Learning protein subcellular localization patterns from peptide sequence

Chung-ha Davis and Olivia Gautier
Neurosciences Graduate Program
Stanford University
chungha@stanford.edu, ogautier@stanford.edu

Abstract

Appropriate subcellular localization of proteins is critical for cell function, and previous work has shown that peptide sequence regulates protein localization. A Long short-term memory network was trained on peptide sequence data along with with subcellular localization labels from the Human Protein Atlas[1]. Class imbalance of the data set was addressed by training set resampling and class weighting. Successful training of LSTM models with peptide sequence data did not improve the validation set evaluation metrics, suggesting that more data may be needed to address this complex classification problem.

1 Introduction

One fundamental aspect of proper protein function in a cell is its subcellular location. In a cell, proteins are directed to distinct subcellular compartments in order to perform their appropriate functions, and failures in protein localization are evident in diseases such as cancer, cardiovascular disease, and neurological disorders [2, 3]. Proteins sequence is specified by the genome, where mRNA is transcribed from genomic DNA, and proteins are translated from mRNA. A functional protein is composed of multiple peptide domains that perform functions including structure, intramolecular interactions, intermolecular interactions, enzymatic activity, and localization. Distinct peptide sequences that regulate localization to subcellular organelles, including the nucleus and the endoplasmic reticulum, have been studied in the past [4, 5].

In the present study, We aim to establish a deep learning-based paradigm for learning peptide sequences that direct protein localization. Success in this aim would be relevant for the fields of cell biology, protein engineering, and synthetic biology. Since proteins are composed of a linear sequence of peptides with multiple domains of diverse functions, we reasoned that a recurrent neural network (RNN) architecture is appropriate for such a task. In particular, a long short-term memory network (LSTM) architecture [6], which is capable of learning long-term dependencies in sequence data, was chosen since we hypothesized that subcellular localization is governed by peptide sub-sequences throughout the length of the protein. The input to our algorithm is a peptide sequence of a protein. Then we use our trained LSTM network to output a predicted subcellular location.

2 Related work

Inference of protein subcellular localization from peptide sequence is a long-studied topic. Prior to the deep learning era, expert systems [7] and support vector machines (SVM) [8] were used to

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

predict subcellular localization. In more recent years, several deep learning-based approaches were utilized. Almagro Armenteros et al. utilized a LSTM architecture [9], Pang et al. used combination of a convolutional neural network (CNN) architecture and eXtreme Gradient Boosting (XGBoost) [10] and Wei et al. used stacked autoencoder (SAE) networks [11].

At a high level, while RNN-based implementations contain more intelligible hidden features, they are more complex with more hyperparameters to tune. In contrast, the CNN and SAE-based architectures are capable of extracting expressive feature representations, but these hidden representations are more difficult to interpret. Because we felt that model explainability is an important aspect of our endeavor, we opted to use an RNN-based approach. In contrast to the previous LSTM-based implementation by Almagro Armenteros et al., we utilize the Human Protein Atlas dataset instead of manually curating our own dataset, as the dataset constructed by Almagro Armenteros et al. was reported to have potentially produced optimistic predictions [10].

While not specifically addressing the question of subcellular localization, the approach of Alley et al. may prove to be the most forward-looking and state-of-the-art of the currently published methods for protein sequence feature extraction. Alley et al. utilized a LSTM-based architecture for deep representation learning that distills protein features that are grounded by structure, biophysics and evolution [12]. Such unpersupervised, data-driven approaches may prove to be instrumental in the study of subcellular localization in the future.

3 Dataset and Features

For this study, we utilized a publicly available dataset from the Human Protein Atlas [1], which contained a list of human proteins along with the proteins' subcellular locations. This dataset was constructed by first imaging various proteins within cells using antibody-based profiling and immunofluorescence confocal microscopy. The final unprocessed dataset from the Cell Atlas consists of gene names and corresponding protein localization information for 12,390 genes (63% of the human genome) for which there are available antibodies. Subsequently, the images were used to classify proteins into one or more of 32 different categories consisting of organelles and fine subcellular structures [1]. The 32 categories were collapsed to 13 Protein localization classes according to the class hierarchy established in [1]. The class distribution is uneven (Figure 1A). We only used proteins with a single main location, so that multi-class, as opposed to multi-label, classification could be performed.

We extracted peptide sequences from a separate publicly available database, Ensembl [13]. In instances where there were multiple protein annotations per gene, the longest peptide sequence representing the gene was chosen. The peptide sequences were integer-encoded, and because the sequences are of varying peptide lengths of different proteins, we performed zero padding. These integer-encoded sequences served as input into our learning algorithm to predict protein localization (Figure 1B). All data pre-processing was performed with custom Python and Bash code.

