [1]. However, this is prohibited due to our still limited ability of predicting its structure. This issue is commonly broken down into smaller challenges, such as predicting its secondary structure, which refers to the local conformation of the polypeptide backbone [2]. Kabsch and Sanders develop a coding system for classifying secondary structures of proteins into eight classes, including α-helix (H), 3_{10} helix (G), π-helix (I), isolated β-bridge (B), extended strand (E), hydrogen-bonded turn (T), bends (S) and blanks or coils (C), on the basis of hydrogen bonds and atomic-resolution coordinates of a protein [3].

The first secondary structure prediction was done through the analysis of hydrogen-bonding patterns by Pauling and Corey in 1951 [4]. Since then, the field of secondary structure prediction have grown to incorporate more intricate approaches and methods. Examples include using evolutionary information with PSI-BLAST and neural networks in the case of PSIPRED [5] and PROTEUS which employs a homologous template-based prediction alongside conventional – sequence-based – prediction [6].

The goal of this project was to develop a secondary structure predictor using a relatively small amount of training dataset. The use of Support Vector Machines and Decision Tree and Random Forest classifiers as supervised machine learning methods were investigated across multiple window size. It was found that with the available dataset, the Random Forest classifier performed the best in terms of predicting helices and coils and Linear Support Vector Classifier (LinearSVC) for sheets.

Methods

For the development of this predictor, Python was used as the programming language. The computer on which this predictor was developed ran on Ubuntu 16.04.02 LTS, with Intel Core i7 Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz processor and 32 GB of RAM. The scripts for the predictors and the datasets can be found on http://www.github.com/dmtr13/KB8024/tree/master/Final.

Datasets

Train Set of 399 proteins (containing 115,271 residues) was acquired from Elofsson Lab (bioinfo.se). It contained, every three lines, the protein ID, sequence and secondary structure information for each residue. The secondary structure information was based on DSSP 3-class classifications [3]: H for helices, C for coils and S for sheets (later defined as E). Calculations of the H, E and C content for every protein give an average of 36.22 ± 19.69%, 21.15 ± 14.55% and 42.63 ± 10.17%, respectively. No further processing was done on the training set.

Test Set of 50 proteins (containing 11,370 residues) was obtained in steps. Firstly, using PISCES web server [7], a list of PDB entries was obtained from the whole PDB, with culling threshold of less than 5% sequence identity, resolution up to 1.6Å, R-factor lower than 0.3 and sequence length between 40 – 10,000 residues. The low sequence identity threshold was meant for a broader variety of test proteins. Then it was sorted based on resolution (ascending), the ratio of Free R-value to the R-factor (closest to 1) and finally on the R-factor (ascending). Secondly, using the *mkdssp* programme v2.2.7 [3,8], the DSSP 8-class classification was generated. However, it was noticed that for some sequences, there are gaps in the residues, as such they were excluded from the third step. Then, after converting it into the 3-class classification (H, G and I to H; E and B to E; and S, T and blanks to C), the H, E and C content were calculated for each protein. It was found that the majority of the sequences contained mostly (over 80%) E structures. To overcome this, the entire process was repeated with culling threshold of less than 20% sequence identity, resolution up to 1.6Å and R-factor less than 0.25, however, the issue persisted. A more balanced trainset was finally obtained when the culling threshold was less than 20% sequence identity, resolution up to 2.0Å and R-factor of less than 0.25. The H, E and C content were 32.90 ± 11.61%, 45.88 ± 10.71% and 21.22 ± 4.93%, respectively.

Encoding of Residues

For the single-sequence based predictor, each residue of each protein was first mapped into integers, ranging from 1 to 20. One map with the value of 0 was also created for padding, in order to incorporate the residues at the beginning and the ending of each protein. Using the OneHotEncoder pre-processing function from the scikit-learn module [9], the residues were converted into sparse matrix depending on the chosen window size. Conversely, for the predictor that incorporates evolutionary information (defined as PSSM-based (position-specific scoring matrix) predictor in this report), the residues were not mapped nor pre-processed with OneHotEncoder as they were already numerical. Padding was also incorporated but as a list of twenty-zeros. Additionally, they were normalised prior to being used for model-generation: for the substitution matrix, the standard logistic function, as per Equation 1 below, was applied; whereas for the frequency matrix, each number was divided by 100.

$$x_{Normalised} = \frac{1}{1 + e^{-x}} \quad \text{(Eq. 1)}$$

Incorporation of Evolutionary Information

To improve the prediction, evolutionary information was incorporated by using the position-specific scoring matrix generated through PSI-BLAST. Using the BLAST+ package v2.2.31 each protein of the training set was subjected to PSI-BLAST against the Uniref90 database. Whilst a more comprehensive, non-redundant database could have been used, it contained

automatically curated data, such as those from GenBank, which may reduce the accuracy of the PSI-BLAST itself. The parameters were based on one of the fundamental predictor for secondary structure, PSIPRED: E-value threshold of 10^{-2} [10] and number of iterations of 3 [5].

