Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Application Note

OXFORD

Subject Section

SCLpred-Mem: membrane protein subcellular localization prediction by Deep N-to-1 Convolutional Neural Networks

Liam Ellinger [1,3,4], Manaz Kaleel [1,2], Clodagh Lalor [1], Gianluca Pollastri [1,2] and Catherine Mooney [1,3]*

[1] School of Computer Science, University College Dublin, Dublin, Ireland
[2] UCD Institute for Discovery, University College Dublin, Dublin, Ireland
[3] College of Global Studies, Arcadia University, Glenside, Philadelphia, USA
[4] Whitacre College of Engineering, Texas Tech University, Lubbock, Texas, USA

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Summary: The subcellular location of a protein can provide useful information for protein function prediction and drug design. Experimentally determining the subcellular location of a protein is an expensive and time-consuming task. Various computer-based tools have been developed, mostly using machine learning algorithms, to predict the subcellular location of proteins. We introduce SCLpred-Mem: a subcellular localization predictor powered by an ensemble of Deep N-to-1 Convolutional Neural Networks. SCLpred-Mem will work in conjunction with another predictor focused on endomembrane system prediction, and will predict if a protein will be localized to a membrane after being sent to the endomembrane system. SCLpred-Mem was trained via 5-fold cross-validation after previous testing to determine the best way to configure the network. The final predictor uses an ensemble of different trained models to increase general accuracy on any new data submitted by potential users. SCLpred-Mem predicts the subcellular location of a protein into two classes: membranes versus all others. With a Matthews correlation coefficient of 0.52 on our strict independent test set of 118 proteins added to UniProt since 2016 and redundancy reduced with respect to the training set, and MCC of 0.59 for our more relaxed ITS with 240 proteins, SCLpred-Mem outperforms many of the other state-of-the-art web servers we tested with this test set and produces comparable results to the leading predictors in binary classification of membrane proteins.
Contact: catherine.mooney@ucd.ie

Introduction

Proteins that are part of or attached to the membrane of the cell have special tasks that set them apart from others. For example, membrane proteins are used to help the immune system identify the body's own cells, as well as any cells that have marked themselves for destruction, perform passive and active transport of molecules out of and into the cell, act as receptors for signal chemicals and protect the cell from outside threats. Because of the diversity of these functions, membrane proteins are common targets for drugs, and their dysfunction can cause many different problems and even disease (Almén *et al.*, 2009). Machine learning, a subarea of artificial intelligence, is a powerful tool, capable of solving complex problems like the game of Go (Silver *et al.*, 2016), and even the diagnosis or skin cancer (Esteva *et al.*, 2017). Deep Learning is an more specialised subarea of machine learning that is able to solve problems involving minute details that could help to classify a very complex problem, such as predicting a protein's subcellular localization. Computers can help scientists in this way, providing a prediction of where a protein will localize after it is

produced, giving the scientist an idea of where to look for the protein first as soon as it has been sequenced. This allows for work to begin using that protein before the time consuming and expensive task of identifying its subcellular localization experimentally.

Project Goals

This project seeks to create an artificial neural network capable of making predictions on whether or not a given protein is located on the cell membrane or elsewhere via deep learning. The ratio of proteins with useful annotation data such as subcellular localization and proteins with known sequences has been increasing because of the difficulty and costliness of determining these pieces of annotation data experimentally versus the cost of sequencing. Tools like this project aim to use this abundance of sequence data to help make the production of this annotation data easier and more cost effective. The specific role of predictors like SCLpred-Mem is to give researchers an idea of where a protein will be sent after it is manufactured to determine if it is worth further study, and because of the usefulness of membrane proteins, it is valuable just to know if a protein is one, as it is more likely that it is involved in a potential medical procedure or treatment, hence why SCLpred-Mem specialises in membranes.

Overview of Similar Projects

Many systems in this research area are iterative, meaning they improve on a past algorithm, be it their own previous version or a totally different predictor. Another strategy is combining the results of many predictors to make a new, more accurate predictor. Several examples of these techniques can be found in the list of current predictors. The number of classifications each network is capable of is highly variable, with ranges from two onward. For example, BUSCA has upwards of 12 different classes, WoLF PSORT states more than 10, and DeepLoc having 10 classes in its normal mode, but two (membrane and non-membrane) also available as part of its results. The predictors all use machine learning in two different ways, those dealing with sequence data and those dealing with annotation data (Wei *et al.*, 2018). Sequence based predictors use information contained directly in the protein's sequence of amino acids, whereas annotation based predictors use data other than sequence that is discovered by researchers about the protein, such as gene ontology (Wei *et al.*, 2018). Gene Ontology data contains descriptions of cellular components and molecular functions of gene products, among other data concerning protein function (Wei *et al.*, 2018). The technique that has pushed newer predictors up and over the capabilities of the older ones is deep learning. Previously these predictors relied on data that was already processed and designated into certain groups by humans. With deep learning, the predictor can automatically classify the proteins into these groups using just GO data or the protein's sequence, depending on the type of predictor. This simplifies the process significantly and is why machine learning has become so common for this purpose (Min *et al.*, 2017). The most common deep learning techniques for this are deep neural networks, convolutional neural networks (which this project is using), and recurrent neural networks (Min *et al.*, 2017). Some predictors even combine these different techniques.

List of Benchmarked Predictors

These predictors were chosen by examining the predictors released in 2017 and onward that classified proteins into membrane and non-membrane categories. Two other older predictors were included because they were very commonly cited and benchmarked against the newer predictors as well.

WoLF PSORT (2007) This predictor extends the PSORT method. It converts the amino acid sequence into numerical vectors and classifies them using a weighted k-nearest neighbour classifier. It uses a wrapper to select only the features relevant to the current task. It uses UniProt version 45, and is divided into animal, plant and fungi. WoLF PSORT was published in 2007, and uses the UniProt data from that year (Horton *et al.*, 2007).

