Bioinformatics, 33(21), 2017, 3387–3395
doi: 10.1093/bioinformatics/btx431
Advance Access Publication Date: 7 July 2017
Original Paper

OXFORD

Sequence analysis

DeepLoc: prediction of protein subcellular localization using deep learning

José Juan Almagro Armenteros[1,2,*], Casper Kaae Sønderby[2], Søren Kaae Sønderby[2], Henrik Nielsen[1] and Ole Winther[2,3]

[1]Department of Bio and Health Informatics, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark, [2]The Bioinformatics Centre, Department of Biology, University of Copenhagen, 2200 Copenhagen N, Denmark and [3]DTU Compute, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark

*To whom correspondence should be addressed.

Associate Editor: John Hancock

Received on March 16, 2017; revised on June 6, 2017; editorial decision on June 29, 2017; accepted on July 3, 2017

Abstract

Motivation: The prediction of eukaryotic protein subcellular localization is a well-studied topic in bioinformatics due to its relevance in proteomics research. Many machine learning methods have been successfully applied in this task, but in most of them, predictions rely on annotation of homologues from knowledge databases. For novel proteins where no annotated homologues exist, and for predicting the effects of sequence variants, it is desirable to have methods for predicting protein properties from sequence information only.

Results: Here, we present a prediction algorithm using deep neural networks to predict protein subcellular localization relying only on sequence information. At its core, the prediction model uses a recurrent neural network that processes the entire protein sequence and an attention mechanism identifying protein regions important for the subcellular localization. The model was trained and tested on a protein dataset extracted from one of the latest UniProt releases, in which experimentally annotated proteins follow more stringent criteria than previously. We demonstrate that our model achieves a good accuracy (78% for 10 categories; 92% for membrane-bound or soluble), outperforming current state-of-the-art algorithms, including those relying on homology information.

Availability and implementation: The method is available as a web server at http://www.cbs.dtu.dk/services/DeepLoc. Example code is available at https://github.com/JJAlmagro/subcellular_localization. The dataset is available at http://www.cbs.dtu.dk/services/DeepLoc/data.php.

Contact: jjalma@dtu.dk

1 Introduction

Proteins fulfil a wide diversity of functions inside the various compartments of eukaryotic cells. The function of a protein depends on the compartment or organelle where it is located, as it provides a physiological context for its function. However, aberrant protein subcellular localization can affect the function that a protein exhibits and contributes to the pathogenesis of many human diseases; such as metabolic, cardiovascular and neurodegenerative diseases, as well as cancer (Hung and Link, 2011). Therefore, predicting the subcellular localization of the proteins is an essential task which has been extensively studied in bioinformatics (Emanuelsson *et al.*, 2007; Imai and Nakai, 2010; Wan and Mak, 2015).

Most of the current machine learning methods for subcellular localization prediction extract a fixed number of features from the protein sequences and use this fixed length representation as input to a non-linear classifier such as a support vector machine (SVM). However, sequence-based models, which process one position at a time, are more natural for this task as they can learn and make inferences from input of varying length. Unfortunately, these models have not been competitive with non-linear classifiers up until recently. In this paper we take advantage of progress in deep learning, specifically recurrent neural networks (RNNs) with long short-term memory (LSTM) cells, attention models and convolutional neural networks (CNNs), to propose an end-to-end sequence-based model. LSTMs contain memory cells that can hold information from past inputs to the network for in principle an arbitrary number of positions (Hochreiter and Schmidhuber, 1997). Attention (Bahdanau *et al.*, 2014) makes it possible to detect sorting signals in proteins regardless of their position in the sequence. In addition, CNNs are able to train filters that detect short motifs in the input sequence irrespectively of where they occur, and have shown promising performance for protein subcellular localization when combined with LSTMs (Sønderby *et al.*, 2015). We also propose a hierarchical tree likelihood mimicking the biology of the sorting pathway and a transfer learning approach to jointly predict subcellular localization and whether the protein is membrane-bound or soluble.

In the following we discuss some of the caveats with the datasets used in previous subcellular localization tools. First, many methods use homology information for prediction, either by directly using annotated subcellular location annotations of retrieved hits in a database search, as in LocTree3 (with an accuracy of 80% for 18 locations) (Goldberg *et al.*, 2014), or by taking hints from other types of annotation such as GO-terms, as in iLoc-Euk and YLoc (Briesemeister *et al.*, 2010; Chou *et al.*, 2011), or PubMed abstracts linked to the protein's Swiss-Prot entry, as in SherLoc (Briesemeister *et al.*, 2009). These methods are appropriate for annotated proteins or proteins with annotated close homologues. Nonetheless, it should be taken into account that the performance will be much lower for sequences without well-annotated homologues—precisely the sequences for which it would be most relevant to have working prediction methods. In addition, any homology-based method will have very limited chance of being able to predict the consequences of mutations affecting sorting signals because the wild-type and the variant probably would pick up the same homologues in a database search.

Second, the performances of machine learning algorithms are crucially dependent on the datasets used to train and test them. For protein subcellular localization a key aspect is that proteins should have experimental evidence for their subcellular location, so that predictions are not based on predictions in a circular fashion. However, current methods use data from UniProt (The UniProt Consortium, 2017) prior to release 2014_09, where a major change in the annotation standards took place. Before the change, an annotation was regarded as experimental if it lacked qualifiers such as 'Potential', 'Probable' or 'By similarity'; after the change, only annotations with a specific literature reference were annotated as being experimental (evidence code ECO:0000269). This resulted in a considerable decrease in the number of proteins with subcellular location regarded as experimentally confirmed, thus raising the issue that current methods may in part be trained and tested on questionable examples.

Another aspect of the dataset issue is that the amount of homology between training data and test data should be kept at a minimum (Hobohm *et al.*, 1992). The measured test performance should be a true measure of the predictive performance on new proteins and not just a measure of how good the method is at finding homologues with the same subcellular location. Unfortunately, the Höglund dataset (Höglund *et al.*, 2006) which has been used in the training and test of several methods (Blum *et al.*, 2009; Briesemeister *et al.*, 2009, 2010; Shatkay *et al.*, 2007; Sønderby *et al.*, 2015) is only homology reduced to 80% identity. This means that rather close homologues to the training data will occur in the test set, which results in overly optimistic performances that do not reflect the true generalization to new unseen proteins. An example of a state-of-the-art method that uses this dataset set is Sherloc2, which reports an accuracy of 93% for 11 locations.