Window Size Selection and Classifier Optimisation

There are multiple supervised learning classifiers available through scikit-learn [9], for the predictors reported in this paper, four were used: SVC (support vector classification), LinearSVC, Decision Tree and Random Forest. Comparison of the four classifiers were done on default settings and based on the average of 5-fold cross-validation scores and 50% of dataset for the single sequence-based predictor or 25% of dataset for the PSSM-based predictors, due to time constraints.

The range of window sizes was based on the literature review conducted by Chen et al.: 7, 9, 11, 13, 15 and 17. It was mentioned a window size of 7 was commonly chosen for secondary structure prediction [11]. However, throughout the development of the predictors, multiple window sizes were investigated.

LinearSVC and SVC differ on the basis of the library used. According to the user guide, LinearSVC scales better for large numbers of samples, whereas SVC can potentially become problematic with more than 10,000 samples [9]. This was compared to SVC with different kernels: linear, radial basis function (RBF), polynomial and sigmoid, as shown in Figure 1 for the single sequence-based predictor; similar graph for the other predictors can be found in Appendix 1.

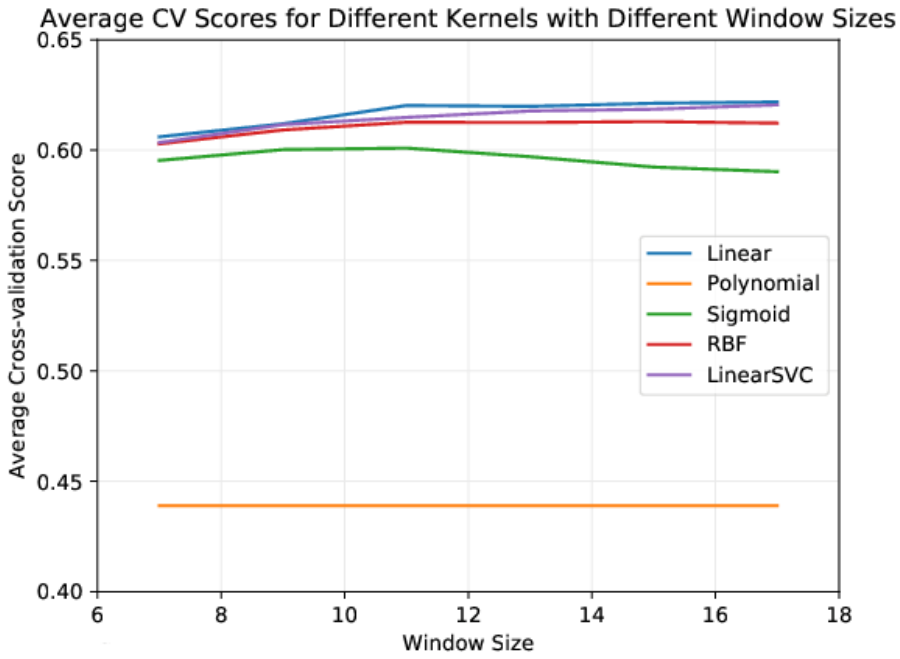


Fig. 1 Single sequence-based predictor 5-fold cross-validation average for SVC with different kernels and LinearSVC across window size of 7-17. Cross-validation was done on 50% dataset.

In terms of performance, SVC with kernel linear and LinearSVC were relatively comparable, though the processing time of SVC was significantly much longer due to the decision function shape being one-versus-one. Subsequently, this was compared to the Decision Tree and Random Forest classifiers, with unbalanced and balanced class weights, as shown in Figure 2 for the single sequence-based predictor; the graphs for the other predictors can be found in Appendix 2.