Available at: https://wolfpsort.hgc.jp/

LocTree2 (2012) & LocTree3 (2014) This predictor classifies eukaryotes into 18 total classes. It uses PSI-BLAST and multiple support vector machines in a hierarchical fashion as detailed in figure 3. Each SVM has a binary output, but they are all arranged to form the final output. The accuracy of LocTree2 is 65% for its 18 classes, but as for its first prediction, which is membrane vs non-membrane, just like this project, is 94% (Goldberg *et al.*, 2012). LocTree3 is simply an enhanced version of LocTree2 that has a better accuracy of 80 ± 4 % for eukaryotic proteins in their 18 classes (Goldberg *et al.*, 2014).

Available at: https://rostlab.org/services/loctree3/

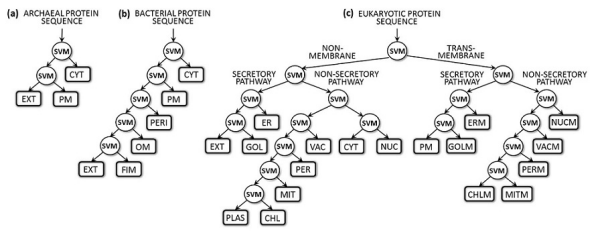


Fig. 1. LocTree2 Classification Hierarchy (Goldberg et al., 2012)

MDLoc (2015) This predictor makes use of a dependency based generative model. It can classify proteins into more than one output class, and uses DBMLoc dataset . As this predictor can place proteins into more than one class, a procedure to adapt its results into a format comparable to the results from this project will need to be created (Simha *et al.*, 2015).

Available at: http://128.4.31.235/

DeepLoc (2017) This predictor uses a convolutional neural network in conjunction with a recurrent neural network and the UniProt dataset, and is sequence based. It predicts 10 locations, and has a another mode that just classifies membrane bound proteins versus soluble ones. This second mode is exactly what we hope to create with SCLpred-Mem, so its benchmarking results will be valuable for evaluating the effectiveness of SCLpred-Mem(Almagro Armenteros *et al.*, 2017).

Available: http://www.cbs.dtu.dk/services/DeepLoc/

SubCons (2017) This predictor uses an ensemble of 4 predictors (CELLO2.5, LocTree2, MultiLoc2 and SherLoc2) for its predictions, therefore taking advantage of sequence and annotation based predictions. It reports an F1-Score of 0.79 (Salvatore *et al.*, 2017).. It places proteins into 9 different classes including membrane, and uses the Mass-Spec dataset, the SLHPA dataset and UniProt

Available at: http://subcons.bioinfo.se/pred/

FUEL-mLoc (2017) This predictor can also predict more than one location for a protein, and handles all eukaryotes and some prokaryotes. It uses gene ontology for its predictions, and is therefore an annotation based predictor. It uses two new databases for its datasets: ProSeq and ProSeq-GO. It has a

79.3% accuracy for eukaryotes, and a 74.3% accuracy for humans. It uses an elastic net multi label classifier for its predictions (Wan *et al.*, 2017a).

Available at: http://bioinfo.eie.polyu.edu.hk/FUEL-mLoc/index.html

BUSCA (2018) This predictor combines 8 other predictors (DeepSig, SChloro, TPpred3, BetAware, MemLoci, PredGPI, BaChelLo, ENSEMBLE3.0) to produce its results. It uses different pipelines for each type of input, for example eukaryotes. The predictors are chosen so that the prediction gets more refined as the system runs. Its eukaryotic workflow can be seen in Figure 2. It uses methods for identifying signal and transit peptides, GPI anchors, transmembrane domains and tools for discriminating location of globular and membrane proteins. For animals and fungi it has 9 different locations it can place a protein in, and 16 for plants. It works with plant, animal and gram positive and negative bacteria proteins. They report an F1 accuracy score of 0.49 (Savojardo *et al.*, 2018).

Available at: http://busca.biocomp.unibo.it/

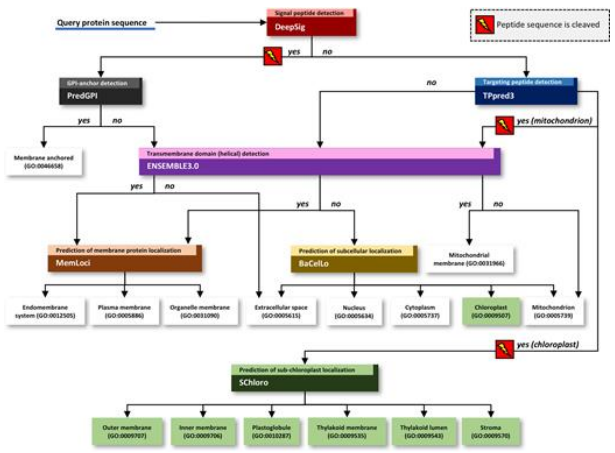


Fig. 2. BUSCA's workflow for eukaryotic prediction (Savojardo et al., 2018)

Methods

Architecture SCLpred-Mem uses a stack of N-to-1 neural networks. The N-to-1 neural network is made with an input kernel mapping a window of amino acids into a feature vector followed by a stack of hidden convolutional kernels followed by a fully connected network. The input Kernel learns a non-linear function I from a window of amino acids $\hat{ic}_i$ at position i and predicts an intermediate state vector is^i at position i.

$$\hat{is}_i = I(\hat{ic}_i)$$

$$\hat{ic}_i = (i-c, ..., i, ..., i+c)$$

Each hidden convolutional kernel learns a non-linear function H^k at hidden layer k from a window of intermediate states $\hat{hc}_j^{\,k}$ at position j and predicts an intermediate state vector hs_i^k at position i.

$$hs_i^k = H^k(\hat{hc}_j^{\,k})$$

$$\hat{hc}_j^{\,k} = (j-\gamma, ..., j, ..., j+\gamma)$$

The output vectors hs_p^l of the last hidden kernel at each position p is averaged element-wise into a single vector $\hat{v}$. A fully connected network predicts the final subcellular location cls of each protein from the final vector $\hat{v}$. The fully connected network learns a non-linear function O.

$$cls = O(\hat{v})$$

In this work, we tested different configurations of the N-to-1 neural network and employed a wider neural network with a large number of weight sharing as opposed to a deeper network. This is due to the fact that wider networks with larger input window of amino acids tend to capture more of the targeting signals encoded within the sequence resulting in better performance whereas deeper networks tend to increase the capacity of the network resulting in overfitting.