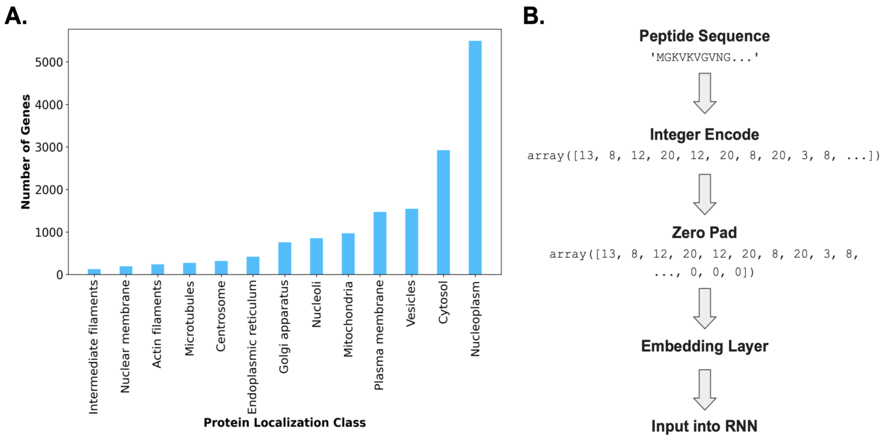

Figure 1: Data labels and model input. (A) Bar plot of all genes in the dataset grouped by the protein localization class. (B) Schematic of peptide sequence pre-processing and input into the model.

4 Methods

Because the model input is sequence data, we chose to use an LSTM neural network architecture (Figure 2). We split the data into training, dev, and test sets (80%, 10%, 10%). After peptide sequence pre-processing and embedding (Figure 1B), sequences were passed into an LSTM layer, and output from the LSTM layer was in turn passed to a fully connected layer. The softmax activation function was applied to the output of the fully connected layer, and a class prediction was made. Categorical cross entropy was used as the loss function for this multi-class classification problem (Equation 1), and the Adam optimizer was used in training the model. The numerical computing library NumPy [14], deep learning frameworks TensorFlow [15] and Keras [16], and the plotting library Matplotlib [17] were used in data pre-processing, implementation, and visualization of our model.

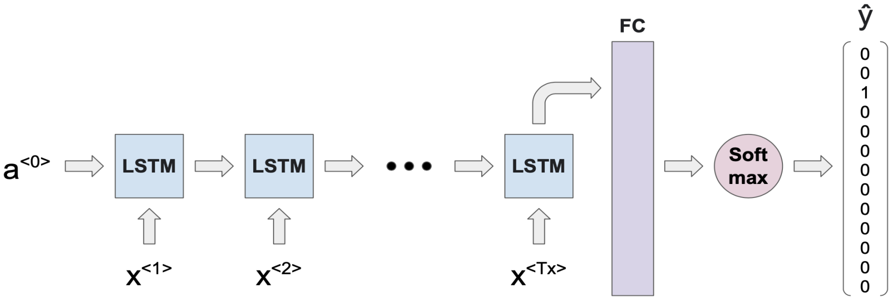

Figure 2: Model architecture.

$$\text{Cross Entropy Loss} = -log\left(\frac{e^{s_p}}{\sum_j^C e^{s_j}}\right) \tag{1}$$

Initial training of the LSTM model did not reduce the training set loss function after ten epochs (Figure 4A). We reasoned that model training may have been impaired because the classes are highly imbalanced. To test this idea, we randomly upsampled rare classes and downsampled abundant classes with replacement. The upsampling and downsampling resulted in 1200 training examples per class, where 1200 is approximately the mean number of examples per class in the original training set. Upsampling and downsampling with this strategy resulted in a training loss that decreases and training evaluation metrics that increase over 10 epochs (Figure 4B).

To adjust for upsampling and downsampling, we applied class weights to upweight the downsampled classes and downweight the upsampled classes. Training this model resulted in decreased training loss and increased evaluation metric scores as a function of epoch number (Figure 5A). The high evaluation metric scores on the training data show that the model has low bias. However, loss on the validation/dev set did not decrease, and evaluation metrics did not increase with epoch number (Figure 5A), suggesting high variance. An attempt to decrease model variance with dropout regularization (Figure 5B) did not prove to be successful.

5 Experiments/Results/Discussion

Ultimately, we were not successful in achieving the desired results of 1) Successfully training a LSTM that satisfactorily classifies validation examples, and 2) analyzing the hidden features of a successfully trained network to recover already-known and novel protein localization sequences. Merely obtaining a model that correctly classifies the training data took much effort, which will be discussed in the following sections.