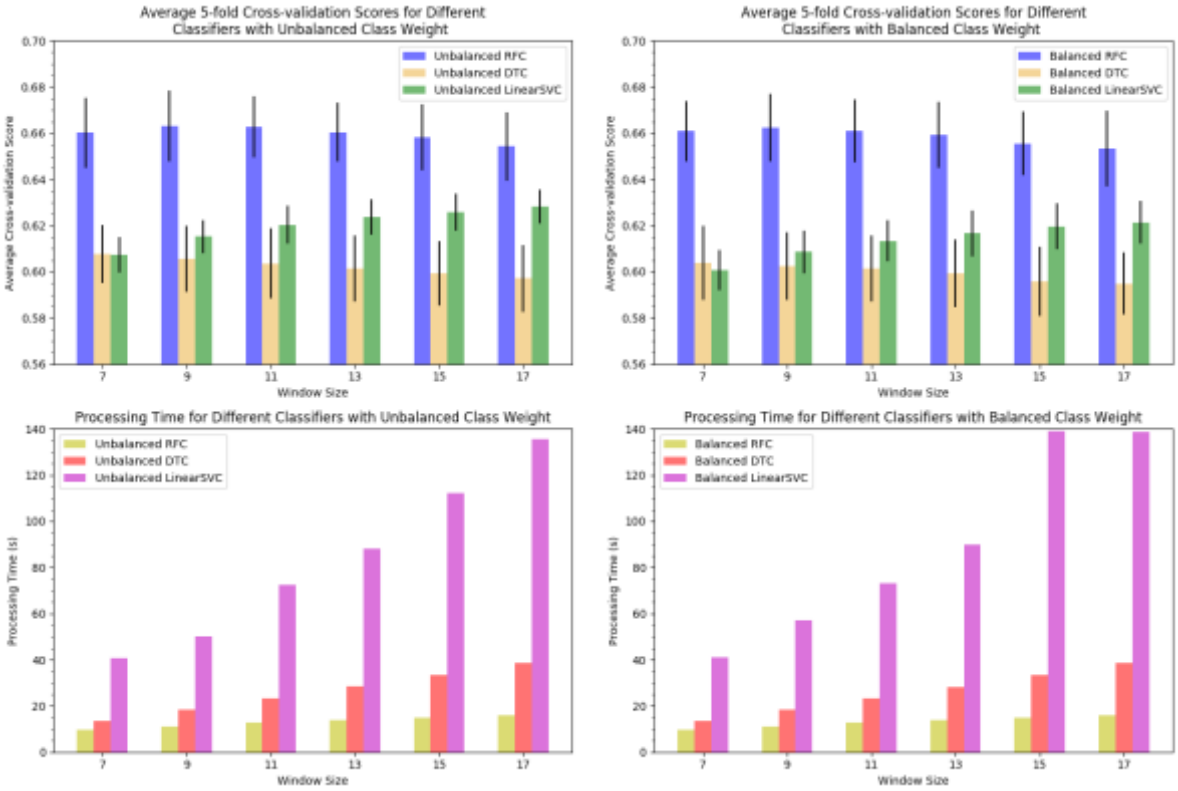


Fig. 2 Single sequence-based predictor 5-fold cross-validation average for LinearSVC, Decision Tree Classifier (DTC) and Random Forest Classifier (RFC) across window size of 7-17 and either unbalanced or balanced class weight. Cross-validation was done on 100% dataset.

As can be seen on Figure 2, the best 5-fold cross-validation score was obtained when the window size was 9 with the Random Forest classifier. Subsequent optimisation was then conducted using Random Forest classifier with window sizes of 7, 9 and 11. Furthermore, noting that the training set had an unbalanced content of H, E and C structures, the class weight was then set to 'balanced'.

To determine what parameter and window size which should be used with the Random Forest classifier, a grid search was conducted with a list of number of estimators: 10, 100, 128, 250 and 300, on the basis of weighted accuracy score and to some extent, the processing time. For the single sequence-based predictor (SSBP), the highest accuracy of 70.545 ± 2.616% was obtained with window size of 11 and number of estimators was 300.

With the PSSM-based predictors: for the frequency matrix-based predictor (FMBP), the highest accuracy was 80.215 ± 1.692% (window size 11, number of estimators 250). For the substitution matrix-based predictor (SMBP), the best accuracy score of 80.019 ± 1.809% was obtained with the window size of 11 and number of estimators was 250.

Evaluation Metrics

The metrics used to evaluate the predicted structures were Q_3 and Q_j, that predicts the overall accuracy and the accuracy for each class over total residues respectively, and SOV (segment overlap) scores for each class as per Zemla's programme [12]. Matthew's Correlation Coefficient [13,14] for each class was also calculated as per Equation 2 below.

$$MCC_j = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN\ )}} \quad \text{(Eq. 2)}$$

Results and Discussion

Performance of Classifiers

The test set of 50 proteins were subjected to all three predictors using three different kernels. For the model generated using LinearSVC and Decision Tree Classifier, the settings were left to default, whereas for the Random Forest Classifier, the parameter used was as described in the Methods section.

The accuracy metrics for LinearSVC and Decision Tree classifier are presented in Table 1. In general, with LinearSVC, in terms of Q3, SOV and per class MCC score, the substitution matrix/based predictor performed the best. Helical structures were best predicted with the SMBP, as was the case for sheets. On the other hand, coils were more accurately predicted using FMBP, close to 3.5% difference to SMBP. The SOV and MCC scores were also very comparable between FMBP and SMBP.