Evaluating Performance To evaluate the performance of SCLpred-Mem we measure the true positive rate (TPR) and false positive rate (FPR) as we increase the discrimination threshold from 0 to 1 where the membrane class is the positive class. The results are shown as a Receiver Operating Characteristic (ROC) curve where TPR is plotted against FPR, which are calculated as follows:

$$\begin{aligned} TPR &= \frac{TP}{TP+FN} \\ FPR &= \frac{FP}{FP+TN} \end{aligned}$$

where:

- True positives (TP): the number of proteins predicted in a class that are observed in that class.
- False positives (FP): the number of proteins predicted in a class that are not observed in that class.
- True negatives (TN): the number of proteins predicted not to be in a class that are not observed in that class.
- False negatives (FN): the number of proteins predicted not to be in a class that are observed in that class.

The area under the curve, AUC, which is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance is also shown. The AUC is a number between 0 and 1 where 0.5 indicates a random model and 1 is perfect. R (R Core Team, 2018; Robin *et al.*, 2011) is used to plot the curves and calculate the AUC. specificity (Spec), sensitivity (Sen), false positive rate (FPR), Matthews Correlation Coefficient (MCC) and the accuracy (Q) at a 0.5 threshold are measured as follows (Baldi *et al.*, 2000):

$$\begin{aligned} Spec &= 100\frac{TP}{TP+FP} \\ Sen &= 100\frac{TP}{TP+FN} \\ FPR &= 100\frac{FP}{FP+TN} \\ MCC &= \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \\ Q &= 100\frac{TP+TN}{TP+TN+FP+FN} \end{aligned}$$

MCC measures the correlation coefficient between the observed and predicted classifications. A value of 1 represents a perfect prediction, 0 a random prediction and -1 an inverse prediction and is a good indicator of the overall performance of the predictive methods for both membrane and non-membrane proteins.

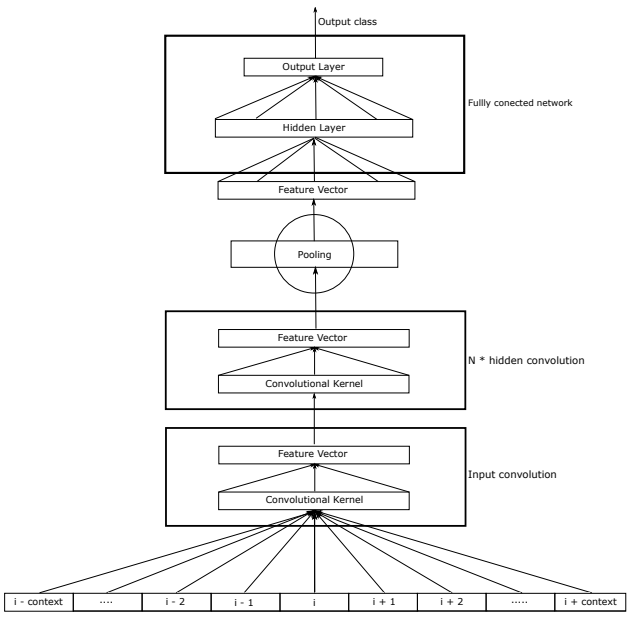
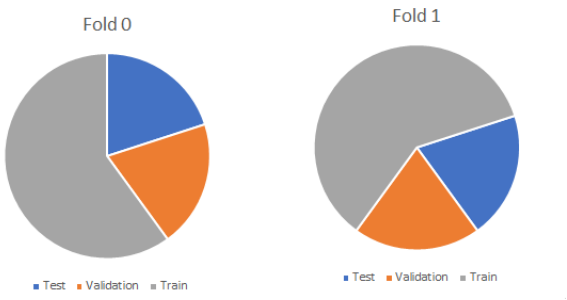


Fig. 3. Deep N-to-1 architecture.

Benchmarking We recast the predicted locations of other available web servers into membrane and non-membrane categories in order to benchmark them against SCLpred-Mem using our independent test sets. After initial benchmarks with relatively low scores from all networks, we opted to create a second ITS with less strict homology reduction. The original ITS will be referred to as the "strict ITS", and the less strict later ITS will be referred to as the "new ITS". The recasting was very simple, any membrane protein passing through the endomembrane system was recorded as positive and anything else was recorded as negative. For the case of predictors with multiple class predictions, if they predicted a membrane location in addition to another, the prediction was counted as the positive class.



(a) Fold 0 (b) Fold 1

Fig. 4. As shown in the charts, the data rotates between test, train and validation sets between folds. This process is furthered with each fold.

Training and Independent Test Datasets All eukaryotic entries were downloaded from the UniProtKB release 2019_05 (UniProt Consortium, 2019)– 154,743 proteins from 7,859 species. These come from the human, rodent, mammal, vertebrate, invertebrate, fungi and plant sections of UniProt. We then removed any proteins that were not supported by published experimental evidence of subcellular location. The entire training set was 80% redundancy reduced by comparing each protein to the rest of the set and removing any sequence more than 80% similar to the query. The individual datasets for each of the five folds of the 5-fold cross-validation were created at this point. For fold zero, the first fifth of the data became fold zero's test set, the second fifth became fold zero's validation set, and the third, fourth and fifth fifths of the data became the training set for that fold. In fold one, all of these numbers were rolled forward one: The test set was the second fifth of the data, validation the third, and train the fourth, fifth and first. This process was continued for each fold until we had five. Figure 4 shows this process graphically. From here, each fold's validation set is redundancy reduced with respect to the fold's test set, shrinking the validation set. The fold's test set is then internally redundancy reduced using the same procedure as the last reduction. Finally both the validation and test sets of each fold are then redundancy reduced with respect to the fold's training set. The Independent Test Set, or ITS is first redundancy reduced internally and then with respect to the training set. We also created a "less strict" set using a 30% internal redundancy reduction and with respect to the training set. After the datasets have been made, PSI-BLAST (Altschul *et al.*, 1997) was used to generate alignments of multiple homologous sequences (MSA) for all proteins. PSI-BLAST was run for three rounds with an e-value of 0.001 against the June 2016 version of UniRef (UniProt Consortium, 2019). The inputs to the models are residue and gap frequency profiles extracted from MSA, where sequences are weighed by the information they carry with respect to the plain profile, and the frequency of the residue present in the original sequence is "clipped" to 1 (Torrisi *et al.*, 2018). This process involves comparing each amino acid in the protein to the other proteins in the database, and making spaces in the datasets to "align" the sections on the query protein with where they are found in the database. The percentage of proteins in the database that have that particular amino acid in that location is then taken and stored as a value between 0 and 1. These percentage values are what makes up the files that the training and prediction programs later run. This approach encodes important evolutionary information that helps the network with its predictions.