This paper has four major contributions:

1. We construct a new dataset from a recent version of UniProt where proteins have experimental evidence for their subcellular locations according to the new stricter definition. We perform stringent homology partitioning to avoid overfitting, providing realistic accuracy measures on new proteins.
2. We show that models trained on the Höglund dataset have poor generalization performance on our new dataset. This reflects the high level of homology and possibly erroneous annotations in the old dataset.
3. We develop deep recurrent neural networks for the protein subcellular localization task with a number of novel state-of-the-art model features. This includes convolutional motif detectors, selective attention on sequence regions important for subcellular localization prediction and a novel hierarchical sorting likelihood. These features are used for interpretation of the model and predictions. Our networks show improved prediction accuracy without using homology information.
4. We implement the resulting model as a user-friendly web-server called DeepLoc (Concurrently with our work, Kraus *et al.* (2017) has introduced a method for protein subcellular location from cell image data also called DeepLoc).

2 Materials and methods

2.1 Neural network models

The deep learning neural network model used is described in detail below. Figure 1 and the following description gives a summary of the architecture used: The input is sequence length (=1000) × size of amino acid vocabulary (=20). The CNN extracts motif information using 120 filters of different sizes (20 for each of the sizes 1, 3, 5, 9, 15 and 21). This gives a 1000 × 120 feature map. Another convolutional layer of 128 filters of size 3 × 120 is applied to this feature map. This gives a 1000 × 128 feature map which is used as input to the recurrent layer. The recurrent neural network scans the sequence using 256 LSTM units in both directions giving in total a 1000 × 512 dimensional output. The attention decoding layer uses an LSTM with 512 units through 10 decoding steps and the attention mechanism feedforward neural network (FFN) has 256 units. The final fully connected dense layer is composed by 512 and the two output layers have one unit (membrane-bound) and 10 units (subcellular localization).

We learn a subcellular localization model which predicts the subcellular localization using the amino acids sequence as input:

$$\boldsymbol{y} = f_{\theta}(\boldsymbol{X}), \tag{1}$$

where y is the predicted localization, f is the prediction model parametrized by parameters θ and $\boldsymbol{X}$ is the input data sequence of size $L \times N$ where L is the protein length and N is the number of input features per sequence position. The parameters θ are optimized using stochastic gradient descent with cross entropy loss between the true and predicted localization distribution.

In practice, the length of protein sequences can vary from tens to thousands of amino acids posing a challenge for many prediction algorithms requiring a fixed size input representation. Instead, recurrent neural networks (RNN) that naturally handle varying input

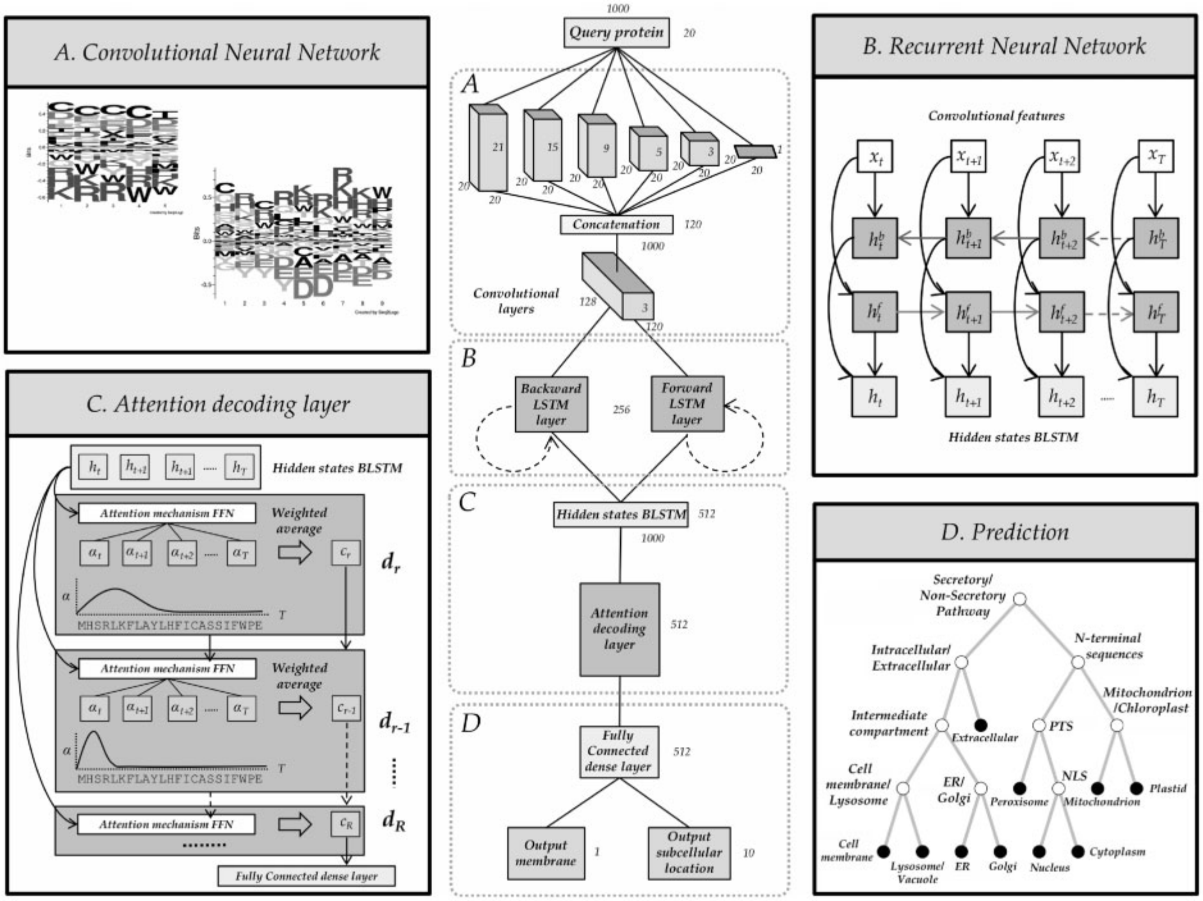

Fig. 1. (**A**) The convolutional neural network (CNN) extracts motif information using different motif sizes. (**B**) The recurrent neural network scans the sequence in both directions, extracting the spatial dependencies between amino acids. (**C**) The attention mechanism assigns higher importance to amino acids that are relevant for the prediction. At each decoding step, the attention weights α are generated based on the hidden states from the RNN and the hidden states from the previous decoding step. The weighted average of these weights at the last decoding step is used as input to a fully connected dense layer. (**D**) All the information gathered from the protein sequence is passed to a softmax function and a hierarchical tree of sorting pathways to calculate the final prediction