Using the Decision Tree classifier gave relatively the same trends. Though it was interesting to note that the overall performance dropped rather significantly when compared to LinearSVC. For the Q3 scores, the drops are about 12%, 14% and 14% for SSBP, FMBP and SMBP, respectively. For per class Q scores, it dropped by about one-third, as with the per class SOV scores. The overall SOV scores dropped by almost one-half and so did the per class MCC scores. The worst drop in MCC score occurred with the SSBP: over 4-fold decrease when compared to LinearSVC.

Decision tree works by splitting a set into the number of possible values of the features contained. It is followed by a checking step where each subset is examined for the number of classes present, if there is only a single class present then the splitting ends at that node, otherwise it is recursively split until each subset contains only a single class [15]. The overall

aim of decision tree is to generate a set of if-then rules such that the label of a feature can be decided [16]. On the other hand, LinearSVC aims to separate two different classes using the hyperplane by a large margin. The underlying method of the separation is the 'one-vs-rest', which means that a binary classifier is constructed for each class [17]. It has also been argued that support vector machines always attempts to find a generalised solution and avoids over-fitting, even with large number of features [18]. Therefore, it is possible that in the case of the results for Decision Tree classifier and LinearSVC as discussed previously, the Decision Tree classifier underfits the data due to inadequate number of training features or unoptimised parameters. Whereas for LinearSVC, the results could have possibly been improved by optimising the relatively few parameters that it has [17].

Table 1 Evaluation Metrics for LinearSVC and Decision Tree Classifier

	Predictor Basis	Observed *j*	Q_j (%)	Q_3 (%)	SOV_j (%)	SOV (%)	MCC_j
LinearSVC	*Single-sequence*	H	61.126 ± 20.302	62.122 ± 8.362	57.672 ± 22.243	56.470 ± 12.642	0.312 ± 0.233
		E	49.336 ± 20.166		54.008 ± 21.519		0.354 ± 0.197
		C	69.118 ± 11.849		57.556 ± 13.424		0.345 ± 0.162
	Frequency-matrix	H	70.310 ± 20.557	70.876 ± 8.161	70.928 ± 22.226	67.578 ± 11.994	0.431 ± 0.206
		E	57.448 ± 18.635		62.796 ± 20.234		0.555 ± 0.199
		C	77.508 ± 9.736		67.030 ± 11.840		0.480 ± 0.145
	Substitution-matrix	H	73.306 ± 18.046	71.032 ± 9.326	73.502 ± 17.973	68.638 ± 12.391	0.459 ± 0.216
		E	61.004 ± 19.406		65.600 ± 21.149		0.575 ± 0.183
		C	74.100 ± 10.488		66.452 ± 13.314		0.484 ± 0.168
Decision Tree	*Single-sequence*	H	50.976 ± 9.758	50.250 ± 6.181	24.618 ± 8.085	29.520 ± 5.405	0.079 ± 0.133
		E	36.358 ± 18.287		34.944 ± 19.375		0.127 ± 0.127
		C	57.324 ± 7.655		39.186 ± 11.561		0.129 ± 0.130
	Frequency-matrix	H	56.876 ± 12.651	56.752 ± 7.806	30.046 ± 10.621	34.248 ± 7.641	0.250 ± 0.193
		E	46.550 ± 19.266		41.084 ± 19.979		0.273 ± 0.160
		C	62.650 ± 8.051		41.90 ± 11.11		0.225 ± 0.140
	Substitution-matrix	H	59.404 ± 12.236	57.544 ± 7.386	33.114 ± 11.498	35.292 ± 8.057	0.264 ± 0.174
		E	46.136 ± 19.723		41.500 ± 21.196		0.307 ± 0.150
		C	63.776 ± 7.809		42.468 ± 11.692		0.240 ± 0.135

In comparison, the result for the Random Forest Classifier (as presented in Table 2), is generally better than the other two classifiers. Overall, the Q3 scores for all types of predictors are significantly better than those of Decision Tree classifier, and slightly better than those of LinearSVC. This is also reflected in the MCC, with around 15% improvement from LinearSVC. However, for all types of predictors, sheets are better predicted by LinearSVC as indicated by the QE and SOVE scores.

Random forest classifier works by fitting a number of decision tree classifiers on multiple subset of the training file recursively. It attempts to prevent over-fitting and increase the

accuracy of the prediction by using the averaging method and redrawing the subsets for each iteration [9]. With this classifier, the optimised number of estimators for each predictor was used (300 for SSBP; 250 for both FMBP and SMBP). Indeed, it is surprising that the results of Random Forest classifier with the non-default number of estimators is relatively comparable to that of LinearSVC on default settings. Ultimately, Random Forest classifier was chosen due to its significantly quick processing time.