Table 1. Distribution of classes in the Training and Testing sets

	Membrane	Other	Total
Training Set	2,766	2,026	4,792
Strict Independent Test Set	45	72	118
New Independent Test Set	119	121	240

Tuning the Hyperparameters There are several parameters used to configure the network, and testing was performed to determine the appropriate configuration of the network to perform the best on any general dataset, and to prevent overfitting. All of these tests were performed on the dataset from fold zero of the 5-fold cross-validation for consistency. The parameters varied were the number of kernel layers (NLayers), the learning rate (LrnRt), and the width of the examined data with each cycle by the kernel layers (context for the first layer and gamma for all subsequent ones), which were varied from zero to 15. This number measures the the number of units of data examined on either side of the center unit. As such, at zero the total number would be one, at 10 the total is 21, etc. These results are shown in 2. MCC here is the best found on the validation set, and is what we used to gauge the capabilities of that configuration of the network. Round 2 of this testing was informed by the results obtained in round 1, and tested more fine parameters once the ones with larger impact were determined such as the size of the input and output layers of the kernel layers (NH and H). We observed increases in performance as the gamma trended towards 15. The size of the inputs and outputs of the kernel layers yielded the best performance when they were both set to 10,

and the most effective number of internal kernel layers was 2, for a total of 3 kernel layers.

Table 2. Results of the different parameter tests and their parameters.

	Round		One			
NLayers	LrnRt	Context	Gamma	NH	H	MCC
0	0.05	5	N/A	10	10	0.558
0	0.01	5	N/A	10	10	0.567
0	0.05	10	N/A	10	10	0.600
0	0.01	10	N/A	10	10	0.616
0	0.05	15	N/A	10	10	0.604
0	0.01	15	N/A	10	10	0.603
1	0.05	0	5	10	10	0.608
1	0.01	0	5	10	10	0.608
1	0.05	0	10	10	10	0.600
1	0.01	0	10	10	10	0.635
1	0.05	0	15	10	10	0.613
1	0.01	0	15	10	10	0.638
2	0.05	0	5	10	10	0.635
2	0.01	0	5	10	10	0.631
2	0.05	0	10	10	10	0.639
2	0.01	0	10	10	10	0.655
	Round		Two			
NLayers	LrnRt	Context	Gamma	NH	H	MCC
1	0.01	0	5	10	5	0.611
1	0.01	0	10	10	5	0.632
2	0.01	0	5	10	5	0.604
2	0.01	0	5	10	10	0.643
2	0.01	0	5	25	15	0.659
2	0.01	0	10	10	10	0.653
2	0.01	0	10	10	5	0.650
2	0.01	0	15	10	10	0.706
2	0.01	0	15	10	5	0.643
3	0.01	0	5	10	10	0.475
3	0.01	0	10	10	10	0.674

Training and Ensembling After finding a final model from hyperparameter testing, we trained the full 5-fold cross-validation network. This network is what generated the final predictions to represent our network in the results section. The models we trained are stacks of three convolutional kernels followed by average pooling and two fully connected layers, with six inner layers in total containing roughly 540 weights and taking in all motifs of 21 residues. There are no learn-able weight parameters between the kernel layers. To run both versions of the ITS, the network uses the five most effective models from each fold in an ensemble of 25 total models. This same process is repeated for the test sets of each fold and the results of each fold were averaged to yield the results for the training set.