sequence lengths were used. The networks applies a recurrent calculation at each sequence position t

$$\boldsymbol{h}_t = f_{\mathrm{E}}(\boldsymbol{x}_t, \boldsymbol{h}_{t-1}), \;\; t = 1 \ldots L \tag{2}$$

where f_{E} is an RNN denoted the encoder, $\boldsymbol{x}_t$ is the input features of $\boldsymbol{X}$ at position t and $\boldsymbol{h} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_L]$ is the hidden states of the RNN where $\boldsymbol{h}_t$ is a vector of same length as the number of hidden units in the RNN. The encoder can be viewed as a trainable feature extractor encoding the amino acid sequence into a feature space suitable for subcellular localization prediction. Naively, the final subcellullar location $\boldsymbol{y}$ could be predicted by applying a classifier f_y to the final hidden state of the encoder $\boldsymbol{h}_L$

$$\boldsymbol{y} = f_y(\boldsymbol{h}_L)\,. \tag{3}$$

However, this approach is not ideal for several reasons. Firstly the RNN has to remember all useful information across the entire, often very long, input sequence. In subcellular localization this is especially problematic since most of the information is known to reside in the beginning (N-terminus) and end (C-terminus) of the sequence. Secondly all information about the protein has to be compressed into the same size vector regardless of the length of the protein. Two different solutions were used to alleviate these problems, Bidirectional RNNs and Attention RNNs. In bidirectional RNNs, the protein sequence is processed both forwards and backwards by two separate RNNs and the input to the final classifier is then the concatenated outputs of the last hidden state of both RNNs. The forwards and backwards RNNs will then be better at remembering motifs in the C-terminus and N-terminus respectively. Nevertheless, for long sequences these algorithms still have to remember information across many steps. To solve this problem, as well as identify protein regions important for classification, we augmented the bidirectional RNN encoder with an attentive decoder (Bahdanau *et al.*, 2014). Using the last hidden state of the encoder $\boldsymbol{h}_L$ as input the attentive decoder f_D is run for D decoding steps. Note that D does not depend on the input sequence length L. At each step, the hidden state of the attentive decoder $\boldsymbol{d}_r$ is used by an attention function f_A to assign a normalized importance weight to each sequence position of the encoder hidden states $\boldsymbol{h} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_L]$ as

$$\boldsymbol{d}_r = f_D(\boldsymbol{h}_L, \boldsymbol{d}_{r-1}, \boldsymbol{c}_{r-1}), \;\; r = 1 \ldots D \tag{4}$$

$$e_{t,r} = f_A(\boldsymbol{h}_t, \boldsymbol{d}_r) = \tanh(\boldsymbol{h}_t \boldsymbol{W}_e + \boldsymbol{d}_{r-1} \boldsymbol{W}_d)\boldsymbol{v}^T \tag{5}$$

$$\alpha_{t,r} = \frac{\exp\left(e_{t,r}\right)}{\sum\limits_{t'=1}^{L} \exp\left(e_{t',r}\right)}\,, \tag{6}$$

where $\boldsymbol{d}_r$ is the hidden state of the decoder at step r, matrices $\boldsymbol{W}_d$ and $\boldsymbol{W}_e$ and column vector $\boldsymbol{v}$ are the trainable parameters of the attention function. $\boldsymbol{d}_r$ is vector of same size as the number of hidden units in the decoder LSTM, which can be different from the dimensionality of the encoder $\boldsymbol{h}_t$. $\alpha_{t,r}$ is the normalized importance weights and $\boldsymbol{c}_r$ is a weighted average of the encoder RNN hidden states calculated as

$$\boldsymbol{c}_r = \sum_{t=1}^{L} \alpha_{t,r} \boldsymbol{h}_t . \tag{7}$$

The initial value of $\boldsymbol{c}_r$, i.e. $\boldsymbol{c}_0$, is a learned parameter vector that is trained as part of the neural network model. The subcellular localization is then predicted using the weighted average of the encoder RNN hidden states at the last step of the decoder

$$\boldsymbol{y} = f_y(\boldsymbol{c}_D) . \tag{8}$$

This allows the model to selectively assigns weight to sequence positions important for classification, which reduces the need for remembering all information across the entire length of the sequence. Both f_E and f_D are implemented as a special type of RNN units called Long-Short Term Memory (LSTM) cells (Hochreiter and Schmidhuber, 1997). LSTMs share the same chain structure as RNNs, but the recurrent calculation is augmented with an internal memory cell capturing long range dependencies.

Furthermore, convolutional filters were used to detect protein motifs. Here a filter, akin to position specific scoring matrices, is slid across the sequence. It will then detect a motif regardless of its position in the sequence. The weights of each filter can be adjusted to find the motifs that help to better predict each class. These new features created with a CNN can represent the inputs in a more abstract way, which, in combination with LSTMs, has been shown to be beneficial for protein classification (Sønderby *et al.*, 2015).

2.2 Hierarchical tree likelihood

To include information from protein sorting pathways into our model, a hierarchical tree with multiple nodes was developed. Each node represents a binary decision attempting to assign the protein to the right pathway from high-level to detailed classification. As an example, the first binary decision in the tree classifies proteins in the secretory or non-secretory pathway, whereas the last nodes separate related compartments such as mitochondria and chloroplasts, see Figure 1 panel D. The leaf nodes correspond to the final subcellular localizations, and the likelihood is calculated as the joint probability of decisions in the tree. So for example, if we have decisions A, B, y then according to the tree decomposition the probability of y given input sequence X is given by

$$P(y|X) = P(y|B,X)P(B|A,X)p(A|X) . \tag{9}$$

An example path is $A =$ Non-Secretory Pathway, $B =$ N-terminal Sequence and $y =$ Mitochondria. Each of the nine binary classifiers is implemented by a logistic output connected to the fully connected dense layer. By construction, the tree probabilities are normalized $\sum_y p(y|X) = 1$.