Table 2 Evaluation Metrics for Random Forest Classifier

	Predictor Basis	Observed *j*	Q_j (%)	Q_3 (%)	SOV_j (%)	SOV (%)	MCC_j
Random Forest	*Single-sequence*	H	65.376 ± 17.867	63.414 ± 8.605	59.454 ± 20.457	55.184 ± 12.840	0.309 ± 0.205
		E	33.846 ± 21.981		39.626 ± 23.374		0.361 ± 0.192
		C	76.914 ± 9.243		59.844 ± 13.672		0.335 ± 0.168
	Frequency-matrix	H	70.322 ± 17.196	74.602 ± 9.763	71.29 ± 21.27	67.774 ± 15.514	0.510 ± 0.213
		E	48.814 ± 22.875		57.016 ± 24.867		0.654 ± 0.159
		C	91.658 ± 6.283		71.724 ± 13.476		0.517 ± 0.146
	Substitution-matrix	H	70.634 ± 16.524	74.362 ± 9.011	70.774 ± 21.396	67.076 ± 14.440	0.488 ± 0.209
		E	47.362 ± 22.967		55.542 ± 24.794		0.648 ± 0.159
		C	91.108 ± 5.818		70.688 ± 12.412		0.509 ± 0.135

Comparison to Other Methods

There are several secondary structure predictors currently available as webservers, such as PROTEUS, PSIPRED and FLOPRED. PROTEUS works in a concept similar to threading, where a homologous protein (of over 25% sequence identity) is used as the template to predict part of an input protein secondary structure. A sequence-based approach is also done prior to taking the consensus between the two approaches [6].

FLOPRED works in a slightly similar manner: it combines structural information from the CATH database with knowledge-based potentials to predict the secondary structure of an input sequence. The prediction is made using a neural network-based extreme learning machine and advanced particle swarm optimisation which the author argued provide comparable results to other predictors but more efficiently in terms of cost and time [19].

PSIPRED, one of the most top-performing and most established predictor, employs evolutionary information from the third iteration of PSI-BLAST with an E-value threshold of 10^{-2} to predict protein secondary structure. PSIPRED utllises a standard feed-forward back-propagation neural network architecture with a window size of 15 [5].

The overall evaluation metrics of the developed predictors are compared to those of established predictors as shown in Table 3. It can be seen that the Q3 scores for the FMBP and SMBP are relatively close (~4% difference) to that of PSIPRED. This can possibly be contributed by the fact that the procedure of prediction, up to PSSM processing, is the same. The neural network employed by PSIPRED may be the explanation to why its SOV score is

higher than those of FMBP and SMBP. PROTEUS and FLOPRED performed significantly much better, inherently because both methods utilise the known structural information of a protein(s). The single sequence-based predictor fares the worst in this comparison as it does not employ anything more than the sequence itself for training.

Table 3 Comparison of evaluation metrics to those of available predictors.

Predictor	Q_3	SOV	Ref.
SSBP	63.414 ± 8.605%	55.184 ± 12.840%	-
FMBP	74.602 ± 9.763%	67.774 ± 15.514%	-
SMBP	74.362 ± 9.011%	67.076 ± 14.440%	-
PROTEUS	Up to 88%	Up to 90%	[6]
PSIPRED	Up to 78.3%	78.7%	[5]
FLOPRED	Up to 87%	78%	[19]

Conclusion and Future Work

The development of this predictor entails the investigation of different classifiers with different kernels. On the basis of 5-fold cross validation scores, LinearSVC was the best out of SVC with various kernel types. The same metric, in addition to processing time as another factor, were compared with those of Decision Tree classifier and Random Forest classifier. With Random Forest classifier being the best, its performance was optimised to implement a balanced class weight due to an unbalanced H, E and C content in the training set. Additionally, the number of estimators was also increased to 300 for the single sequence-based predictor and 250 for the substitution matrix- and frequency matrix-based predictors. Ultimately, the optimised Random Forest classifier performed the best in terms of predicting helices and coils and Linear Support Vector Classifier (LinearSVC) for sheets.

Comparing the overall 3-class evaluation metrics to those of the available predictors, the developed predictors can be said to have a relatively comparable Q_3 score to that of PSIPRED and about 10% less for the SOV score.