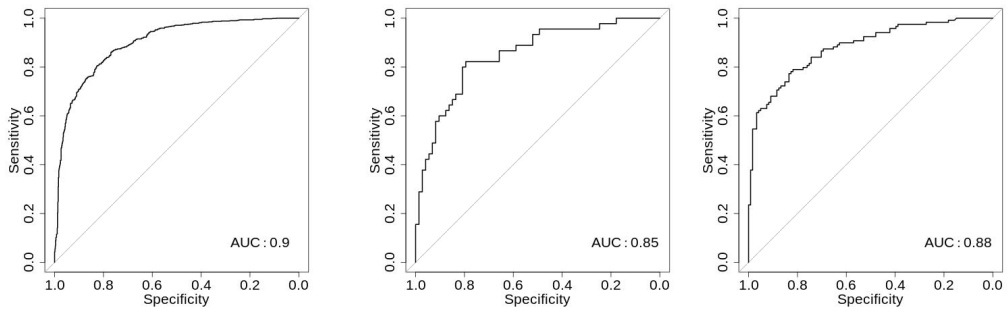
Results and Discussion

For every protein, SCLpred-Mem predicts the probability (between 0 and 1) of that protein localizing to a membrane. The application predicts if a given protein will be localized to any of the cell's membranes, provided they have been routed through the endomembrane system. This is because this predictor is planned to be used in conjunction with another by some of the authors that predicts proteins into the EMS or otherwise. The output from that predictor will then be used with this predictor. Because of this, two membranes, the nuclear membrane and the mitochondrial membrane, are not sorted into the membrane (positive) class for the purposes of benchmarking the other predictors or SCLpred-Mem. The closer the predicted probability is to 1, the more confident SCLpred-Mem is in that prediction. The results of 5-fold cross-validation for the training dataset, are shown in Table 3 and in Figure 5 as a Receiver Operating Characteristic (ROC) curve with thresholds increasing from 0 to 1, i.e. the cut-off above which a protein is considered to be predicted as localizing to one of the cell's membranes. Details of the calculation of these metrics are provided in methods. We benchmarked SCLpred-Mem against WoLF PSORT (Horton *et al.*, 2007), LocTree3 (Goldberg *et al.*, 2014), DeepLoc 1.0 (Almagro Armenteros *et al.*, 2017), HPSLPred (Wan *et al.*, 2017b), SubCons (Salvatore *et al.*, 2017), FUEL-mLoc (Wan *et al.*, 2017a), and BUSCA (Savojardo *et al.*, 2018). Benchmarking against MDLoc (Simha *et al.*, 2015) was also planned, but the web server only accepts one protein sequence at once, and would have taken excessive time to feed in the ITS. SCLpred-Mem outperforms all other predictors tested based on MCC bar DeepLoc, which scored very similarly to our predictor.

Table 3. Benchmarking of SCLpred-Mem against other available web servers on the independent test set.

	MCC	Spec	Sen	FPR
SCLpred-Mem (SITS)	0.52	77.6%	90.4%	42.2%
SCLpred-Mem (NITS)	0.59	68.8%	98.3%	45.4%
5-fold cross-validation	*0.63*	*79.4%*	*81.1%*	*18.5%*
Strict ITS				
DeepLoc	0.51	87.5%	46.6%	4.1%
LocTree3	0.35	90.9%	22.2%	1.4%
BUSCA	0.27	70.6%	27.3%	7.4%
SubCons	0.37	91.7%	25.0%	1.5%
WoLF PSORT	0.20	64.3%	20.0%	6.9%
FUEL-mLoc	0.03	40.7%	24.4%	22.2%
New ITS				
DeepLoc	0.63	93.8%	63.9%	4.1%
LocTree3	0.42	91.5%	45.4%	6.3%
BUSCA	0.33	84.4%	31.9%	5.9%
SubCons	0.48	96.0%	41.2%	1.7%
WoLF PSORT	0.40	83.9%	43.7%	8.3%
FUEL-mLoc	0.27	70.8%	42.9%	17.6%

Assessing SCLPred-Mem on the TS and ITS A general overview of the performance of SCLPred-Mem can be seen in Figure 5. In these ROC charts, sensitivity is plotted on the X axis, and the specificity on the Y axis. Ideally, as the specificity approaches 1, sensitivity also approaches 1. On the graph, an ideal point is in the top left corner, with an ideal curve having an inverted L shape. Here, specificity is the probability that a positive prediction is correct, and sensitivity is the probability of correctly predicting a positive example (Baldi *et al.*, 2000). SCLpred-Mem has a moderate specificity, with a very high false positive rate. This indicates that the network may be setting its biases too high in regards to membrane proteins. The sensitivity of SCLpred-Mem is also much higher than any other predictor in this list, meaning that it is very likely to get classify a membrane protein correctly. Its high false positive rate also shows that it predicts membrane too often. This is probably at least somewhat caused by there being more membrane proteins than non-membrane proteins in the training set, but the design of the network is also most likely to blame. SCLpred-Mem's performance did improve from the strict ITS to the new ITS, but not as much as the other predictors. This is likely because SCLpred-Mem's training set was created using the same strict standards as the strict ITS. It is also an indicator that SCLpred-Mem is more consistent across different datasets. Because of the distribution of the specificity and sensitivity on both versions of the ITS, there is likely a small change we could make without retraining that would improve results. In this benchmark, the probability required to cast a protein into the membrane



(a) 5-fold cross-validation (b) Strict Independent test set (c) New Independent test set

Fig. 5. ROC plot of SCLpred-Mem predictor performance (a) in 5-fold cross-validation on the training set, (b) on the strict independent test set and (c) on the new independent test set.

class was 0.5, or 50%. Since the membrane class is overpredicted, it is likely that simply increasing this number to make it harder for proteins to be classed this way would improve results, and could be accomplished without retraining of the network.

Benchmark Classes Benchmarking was carried out by submitting both versions of our ITS to each predictor we benchmarked against. All of these predictors have different output formats and different classes they predict into. Below is a list of all of the predictors, with the number of each one's classes, and which of our classes their classes were placed into. Each predictor's performance is also discussed. Initially the results on the ITS were low, so we created a new ITS later in testing with less strict redundancy reduction standards in order to verify the reason for the low results on the strict ITS. This new ITS has 240 sequences and was created by taking the original 80% redundancy reduced ITS of 529 sequences and 80% redundancy reducing it with respect to the training set, then 30% internally redundancy reducing it. The final results can be seen in table 3, with explanations of the statistics in the SCLpred-Mem assessment section.

WoLF PSORT (Horton *et al.*, 2007) states more than 10 classes, represented in their output by 4 letter codes. These codes are supposedly documented on the server, but no such documentation could be found. It should be noted that the server mentioned in the original 2007 publication is no longer online, and the predictor has since been rehosted on at a new site. Of all of the 4 letter codes, only one, "plas" was taken to be a membrane, and counted as our positive class. All others were counted as negative. Its MCC on our strict ITS was 0.20. On the new ITS, the MCC is doubled to 0.40, which indicates the lower score on the strict ITS was caused by the relative difficulty of the strict ITS. This predictor's specificity and sensitivity also showed marked improvements across the ITS versions, so it likely has a less diverse training set than some of the higher performing predictors.

SubCons This predictor predicts into 9 classes (Salvatore *et al.*, 2017). The output format for its predictions is 3 letter codes. The only membrane code was "MEM", and was counted as the positive class. All others were counted negative. All statistics improved over the different ITS versions, markedly the sensitivity. This could mean that their training data had only the more common membrane proteins, which the new ITS contained more of. This explains the increase in MCC and sensitivity while the specificity and false positive rate remain the same.