5.1 Model Design Considerations and Runtime Bottleneck

Our ground-truth data consisted of full-length protein sequences and localization labels. Because we wanted out model to identify peptide motifs within full-length protein sequences, we opted against training a model on smaller, truncated subsequences. We reasoned that some, if not most, subsequences would not contain peptide motifs that are relevant for the subcellular localization label. As a result, we opted to use the longest-length protein length (length 6907) as the length of our input examples, and we zero-padded all smaller examples. Based on the distribution of protein lengths (Figure 3), one opportunity for decreasing space complexity in our efforts would be to exclude larger proteins (greater than length 2000) from our model.

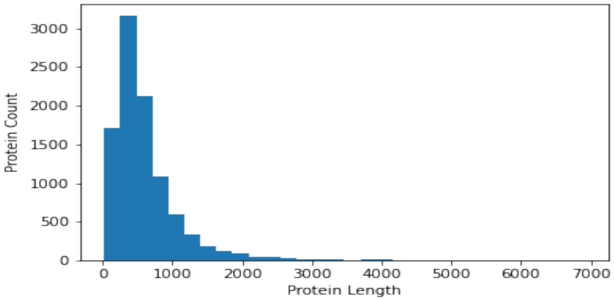

Figure 3: Distribution of protein length

5.2 Hyperparamter Choice and Evaluation Metrics

We used a mini-batch size of 128 to stay within our allotted GPU memory (24GB). The default learning rate was sufficient for model training. For evaluation metrics, we monitored prediction accuracy and area under the precision-recall curve (AUPRC).

5.3 Addressing Class Imbalance by Resampling

One challenge that we encountered during this study was getting our model to fit to our training data given the class imbalance. In order to address class imbalance, we resampled our training data so that each class was represented equally, and we modified class weights to counterbalance our resampling procedure. Prior to resampling, we were not able to observe loss decrease or model performance increase with accuracy and AUPRC through 10 training epochs (Figure 4A). Following resampling, we were able to observe decrease in loss and increase in model performance for the training data set, while the dev set did not show a corresponding increase in performance(Figure 4B). We concluded that we overfit to the training data and that we needed to address variance problems with our model.

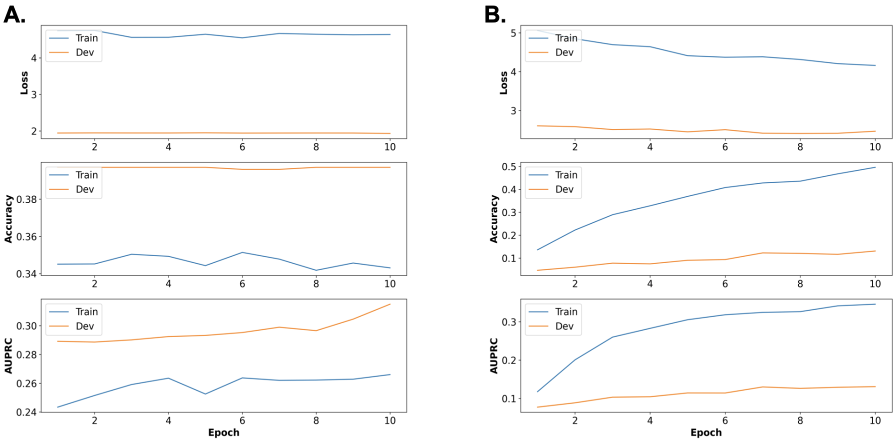

Figure 4: LSTM model without (A) and with upsampling/downsampling (B).

5.4 Dropout Regularization

After observing gains in model performance on the training data, we attempted to address the issue of variance by dropout regularization on the fully connected layer. Without dropout regularization, the dev set did not show an increase in performance as measured by accuracy and AUPRC (Figure 5A), and with dropout regularization, we similarly did not observe an increase in performance by accuracy and AUPRC (Figure 5B). AUPRC did decrease for the training set with regularization (Figure 4A and 4B, last row), which suggests that the regularization was taking effect.