To improve the predictor performance, some possible steps that can be considered include homology reduction of the training set, making a consensus out of different predictors with different kernels and implementing parameters associated with the properties of each structural class

List of References

1. Wang Y, Mao H, Yi Z. Protein Secondary Structure Prediction by using Deep Learning Method. Knowledge-Based Syst. 2016;118:115–23.
2. Wang S, Peng J, Ma J, Xu J. Protein Secondary Structure Prediction Using Deep Convolutional Neural Fields. Sci Rep. 2016;6(January):18962.
3. Kabsch W, Sander C. Dictionary of protein secondary structure: Pattern recognition of hydrogen bonded and geometrical features. Biopolymers. 1983;22:2577–637.
4. Yang Y, Gao J, Wang J, Heffernan R, Hanson J, Paliwal K, Zhou Y. Sixty-five years of the long march in protein secondary structure prediction: the final stretch? Brief Bioinform. 2016;(November):bbw129.
5. Jones DT. Protein secondary structure prediction based on position-specific scoring matrices. J Mol Biol. 1999;292:195–202.
6. Montgomerie S, Sundararaj S, Gallin WJ, Wishart DS. Improving the accuracy of protein secondary structure prediction using structural alignment. BMC Bioinformatics. 2006;7:301.
7. Wang G, Dunbrack RL. PISCES: Recent improvements to a PDB sequence culling server. Nucleic Acids Res. 2005;33:94–8.
8. Touw WG, Baakman C, Black J, Te Beek TAH, Krieger E, Joosten RP, Vriend G. A series of PDB-related databanks for everyday needs. Nucleic Acids Res. 2015;43(D1):D364–8.
9. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay É. Scikit-learn: Machine Learning in Python. J Mach Learn Res. 2012;12:2825–30.
10. UCL Bioinformatics Group. UCL-CS Bioinformatics: PSIPRED Help [Internet]. [cited 2017 Mar 3]. Available from: http://bioinf.cs.ucl.ac.uk/index.php?id=5399
11. Chen K, Kurgan L, Ruan J. Optimization of the Sliding Window Size for Protein Structure Prediction. In: 2006 IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology. 2006. p. 1–7.
12. Fidelis K, Rost B, Zemla A. A Modified Definition of Sov, a Segment-Based Measure. PROTEINS Struct Funct Genet. 1999;34(2):220–3.
13. Rashid S, Saraswathi S, Kloczkowski A, Sundaram S, Kolinski A. Protein secondary

structure prediction using a small training set (compact model) combined with a Complex-valued neural network approach. BMC Bioinformatics. 2016;(17):1–18.

14. Jurman G, Riccadonna S, Furlanello C. A comparison of MCC and CEN error measures in multi-class prediction. PLoS One. 2012;7(8):1–8.

15. Selbig J, Mevissen T, Lengauer T. Decision tree-based formation of consensus protein secondary structure prediction. Bioinformatics. 1999 Dec;15(12):1039–46.

16. Nguyen MN, Zurada JM, Rajapakse JC. Extracting decision rules in prediction of protein secondary structure. In: 2008 8th IEEE International Conference on BioInformatics and BioEngineering. IEEE; 2008. p. 1–6.

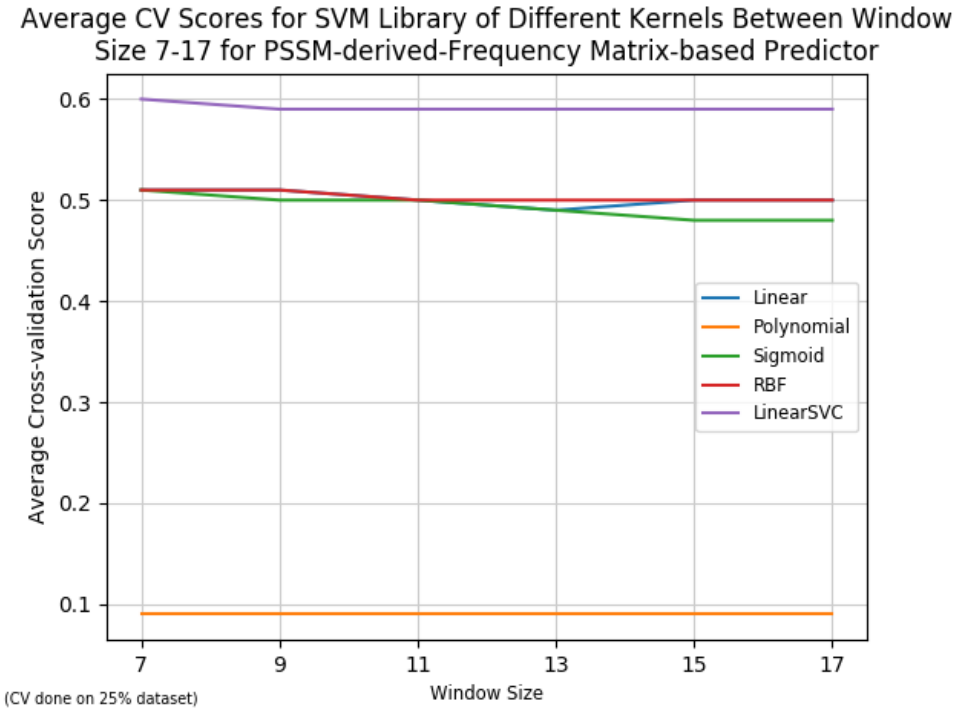
17. Ward JJ, McGuffin LJ, Buxton BF, Jones DT. Secondary structure prediction with support vector machines. Bioinformatics. 2003 Sep 1;19(13):1650–5.