LocTree3 (Goldberg *et al.*, 2014) predicts eukaryotic proteins into 18 classes. Its output format simply has the classes listed by their names. The classes "golgi apparatus membrane", "endoplasmic reticulum membrane" and "plasma membrane" were sorted into the positive class. All others were placed in negative. There may have been more membrane classes, but these were the only ones to be predicted for our ITS. LocTree3's change in statistics over the ITS versions is very similar to SubCons (above), but the FPR increases more on the new ITS. This combined with the increase in sensitivity means that this predictor predicted membrane on the New ITS more than on the old, again indicating that its training data may not have contained the less common membrane sequences.

FUEL-mLoc (Wan *et al.*, 2017a) outputs its data with the name of the predicted location. Of all of the predictions for both versions of the ITS, the only membrane class was "Cell-Membrane", so that prediction was counted positive, with the rest as negative. FUEL-mLoc was the lowest scoring predictor, but its performance did increase with the new ITS. This predictor's high false positive rate and low specificity and sensitivity indicate that it did predict membrane often, but was also wrong somewhat often. Every statistic improved markedly with the newer dataset, possibly indicating a narrow dataset, but the low overall performance may indicate a deeper issue. This predictor was the only predictor to predict more than one class for each protein, which may mean that it sacrifices some of its accuracy on a smaller scale task such as membrane prediction in order to be more accurate on criteria used for multi-class prediction.

BUSCA (Savojardo *et al.*, 2018) is a very recent predictor from 2018 that predicts into more than 12 classes. In its output for the ITS, we treated "anchored component of plasma membrane" and "plasma membrane" as positive, all others negative. BUSCA also sometimes would return "organelle membrane," which is not specific enough to be classed as a membrane. In this case, the gene ontology terms associated with the protein were included with the results, and if one of these was associated with a membrane whose proteins travel through the EMS, it would be counted positive. If it had GO terms counting it outside the membrane class we counted it negative. Some of these had no GO terms, and these results were simply discarded and placed with the proteins that BUSCA was incapable of running. BUSCA, like all others, showed increased performance with the new ITS. Its primary increase was in specificity, meaning its positive predictions were more accurate. Interestingly, sensitivity did not increase as much, meaning that the improvement this time was in the accuracy of the negative class. The percentage of correct positive predictions was increased, but the percentage of correct positives only improved a little. This indicates fewer false positives and more true negatives, and means that the predictor may have different types of membrane proteins than what were offered in the strict ITS, but were more prevalent in the new ITS.

DeepLoc (Almagro Armenteros *et al.*, 2017) has the best results of the benchmarked predictors. Its output has two fields, one for subcellular localization, and another for membrane or non-membrane, just like SCLpred-Mem. Benchmarking was done based on the binary membrane classifier, as that is the most relevant comparison we could have made. SCLpred-Mem reports slightly better performance on the strict ITS than DeepLoc, but DeepLoc reports slightly superior results on the new ITS. This may be because of SCLpred-Mem's stricter training set, which had

less representation of more common membrane proteins. This would explain the better performance (relative to DeepLoc's performance) on the strict ITS. DeepLoc's training data may have been less strict, enabling it to perform better on a less strict dataset seeing more representation of common membrane sequences. DeepLoc has high specificity, but low sensitivity, indicating that the membrane proteins it does predict are usually correct, but that it misses many other membrane proteins by classifying them as non-membrane. This is interesting because it is almost the exact opposite of the results seen from SCLpred-Mem, especially on the new ITS. This means that SCLpred-Mem and DeepLoc have different strengths, and could possibly be used in conjunction with each other, an idea given more thought in the conclusion.

Table 4. Outputs of benchmarked predictors that were categorised into our membrane (positive) class; any class not in this table was counted negative.

Predictor	Membrane Classes
WoLF PSORT	"plas"
SubCons	"MEM"
LocTree3	"golgi apparatus membrane" "endoplasmic reticulum membrane" "plasma membrane"
FUEL-mLoc	"Cell-Membrane"
BUSCA	"anchored component of plasma membrane" "plasma membrane" "organelle membrane" (with positive GO terms)
Deeploc	"Membrane"
Preditctor	**Non-Membrane Classes**
WoLF PSORT	"E.R", "extr", "nucl", "cyto", "golg", "mito", "cysk"
Subcons	"MIT", "EXC", "CYT" "NUC," "ERE", "PEX"
LocTree3	"secreted", "mitochondrion", "cytoplasm", "peroxisome", "golgi apparatus", "nucleus", "vacuole" "endoplasmic reticulum", "mitochondion membrane"
FUEL-mLoc	"Vacuole", "Cytoplasm", "Endoplasmic-Reticulum", "Golgi-Apparatus", "Extracellular", "Cytoskeleton", "Nucleus", "Mitochondrion", "Chloroplast", "Peroxisome", "Lysosome", "null", "Acrosome"
BUSCA	"endomembrane system", "extracellular", "nucleus", "chloroplast", "cytoplasm", "mitochondrion", "organelle membrane" with negative GO terms
DeepLoc	"Soluble"

These are all the classes that were predicted for the data we submitted. Each predictor may be capable of more, but those classes were not predicted for any of the sequences in both ITS

Benchmark Analysis As seen in Table 3, most predictors had a similar trend between the two versions of the ITS. All showed improvement on the new ITS, and all had higher specificities than sensitivities. This means that, overall, most predictors are predicting less membrane predictions than there are membranes, as shown by the low false positive rates across the board. This is in direct contrast to the results from SCLpred-Mem, which has an opposite trend. SCLpred-Mem predicts more membrane proteins than there are most times, yielding a very high sensitivity, and a false positive rate far higher than that of any predictor benchmarked. Although SCLpred-Mem yielded a similar accuracy score to the current leader in the field, their results were actually quite contrary to each other in every other way. It is also interesting that there is only one other predictor that can produce results at the same level as SCLpred-Mem. The rest of the predictors may have lower results in comparison to DeepLoc and SCLpred-Mem because they are not specifically designed with this type of prediction in mind.