2.3 Datasets

2.3.1 DeepLoc dataset

The protein data used to train DeepLoc were extracted from the UniProt database, release 2016_04 (The UniProt Consortium, 2017). The protein dataset was filtered using the following criteria: eukaryotic, not fragments (they could have the N-terminal or C-terminal missing), encoded in the nucleus, longer than 40 amino acids and experimentally annotated (ECO:0000269). Similar locations or subclasses of the same location were mapped to 10 main locations in order to increase the number of proteins per compartment. Furthermore, proteins were classified as membrane or soluble if they were found on either the membrane or the lumen of the organelle; if no information was provided, they were tagged as unknown. Finally, proteins with more than one subcellular localization were filtered out. A total of 13 858 proteins were obtained after the filtering process. The mapped sublocations and the number of proteins in each main localization are summarized in Table 1.

To ensure that the model generalizes to new data a stringent homology partitioning was performed. Homologous proteins that fulfil a certain threshold of similarity were clustered as detailed below. Then, each cluster of homologous proteins was assigned to one of the five folds, ensuring that similar proteins were not mixed between the different folds. PSI-CD-HIT (Li and Godzik, 2006) was used to cluster proteins with 30% of identity or 10^{-6} E-value cutoff and the alignment must cover 80% of shorter (redundant) sequences, which produced 8410 clusters for the whole dataset. The five folds generated had approximately the same number of proteins in each location. Four were used for the training and validation and one heldout set for testing.

2.3.2 Höglund dataset

The Höglund dataset (Höglund *et al.*, 2006) have been used to train both the MultiLoc and RNN prediction methods in Höglund *et al.* (2006) and Sønderby *et al.* (2015). This dataset consist of 5959 proteins with 11 possible locations (cytoplasm, nucleus, extracellular, mitochondria, plasma membrane, ER, chloroplast, Golgi apparatus, lysosome, vacuole and peroxisome) and is homology reduced to 80% identity. Apart from grouping together lysosomal and vacuolar proteins no modifications were made to the dataset.

Table 1. Number of proteins in each location and sublocations that were grouped together under the same main location

Location	No. of proteins	Sublocations
Nucleus	4043	Envelope, inner and outer membrane, matrix, lamina, chromosome, nucleus speckle
Cytoplasm	2542	Cytoplasm (cytosol and cytoskeleton)
Extracellular	1973	Extracellular
Mitochondrion	1510	Envelope, inner and outer membrane, matrix, intermembrane space
Cell membrane	1340	Apical, apicolateral, basal, basolateral, lateral, cell membrane, cell projection
Endoplasmic reticulum (ER)	862	ER membrane and lumen, microsome, rough ER, smooth ER, Sarcoplasmic reticulum
Plastid	757	Plastid membrane, stroma and thylakoid
Golgi apparatus	356	Golgi apparatus membrane and lumen
Lysosome/Vacuole	321	Contractile, lytic and protein storage vacuole, vacuole lumen and membrane, lysosome lumen and membrane
Peroxisome	154	Peroxisome matrix and membrane

2.4 Comparison to current prediction algorithms

The performance of our models were compared with a number of current prediction algorithms using the following approaches: LocTree2 (Goldberg *et al.*, 2012), MultiLoc2 (Blum *et al.*, 2009) and SherLoc2 (Briesemeister *et al.*, 2009) were run with local command-line versions installed on our own server, while CELLO (Yu *et al.*, 2006), iLoc-Euk (Chou *et al.*, 2011) and WoLF PSORT (Horton *et al.*, 2007) were run on their web servers. YLoc (Briesemeister *et al.*, 2010) was run offline by the maintainer of the web service. Results for YLoc are given with the option to include GO terms turned on. For MultiLoc2 and SherLoc2, a newer version of InterProScan (5.21-60) was used instead of the recommended one (4.4) due to compatibility problems with the older version. As a reference the performance of Höglund test set was measured on our local installation obtaining an accuracy of 0.8300 for Multiloc2 and 0.9179 for SherLoc2.

In the cases where current methods predict more than ten locations, the predicted locations were mapped onto our ten locations. Two of the methods, iLoc-Euk and WoLF PSORT, in some cases predict dual locations (such as cytoplasm/nucleus). Since proteins with dual locations were filtered out in the construction of the dataset, those predictions were counted as erroneous, unless both the predicted locations mapped to the same location in our classification.

2.5 Experiments

Two different set of experiments were carried out. The first experiments were used for model selection comparing the relative performances of the following model architectures:

- Feedforward neural network (FFN)
- Bidirectional LSTM neural network (BLSTM)
- BLSTM neural network with attention mechanism (A-BLSTM)
- Convolutional BLSTM neural network with attention mechanism (Conv A-BLSTM)

Using the best model architectures the second set of experiments is designed to test the generalization performance of models trained on either our new DeepLoc dataset or the Höglund dataset.

Hyperparameters were optimized on three of four splits of the training data and the performance was evaluated on the last validation split. The hyperparameter selection was done using unidimensional search where one hyperparameter was changed and the rest were kept fixed. If a hyperparameter had not yet been tested, the median value in the range of that hyperparameter was chosen. Each hyperparameter setting was run for 150 epochs (epoch = full pass over the training set) and the performance was measured as the highest seen performance on the validation set. This strategy was used for computational reasons since a full grid search over all parameters was not computationally feasible. After the best hyperparameters were identified, a final run of experiments were used to identify the best combination of amino acid encodings among BLOSUM62 (Henikoff and Henikoff, 1992), sparse, protein profiles or HSDM encoding (Prlić *et al.*, 2000). We further found that protein profiles gave the highest performance and included these as input features for the final models. The profiles were generated using the same method as the TOPCONS web server (Tsirigos *et al.*, 2015).