18. Cai Y-D, Liu X-J, Xu X, Zhou G-P. Support Vector Machines for predicting protein structural class. BMC Bioinformatics. 2001;2(1):1.

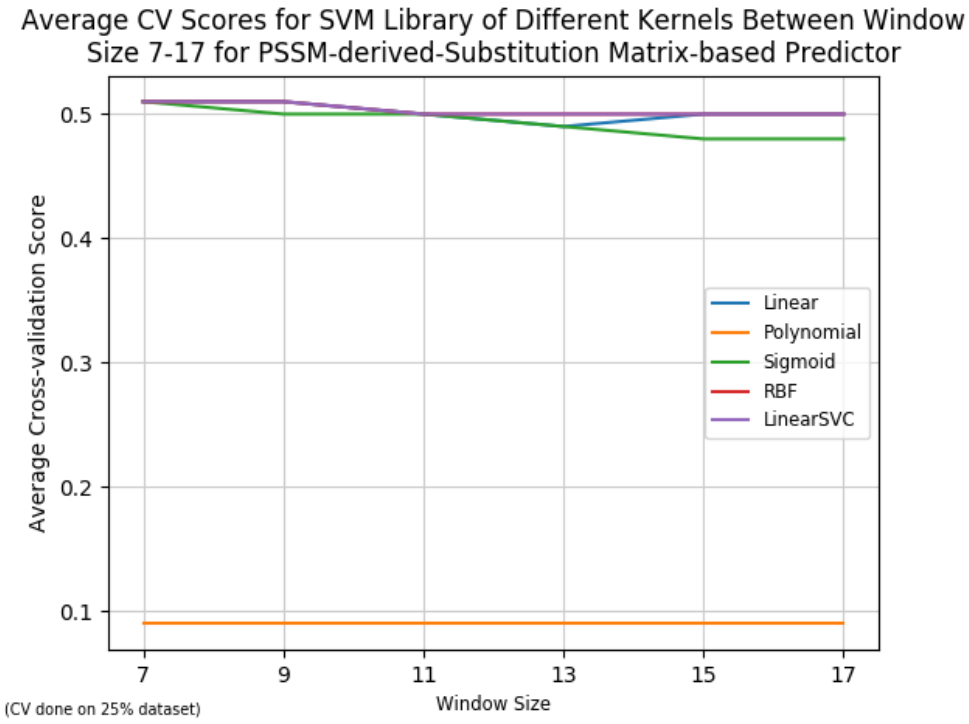
19. Saraswathi S, Fernández-Martínez JL, Kolinski A, Jernigan RL, Kloczkowski A. Fast learning optimized prediction methodology (FLOPRED) for protein secondary structure prediction. J Mol Model. 2012 Sep;18(9):4275–89.