Usability Analysis All of the different predictors had totally different user interfaces, and some had more robust features than others. For example, one predictor that was planned to be used for benchmarking, MDLoc (Simha *et al.*, 2015), had a user interface that only accepted one protein sequence at a time. This is why it was cut from benchmarking, as it would have taken hours to complete benchmarking using the 118 or 240 sequence ITS. Usability of the server is a large factor in why one predictor may be more usable than another, and this section will analyse what is something to include in our server and what should be left out. Several predictors had systems in place that allowed the results to be emailed to the user after all of the sequences were finished running. This is a very convenient feature and is one that will be included on SCLpred-Mem's server. The implementation of this varied. For DeepLoc, an email can be sent for any job, and the resultant email contains a link that can be used to retrieve the results. This is convenient, however the link eventually expires and the results are lost. Upon clicking the link, the user can download a file summary of the results, and it would be convenient if that file were attached to the email. FUEL-mLoc includes this feature in their server, and sends a summary of their predictions with their email. Unfortunately, it is bugged and returns every prediction as either "Vacuole", or "null". This happens whenever a FASTA file is submitted, not just when the email feature was used. The only way to get a real prediction from FUEL-mLoc is by using the text input box, which did not have the email feature. BUSCA's interface was well designed, despite lacking an email feature, however it had a serious flaw in that it cannot accept any input protein shorter than 40 residues. Instead of filtering these proteins out and not running them, it just rejects the entire job until they are removed, requiring the production of a script to filter out these short proteins in order for the job to run. SubCons also has limits on length for its proteins, but if it encounters one outside these limits, it places the sequence in a text file containing all of the offending sequences for the user to view once the job is completed. This is how SCLpred-Mem will be handling any problem proteins. Most predictors had an option to download a summary of their results, but two, LocTree3 and WoLF PSORT did not. WoLF PSORT outputs all of its data in one massive line of text that was not easily parsed. A script had to be made in order to break up this text into a more readable form. LocTree3 had a problem where, when under load, it would predict some of the submitted proteins but not all of them, in a seemingly random order. This put the results out of order, and it reported the job as done when it was not as, often, more than half of the sequences remained unfinished. One complication in benchmarking arose over the fact that many of the predictors (all except WoLF PSORT and FUEL-mLoc) did not return their results in the same order that they had been submitted, complicating the design of the benchmarking scripts greatly. SubCons has a very useful feature in that it remembers the user and will allow them to access the results of their past jobs from the home page of the server. This is a very convenient feature. DeepLoc has a strange requirement of only accepting 50 sequences at a time, which did not allow us to use the FASTA file submission feature, as the file contained all 118 or 240 sequences. Multiple jobs of 50 or less sequences had to be run in parallel, and their resulting files combined together after deleting their headers to form a continuous file. The most important flaw for many of these predictors is that their results are in a coded format that has no explanation. Some publications stated a guide to be available on the server but they were not. In the case of WoLF PSORT and SubCons, codes such as "plas" and "MEM" were used, and their meanings could only be assumed because there was no explanation given. BUSCA classified some proteins only as "organelle membrane", which is highly unspecific. Occasionally these had more specific gene ontology terms that could be substituted, but not always. These flaws complicated

Table 5. Results of the strict and new ITS shown side-by-side.

	MCC(SITS)	MCC(NITS)	Spec(SITS)	Spec(NITS)	Sen(SITS)	Sen(NITS)	FPR(SITS)	FPR(NITS)
SCLpred-Mem	0.52	0.59	77.6%	68.8%	90.4%	98.3%	42.2%	45.4%
DeepLoc	0.51	0.63	87.5%	93.8%	46.6%	63.9%	4.1%	4.1%
LocTree3	0.35	0.42	90.9%	91.5%	22.2%	45.4%	1.4%	6.3%
BUSCA	0.27	0.33	70.6%	84.4%	27.3%	31.9%	7.4%	5.9%
SubCons	0.37	0.48	91.7%	96.0%	25.0%	41.2%	1.5%	1.7%
WoLF PSORT	0.20	0.40	64.3%	83.9%	20.0%	43.7%	6.9%	8.3%
FUEL-mLoc	0.03	0.27	40.7%	70.8%	24.4%	42.9%	22.2%	17.6%

benchmarking greatly and made working with these systems far more work than it needed to be. With the future implementation of SCLpred-Mem in a server, we will use this information on design to include these innovations and exclude these flaws. The following will be implemented in our server:

- Functional email support for all jobs, no matter the input format.
- Emails will contain a link to the results that will be temporarily available on the server. A summary of the results will also be attached in case the data is needed after the link expires.
- Output will be in plain English and as specific as possible, although this is less of a problem with a binary classifier like this one.
- Any protein that cannot be predicted by our system will be communicated to the user in a text file. Any FASTA file will be accepted by the system, with sequences too long or short being removed automatically.
- Direct text box input and FASTA file input will be implemented in the exact same way to eliminate problems with inconsistent results between the FASTA and text input formats.
- If the server is under load, new jobs will be queued until the system has time to run all of the sequences, and will not report the job completed until it is.
- Results will list the submitted sequences in the same order that they were submitted.
- The user will be remembered by the system and a link on the homepage will allow them to view all previously completed jobs they have submitted, provided that the jobs aren't past a certain age.