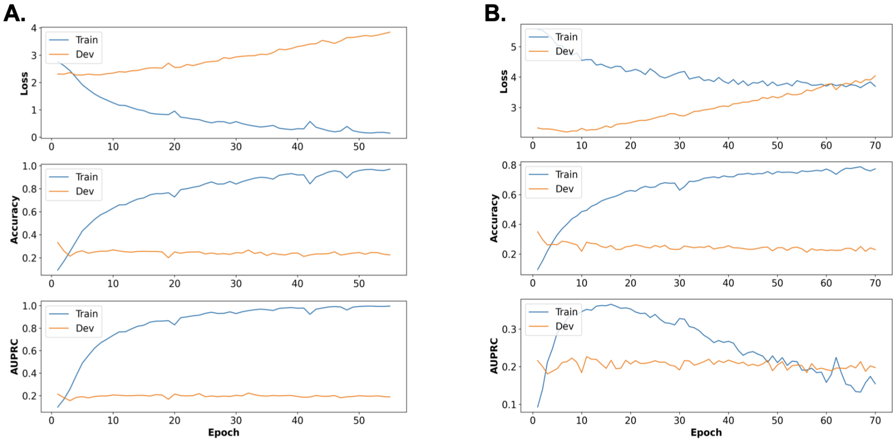

Figure 5: LSTM model without (A) and with dropout regularization (B).

6 Conclusion/Future Work

In conclusion, this study reveals that an-LSTM-based model is a promising candidate for learning peptide features that determine subcellular localization. Additional hyperparameter tuning of the model and more training data would likely lead to performance gains. With a better-performing model, we hope to analyze hidden features in the embedding layer in order to gain a deeper understanding of how peptide sequences contribute to subcellular localization.

7 Contributions

Olivia Gautier and Chung-ha Davis contributed equally to: Project conception, data pre-processing, training network, evaluating results, literature research, and writing final report.

References

[1] Thul PJ, Åkesson L, Wiking M, Mahdessian D, Geladaki A, Ait Blal H, Alm T, Asplund A, Björk L, Breckels LM, Bäckström A, Danielsson F, Fagerberg L, Fall J, Gatto L, Gnann C, Hober S, Hjelmare M, Johansson F, Lee S, Lindskog C, Mulder J, Mulvey CM, Nilsson P, Oksvold P, Rockberg J, Schutten R, Schwenk JM, Sivertsson Å, Sjöstedt E, Skogs M, Stadler C, Sullivan DP, Tegel H, Winsnes C, Zhang C, Zwahlen M, Mardinoglu A, Pontén F, von Feilitzen K, Lilley KS, Uhlén M, Lundberg E. (2017) A subcellular map of the human proteome. *Science* **356(6340)**:eaal3321.

[2] Wang X, Li S. (2014) Protein mislocalization: mechanisms, functions and clinical applications in cancer. *Biochim Biophys Acta* **1846(1):** 13–25.

[3] Hung M, Link W. (2011) Protein localization in disease and therapy. *Journal of Cell Science* **124:** 3381-3392.

[4] Bauer NC, Doetsch PW, Corbett AH. (2015) Mechanisms Regulating Protein Localization. *Traffic* **16(10):** 1039-1061.

[5] Dönnes P, Höglund A. (2004) Predicting Protein Subcellular Localization: Past, Present, and Future *Genomics, Proteomics Bioinformatics* **2(4):** 209-215.

[6] Hochreiter S, Schmidhuber J. (1997) Long Short-Term Memory *Neural Computation* **9(8):** 1735-1780.

[7] Nakai K, Kanehisa M. (1991) Expert system for predicting protein localization sites in gram-negative bacteria *Neural Computation* **11(2):** 95-110.

[8] Tung C, Chen C, Sun H, Chu Y. (2017) Predicting human protein subcellular localization by heterogeneous and comprehensive approaches *Plos One* **12(6):** e0178832

[9] 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), pp.3387-3395.

[10] Pang L, Wang J, Zhao L, Wang C, Zhan H. (2019) A Novel Protein Subcellular Localization Method With CNN-XGBoost Model for Alzheimer's Disease *Frontiers in Genetics* **9(751):** doi: 10.3389/fgene.2018.00751

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

[12] Alley EC, Khimulya G, Biswas S, AlQuraishi M, Church GM. (2019)Unified rational protein engineering with sequence-based deep representation learning. *Nature Methods* **16:** 1315–1322.

[13] Cunningham F, Achuthan P, Akanni W, Allen J, Amode MR, Armean IM, Bennett R, Bhai J, Billis K, Boddu S, and Cummins C. (2019) Ensembl 2019. *Nucleic acids research*, **47(D1)**, D745-D751.

[14] Harris CR, Millman KJ, van der Walt SJ, et al. (2020) Array programming with NumPy. *Nature* **585:** 357–362.

[15] Abad, M, et al. (2016) Tensorflow: A system for large-scale machine learning. *In 12th Symposium on Operating Systems Design and Implementation* **16:** 265–283.

[16] Chollet F, et al. (2015) Keras. Available at: https://github.com/fchollet/keras

[17] Hunter JD. (2007) Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering* **9(3):** 90-95.