The test performance was measured by training four models on the training set using the four different combinations of training and validation set. The reported test performance is the average of the four models evaluated on the held-out test set. We stress that we never optimized any parameters on the test set leaving the reported performances unbiased.

To decrease the training time, the maximum protein length was 1000. If a protein exceeded this length, amino acids from the middle of the sequence were removed in order to not to lose information about the N-terminal and C-terminal sorting signals. 9.98% of the proteins were truncated using this rule.

The performance measurements used to assess the performance of our models were accuracy and the Gorodkin measure (Gorodkin, 2004). For the binary prediction, the accuracy and the Matthew's Correlation Coefficient (Matthews, 1975) (MCC) were used. The Gorodkin measure can be seen as a generalization of MCC that applies to K-categories, which is more informative than the accuracy when there is an imbalance of classes. For K=2, the Gorodkin measure squared is the 'generalized squared correlation' (GC^2) of Baldi *et al.* (2000).

All models were implemented in Python 2.7.11 using the neural network library Lasagne 0.2 (Dieleman *et al.*, 2015) and Theano 0.9.0 (Theano Development Team, 2016) for efficient GPU implementation.

3 Results

We designed experiments to address the following questions:

- What are the relative performances of the proposed neural network model architectures? → Section 3.1
- How does the generalization performances of models trained on either the DeepLoc or Höglund datasets compare? → Section 3.2
- How does the final DeepLoc model compare to current state-of-the-art protein subcellular prediction models? → Section 3.3

3.1 Model selection

In Table 2 we compare the performances of different model architectures trained on the DeepLoc dataset. Note that we are interested in the relative performance of the models. Due to this, we only used BLOSUM62 encodings as input features, which resulted in a slightly degraded performance compared to the final performances described in the following sections.

The A-BLSTM and the CONV A-BLSTM models achieved the highest performance predicting the subcellular localization with accuracies of 0.7290 and 0.7289, respectively. Comparing these results with the performance of the BLSTM without attention (accuracy 0.6925), we see that attention improves performance. These results confirm the benefit of selective, context dependent, attention for protein classification. All of the A-BLSTM models performed significantly better than the baseline FFN model which achieved an accuracy of 0.5234. This is expected since FFN models do not take into account the order of the amino acids, whereas the LSTM models naturally consider the relationships between amino acids. Furthermore, we observed that including 10 decoding steps in the attention mechanism increased the accuracy (a difference of 1%) in

Table 2. Comparison of performances for different model architectures using BLOSUM62 input features

Model	Subcellular location		Membrane	
	Accuracy	Gorodkin	Accuracy	MCC
FFN	0.5234	0.4229	0.7301	0.4509
BLSTM	0.6925	0.6278	0.9004	0.8023
A-BLSTM	0.7290	0.6729	0.9163	0.8345
CONV A-BLSTM	0.7289	0.6780	0.9111	0.8218

comparison with a single decoding step. Increasing the decoding steps beyond 10 resulted in a reduction in the accuracy. Lastly, the A-BLSTM models predicted whether the proteins were membrane-bound or soluble with accuracies of 0.9163 and 0.9111 respectively.

From the amino acid encoding comparison, we found that the CONV A-BLSTM model using protein profiles encoding had the highest accuracy, with a difference of 2% compared to the A-BLSTM model. Therefore, we decided to use this encoding and this model for the rest of the experiments.

3.2 Dataset comparison

To compare the generalization performance of models trained on either the DeepLoc or the Höglund datasets, we trained a CONV A-BLSTM model on each dataset and evaluated the performances on the test sets from both datasets. Table 3 shows that (i) the Höglund training set achieves a good test performance only on the Höglund test set and (ii) the DeepLoc training set achieves a good test performance on test sets with stringent independence between training and test sets.

These results show that models trained on the Höglund dataset generalize poorly compared to models trained on the DeepLoc dataset. As a qualitative comparison of the two datasets we visualized the context vectors $\boldsymbol{c}_r$ for CONV A-BLSTM models trained on both datasets as seen in Figure 2. The compartments are notably more separated for the model trained on the Höglund dataset compared to the model trained on the DeepLoc dataset

3.3 DeepLoc model

From the model comparisons we identified the CONV A-BLSTM as the best performing model architecture. To further improve prediction accuracy we trained an ensemble of 16 models using nested cross validation. Eight of the models were trained using a softmax output distribution (class probability from softmax function) and eight of the models using the hierarchical tree distribution (joint probability of multiple logistic functions). Further we mitigate the effect of the class imbalances by using a cost matrix (Zhou and Liu, 2006) to recalculate the class probabilities based on the number of samples in the training set. The full ensemble achieved an accuracy of 0.7797 and Gorodkin of 0.7347 on the subcellular localization and an accuracy of 0.9234 and a MCC of 0.8435 on the membrane-bound or soluble prediction. We found that the softmax models had a slightly higher accuracy than the hierarchical tree model with the 8-ensembles achieving an accuracy of 0.7717 and 0.7695, respectively. We show in Table 4 the accuracy and the MCC for each binary decision in the hierarchical tree model. We experimented with increasing the ensemble size but found no improvement in performance.

The training time for the full ensemble was 80 hours, approximately five hours per model. When testing, the ensemble takes three seconds per protein on average to perform a prediction. Nonetheless, this ensemble used protein profiles, which were already generated for this dataset. This profile generation is the most time-consuming step usually taking approximately 30 seconds per protein. If a hit with the PFAM database is not found the profile generation uses Uniref90 instead. This can take even longer and therefore can be problematic for large protein datasets. To solve this we trained the same ensemble using BLOSUM62 encoding. This model has an accuracy of 0.7360 and Gorodkin of 0.6832 on the

Table 3. Comparison of generalization performances using the CONV A-BLSTM model between the DeepLoc dataset and the Höglund dataset

Training set	Test set	Accuracy	Gorodkin
DeepLoc	DeepLoc	0.7511	0.6988
Höglund	DeepLoc	0.6426	0.5756
DeepLoc	Höglund	0.8301	0.8010
Höglund	Höglund	0.9138	0.8979

Note: Sequence profiles were used as input features.