Conclusion

SCLpred-Mem's results are superior to or comparable to other state of the art predictors available today. Of the predictors benchmarked, there was only one capable of producing results in the same format and for the same purpose as SCLpred-Mem, and that was DeepLoc. While DeepLoc performed comparably to SCLpred-Mem on the strict ITS, they performed slightly better on the new ITS. With this information gained in this analysis, SCLpred-Mem could even be improved in the future to fix some of its issues outlined in the results section, such as overpredicting the membrane class. In the context of the field that these predictors would actually be performing in, it is good to have more than one capable and accurate tool for a specific task. Since none of these predictors perform perfectly, and none ever will with this machine learning approach given the nature of the problem, it is good to run queries meant for scientific purposes through more than one predictor in order to guarantee the veracity of the results. The process of identifying a protein's subcellular localization is a time consuming and costly one, and these predictors are meant to give scientists an idea of where to look for these proteins first. They would not want to rely on the results of just one predictor, but if the next best predictor were not nearly as accurate, its results would not be worth counting as a possible location for the protein. Because of this, it is valuable to have multiple predictors with comparable results for general datasets, as that allows for biologists to have more than one prediction at their disposal and can diversify the data that they work with. With this in mind, SCLpred-Mem has a valid place in the network of subcellular localization predictors, and can even possibly be improved for a future version before it is hosted publicly online.

Future Work This predictor is planned to be a part of a larger predictor. Much like how BUSCA (Savojardo *et al.*, 2018) functions, this predictor will be a part of a larger ensemble predictor. Another piece planned is the Endomembrane System version of this predictor, which is currently being worked on by some of the authors. It is important that the individual pieces of an ensemble predictor all perform well, as if one branch of the path were to be weak, the predictor's results would be brought down significantly. Another important step forward would be to host this predictor on a webserver and have it be available to the public. Many of the other predictors available had strange output formats and user interfaces. Many lacked important information, such as a list of all of the possible prediction classes, or a guide to understanding the coded output. In our server we will have all of this information available and easy to find. The key here is to maintain the predictor and server. One of the predictors we benchmarked, WoLF PSORT, had its original server go offline, and was assumed rehosted on the current server it resides on now. Certain information mentioned in the original publication that was supposed to be on the server was not available on the rehosted version. If the server were to go down, it would be important that we keep the URL up to a redirect page explaining what had been changed from what was detailed in the original publication and where to find any moved information. Another important maintenance task could be the occasional retraining of the network. Many of the benchmarked predictors seemed to be using the original model that had been trained with the original publication of the network. Databases are updated, and each update of the database gives us the opportunity to update the training set of our predictor to be more diverse and perform better with the proteins that are currently being focused on by the scientists who would be the users of our predictor.

Availability

SCLpred-Mem has not yet been put online as a server yet, but plans are in place to make it usable by the public depending on the results of this work. The user will be able to submit a list of protein sequences in FASTA format, and SCLpred-Mem predicts the probability that each of these proteins will localize to the cell's membranes versus all other locations.

Funding

This work was supported by the Irish Research Council [GOIPG/2014/603 to M.K.] and a UCD School of Computer Science Bursary.

Acknowledgements

The authors acknowledge contributions of the Research IT Service at University College Dublin for providing HPC resources that have contributed to the research results reported within this paper.

References

Almagro Armenteros, J. J., Sønderby, C. K., Sønderby, S. K., Nielsen, H., and Winther, O. (2017). Deeploc: prediction of protein subcellular localization using deep learning. *Bioinformatics*, **33**(21), 3387–3395.

Almén, M. S., Nordström, K. J., Fredriksson, R., and Schiöth, H. B. (2009). Mapping the human membrane proteome: a majority of the human membrane proteins can be classified according to function and evolutionary origin. *BMC Biology*, **7**, 50.

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, **25**(17), 3389–3402.

Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A., and Nielsen, H. (2000). Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, **16**(5), 412–424.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, **542**(7639), 115–118.

Goldberg, T., Hamp, T., and Rost, B. (2012). Loctree2 predicts localization for all domains of life. *Bioinformatics*, **28**(18), i458–i465.

Goldberg, T., Hecht, M., Hamp, T., Karl, T., Yachdav, G., Ahmed, N., Altermann, U., Angerer, P., Ansorge, S., Balasz, K., and et al. (2014). Loctree3 prediction of localization. *Nucleic Acids Research*, **42**(W1), W350–W355.

Horton, P., Park, K.-J., Obayashi, T., Fujita, N., Harada, H., Adams-Collier, C., and Nakai, K. (2007). Wolf psort: protein localization predictor. *Nucleic Acids Research*, **35**(Web Server issue), W585–W587.

Min, S., Lee, B., and Yoon, S. (2017). Deep learning in bioinformatics. *Briefings in Bioinformatics*, **18**(5), 851–869.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, J.-C., and MÃ¼ller, M. (2011). proc: an open-source package for r and s+ to analyze and compare roc curves. *BMC Bioinformatics*, **12**, 77.

Salvatore, M., Warholm, P., Shu, N., Basile, W., and Elofsson, A. (2017). Subcons: a new ensemble method for improved human subcellular localization predictions. *Bioinformatics*, **33**(16), 2464–2470.

Savojardo, C., Martelli, P. L., Fariselli, P., Profiti, G., and Casadio, R. (2018). Busca: an integrative web server to predict subcellular localization of proteins. *Nucleic Acids Research*, **46**(W1), W459–W466.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., and et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, **529**(7587), 484–489.

Simha, R., Briesemeister, S., Kohlbacher, O., and Shatkay, H. (2015). Protein (multi-)location prediction: utilizing interdependencies via a generative model. *Bioinformatics*, **31**(12), i365âŁ“i374.

Torrisi, M., Kaleel, M., and Pollastri, G. (2018). Porter 5: fast, state-of-the-art ab initio prediction of protein secondary structure in 3 and 8 classes. *bioRxiv*.

UniProt Consortium (2019). UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Research*, **47**(D1), D506–D515.

Wan, S., Mak, M.-W., and Kung, S.-Y. (2017a). Fuel-mloc: feature-unified prediction and explanation of multi-localization of cellular proteins in multiple organisms. *Bioinformatics*, **33**(5), 749–750.

Wan, S., Duan, Y., and Zou, Q. (2017b). Hpslpred: An ensemble multi-label classifier for human protein subcellular location prediction with imbalanced source. *PROTEOMICS*, **17**(17-18), 1700262.

Wei, L., Ding, Y., Su, R., Tang, J., and Zou, Q. (2018). Prediction of human protein subcellular localization using deep learning. *Journal of Parallel and Distributed Computing*, **117**, 212–217.