Appendix 1

Comparison between LinearSVC and the Different Kernels of SVC



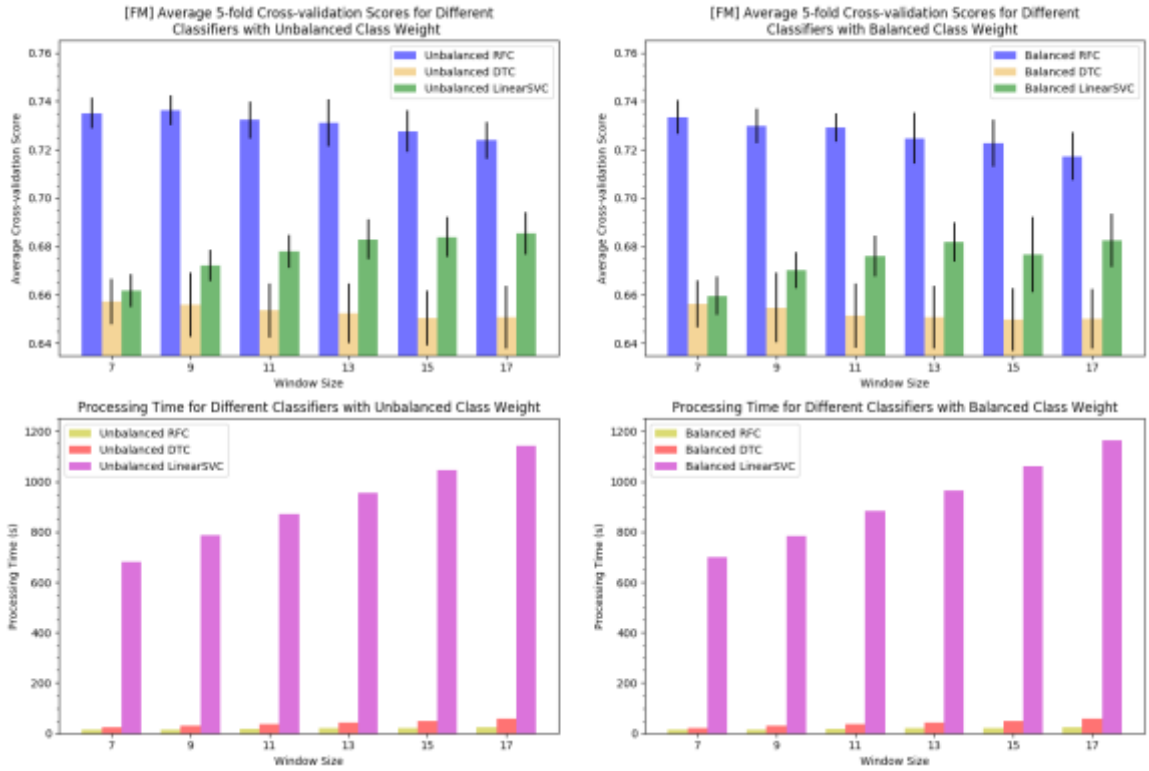
Average 5-fold cross validation score for frequency matrix-based predictor.



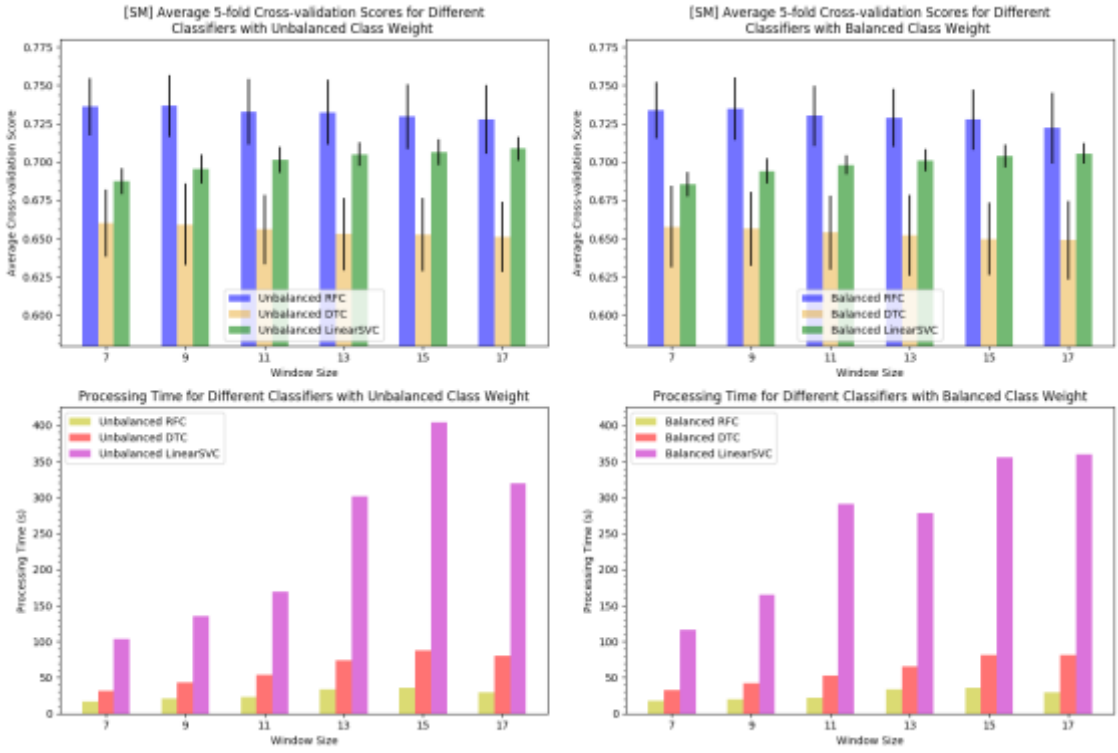
Average 5-fold cross validation score for substitution matrix-based predictor.

Appendix 2

Comparison between LinearSVC, Decision Tree Classifier and Random Forest Classifier



Average 5-fold cross validation score for frequency matrix-based predictor.



Average 5-fold cross validation score for frequency matrix-based predictor.