Table 4. Accuracy and MCC of each node in the hierarchical tree

Node	Accuracy	MCC
Secretory/Non-secretory pathway	0.9502	0.8902
Intracellular/Extracellular	0.9507	0.8979
N-terminal sequences	0.9544	0.8784
Intermediate compartment	0.7982	0.5824
PTS	0.9784	0.4085
Mitochondrion/Chloroplast signals	0.9537	0.8955
Cell membrane/Lysosome	0.8575	0.5002
ER/Golgi	0.8559	0.6376
NLS	0.8138	0.6031

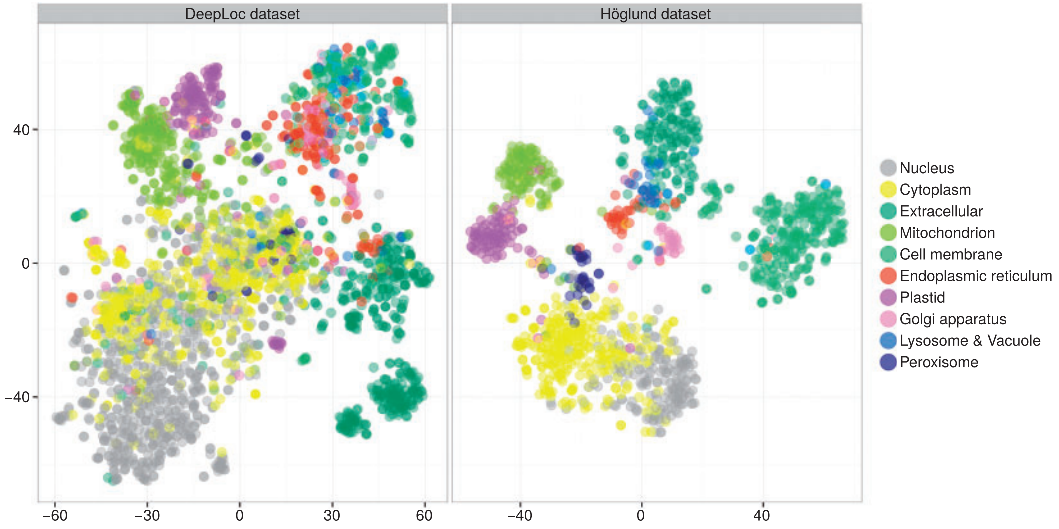

Fig. 2. t-SNE representation of the context vector c_r for a Conv A-BLSTM trained on the DeepLoc and Höglund dataset and visualized for the respective test sets

Table 5. Confusion matrix of the test set on the final DeepLoc model using profiles encoding

Location	Number of predicted proteins										Sens.	MCC
Nucleus	680	103	4	5	2	8	1	2	2	1	0.842	0.784
Cytoplasm	94	361	7	18	5	4	3	8	1	7	0.711	0.608
Extracellular	3	5	365	5	5	4	2	0	4	0	0.929	0.907
Mitochondrion	9	21	0	247	0	5	14	2	1	3	0.818	0.812
Cell membrane	5	15	6	1	203	20	1	4	18	0	0.744	0.732
Endoplasmic reticulum	3	6	6	3	18	120	1	7	8	1	0.694	0.654
Plastid	1	2	0	8	0	0	140	0	1	0	0.921	0.883
Golgi apparatus	4	17	1	0	9	8	1	26	4	0	0.371	0.414
Lysosome/Vacuole	0	7	11	1	20	9	0	4	12	0	0.188	0.194
Peroxisome	0	13	0	4	1	4	0	0	0	8	0.267	0.321

Note: Sens., sensitivity.

Table 6. Confusion matrix for the membrane-bound predictor

Type	Number of predicted proteins	
Soluble	968	38
Membrane-bound	96	647

subcellular localization and an accuracy of 0.9130 and a MCC of 0.8237 on the membrane-bound or soluble prediction. By omitting the profile generation, we achieved a faster prediction at the cost of decrease in accuracy.

Tables 5 and 6 show the confusion matrices of the full ensemble described above for subcellular localization and membrane-bound prediction respectively. The primary sources of error are confusion of the nucleus and cytoplasm, lysosome/vacuole misclassified as cell-membrane and Golgi misclassified as cytoplasm. In Figure 3, we show the attention vector α, i.e. how important different regions of the sequence are for the classification. In general, the DeepLoc model assigns large importance to the N-terminal for secreted proteins whereas e.g. membrane proteins have regions of importance interspersed across the protein length.

To compare the performance of the final DeepLoc model to other approaches we benchmarked a number of current prediction algorithms on the DeepLoc test set as seen in Table 7. The accuracy of the final DeepLoc model (0.7797) is significantly better than all other methods with iLoc-Euk achieving the second best accuracy of 0.6820.

4 Discussion

In this paper we have introduced the DeepLoc dataset: a well assembled protein collection with reliable subcellular localization information. Secondly we have provided a deep neural network based prediction algorithm achieving state-of-the-art performance on this new dataset. The context-dependent annotation vector generated by the attention mechanism is able to represent a protein based on its subcellular localization. In addition, the attention based prediction method allows visualization of the biologically plausible regions used to predict the subcellular localization of the proteins which we believe will provide relevant information.

The comparison of the generalization performances for models trained on our new DeepLoc dataset and the Höglund dataset showed that DeepLoc trained models generalized much better than the Höglund trained model. Here we discuss a number of explanations for these findings. Firstly, with the Uniprot database change the Höglund dataset could contain many wrongly annotated proteins, which generates a model that learns to predict the wrong labels. Secondly, the homology reduction threshold 80% used for constructing the Höglund dataset might not be stringent enough, since it produces similar training and test examples.

In Figure 2, we compared the attention context vector for models trained on either the DeepLoc or Höglund datasets. For the Höglund trained model, all locations are almost perfectly separated implying that there is little variation within the Höglund dataset classes and that the training and test sets are relatively similar. This supports the finding of poor generalization performance for models trained on the Höglund dataset. Hence, we believe that the high performance reported for algorithms trained on this dataset is actually results from overfitting. The true variation within each protein class is larger as indicated by the better generalization performance for models trained on the DeepLoc dataset. This is further corroborated by the poorer separation of classes for the DeepLoc trained models in the same figure.

We compared the performance of the final DeepLoc model with other current prediction algorithms in Table 7. We found that the DeepLoc model performs significantly better than the other approaches. Here we note that the DeepLoc performance is a true test set performance, whereas the performances of the other methods may be overestimated since some sequences in our test set may have been included in their training sets. Further we emphasize that the DeepLoc method is a purely sequence-based method and does not rely on annotation information from homologous proteins. Due to the stringent homology partitioning applied in the dataset construction, the model should generalize to new proteins without known close homologues.

We note that we also compared the performance against the LocTree3 prediction method (Goldberg *et al.*, 2014), which is a combination of LocTree2 and a BLAST search of a database of proteins with known subcellular location. However, as 75% of the proteins in the DeepLoc test set are also in the LocTree3 BLAST database, the measured accuracy was artificially high at 91%, since LocTree3 simply retrieves the same subcellular location used for labelling our test set.

The compartment specific prediction performance of the final DeepLoc model is shown in Table 5. The main source of error is the low performance on the Golgi apparatus, lysosome/vacuole and peroxisome. One possible cause is the low number of samples used to train these classes. However, this finding could also be associated with the similarity between the proteins from these locations and other compartments. For example, Table 5 shows that the lysosome/vacuole is usually misclassified as cell membrane and the peroxisome as cytoplasm.

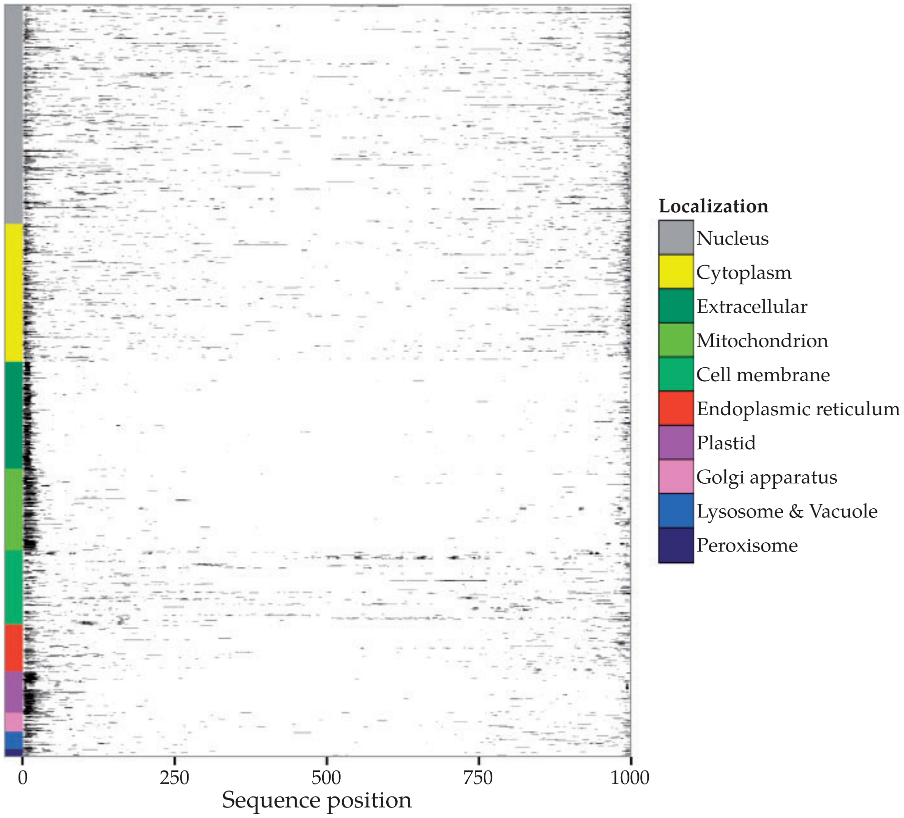

Fig. 3. Sequence importance across the protein sequence of DeepLoc test set when making the prediction. The *x*-axis is the sequence position and alomg the *y*-axis we have the proteins in the test set sorted according to protein localization. For visualization proteins shorter than 1000 amino acids are padded from the middle, so the N-terminus and C-terminus align. Proteins longer than 1000 amino acids have the middle part removed

Table 7. Accuracy and Gorodkin measure achieved by current predictors and the final DeepLoc model on the DeepLoc test set

Method	Accuracy	Gorodkin
LocTree2	0.6120	0.5250
MultiLoc2	0.5592	0.4869
SherLoc2	0.5815	0.5112
YLoc	0.6122	0.5330
CELLO	0.5521	0.4543
iLoc-Euk	0.6820	0.6412
WoLF PSORT	0.5671	0.4785
DeepLoc	**0.7797**	**0.7347**

The highest scores are shown in bold.

In addition to the mentioned under-represented classes, proteins from the cytoplasm and nucleus are also difficult to differentiate (Fig. 2, Table 4) because they both lack N-terminal sorting signals. The only difference between them is the nuclear localization signal (NLS), which is a highly variant short sequence that can be located in multiple regions of the protein sequence, making it hard to recognize. Figure 3 shows the positions in the sequence that the attention mechanism focuses on to generate the attention context vector c_r. For the nucleic and cytoplasmic proteins, the model focuses on the beginning of the sequence (checking for the absence of an N-terminal sorting signal). Moreover, the model also gives importance to small regions across the sequence. The main difference is that there is a higher density of these regions in nucleus examples than in cytoplasm, which could indicate that the model is able to identify some of the most represented NLS.

Figure 3 allows us to visualize what regions in the sequence are relevant for each subcellular localization to perform the prediction. For the extracellular proteins, the model focuses mainly on the signal peptide, which can be seen as a small region at the N-terminus of the sequence. In contrast, the attention is scattered across the sequence for plasma membrane proteins, which could indicate that the algorithm is detecting the transmembrane helices. For the ER proteins we can see attention at the N-terminus, where the signal peptide is located, and also some attention at the C-terminus, which could mean the presence of KDEL or KKXX signals. Golgi proteins have the importance on the N-terminus slightly shifted to the right, in comparison with other proteins from the secretory pathway, as they are mostly type II transmembrane proteins with signal anchors. Mitochondrial and chloroplastic proteins have large regions at the N-terminus, which clearly correlates to the mitochondrial and chloroplastic transit peptides. The lysosomal/vacuolar proteins do not seem to have a clear important region across their sequences. Finally, for peroxisomal proteins, some regions at the N-terminus and at the C-terminus are observed, which could mean that the model is detecting PTS2 and PTS1 signals.

5 Conclusion

We have shown that convolutional BLSTM neural networks with attention mechanism are able to accurately predict the protein subcellular localization and if a protein is membrane-bound or soluble just using the sequence information. Further we have introduced the DeepLoc dataset. The DeepLoc model trained on this dataset is able to generalize better than using previous datasets for subcellular localization. In addition, DeepLoc obtained the highest accuracy using the independent test set, when compared with the current methods.

There are several perspectives of this project that we would like to pursue in the future. One of those is to make better use of existing knowledge about sorting signals. DeepLoc 1.0 is trained in a relatively 'naive' way, where the networks have been provided only with protein profiles and their location labels. It would be beneficial to explicitly model known sorting signals such as N-terminal signal peptides and transit peptides.

In addition, it should be investigated whether performance can be enhanced by training several models with a narrower taxonomical scope instead of treating all eukaryotes by one model. Obviously, animals and fungi do not have plastids, and some false predictions could be avoided by disallowing plastid predictions for these groups, but more subtle differences between sorting signals are also known to exist. However, there is a trade-off between the precision of the taxonomical scope and the sizes of the training datasets. For taxonomic groups with limited numbers of data with experimentally known subcellular location, it may be necessary to employ semisupervised learning, where unlabelled data from genome sequences are used along with labelled data.

Acknowledgements

The authors wish to thank Konstantinos Tsirigos and Arne Elofsson of Stockholm University for permission to use their fast profile construction method in DeepLoc, even though it has not been published yet. In addition, they want to thank Fabian Aicheler of University of Tübingen for kindly running the DeepLoc test set on YLoc.

Funding

S.K.S. and O.W. were supported by a grant from the Novo Nordisk Foundation and the NVIDIA Corporation with the donation of TITAN X GPUs.

Conflict of Interest: none declared.

References

Bahdanau,D. *et al.* (2014) Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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

Blum,T. *et al.* (2009) Multiloc2: integrating phylogeny and gene ontology terms improves subcellular protein localization prediction. *BMC Bioinformatics*, **10**, 1.

Briesemeister,S. *et al.* (2009) Sherloc2: a high-accuracy hybrid method for predicting subcellular localization of proteins. *J. Proteome Res.*, **8**, 5363–5366.

Briesemeister,S. *et al.* (2010) YLoc–an interpretable web server for predicting subcellular localization. *Nucleic Acids Res.*, **38**, W497–W502.

Chou,K.-C. *et al.* (2011) iLoc-Euk: a multi-label classifier for predicting the subcellular localization of singleplex and multiplex eukaryotic proteins. *PLoS ONE*, **6**, e18258.

Dieleman,S. *et al.* (2015) *Lasagne: First Release*. Geneva, Switzerland, Zenodo.

Emanuelsson,O. *et al.* (2007) Locating proteins in the cell using TargetP, SignalP and related tools. *Nature Protoc.*, **2**, 953–971.

Goldberg,T. *et al.* (2012) LocTree2 predicts localization for all domains of life. *Bioinformatics*, **28**, i458–i465.

Goldberg,T. *et al.* (2014) Loctree3 prediction of localization. *Nucleic Acids Res.*, **42**, W350–W355.

Gorodkin,J. (2004) Comparing two k-category assignments by a k-category correlation coefficient. *Comput. Biol. Chem.*, **28**, 367–374.

Henikoff,S. and Henikoff,J.G. (1992) Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA*, **89**, 10915–10919.

Hobohm,U. *et al.* (1992) Selection of representative protein data sets. *Protein Sci.*, **1**, 409–417.

Hochreiter,S. and Schmidhuber,J. (1997) Long short-term memory. *Neural Comput.*, **9**, 1735–1780.

Höglund,A. *et al.* (2006) Multiloc: prediction of protein subcellular localization using N-terminal targeting sequences, sequence motifs and amino acid composition. *Bioinformatics*, **22**, 1158–1165.

Horton,P. *et al.* (2007) WoLF PSORT: protein localization predictor. *Nucleic Acids Res.*, **35**, W585–W587.

Hung,M.-C., and Link,W. (2011) Protein localization in disease and therapy. *J. Cell Sci.*, **124**, 3381–3392.

Imai,K. and Nakai,K. (2010) Prediction of subcellular locations of proteins: where to proceed? *Proteomics*, **10**, 3970–3983.

Kraus,O.Z. *et al.* (2017) Automated analysis of high-content microscopy data with deep learning. *Mol. Syst. Biol.*, **13**, 924.

Li,W. and Godzik,A. (2006) Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, **22**, 1658–1659.

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

Prlić,A. *et al.* (2000) Structure-derived substitution matrices for alignment of distantly related sequences. *Protein Eng.*, **13**, 545–550.

Shatkay,H. *et al.* (2007) Sherloc: high-accuracy prediction of protein subcellular localization by integrating text and protein sequence data. *Bioinformatics*, **23**, 1410–1417.

Sønderby,S.K. *et al.* (2015) Convolutional LSTM networks for subcellular localization of proteins. In: *International Conference on Algorithms for Computational Biology*, volume 9199 of *Lecture Notes in Computer Science*, pp. 68–80. Springer.

The UniProt Consortium (2017) UniProt: the universal protein knowledgebase. *Nucleic Acids Res.*, **45**, D158–D169.

Theano Development Team (2016) Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688.

Tsirigos,K.D. *et al.* (2015) The TOPCONS web server for consensus prediction of membrane protein topology and signal peptides. *Nucleic Acids Res.*, **43**, W401–W407.

Wan,S. and Mak,M.-W. (2015) *Machine Learning for Protein Subcellular Localization Prediction*. De Gruyter, Berlin, Germany.

Yu,C.-S. *et al.* (2006) Prediction of protein subcellular localization. *Proteins*, **64**, 643–651.

Zhou,Z.-H. and Liu,X.-Y. (2006) Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Trans. Knowledge Data Eng.*, **18**, 63–77.