Consensus Sequence Prediction With LSTMs - Submission to PLOS Synbio Review

Michael Teti¹, Rachel St Clair², Abrian Miller², William Hahn³, Elan Barenholtz³, with the Center for Complex Systems and Brain Sciences

1 Florida Atlantic University, Boca Raton, FL, United States 33431 USA

Abstract

Bioinformatics can be immensely valuable with the aid of machine learning algorithms to process and analyze data. In this work, we use TensorFlow, an open-source machine learning library, to implement a Long Short-Term Memory (LSTM) neural network on the task of categorizing proteins based on their primary amino acid sequence. The network was employed on two separate datasets — homeodomain vs. non-homeodomain, and artemisinin binding vs. non-artemisinin binding — to show its ability to generalize. The first dataset containing proteins with one of two categories (binding or non-binding to a target antimalarial drug Artemisinin) were given as input to the LSTM, which learned to recognize characteristic patterns within the sequences of the overall class, accurately identify the proper category of a novel sequence, and predict an unknown sequence’s binding ability with high accuracy. The second dataset involved proteins containing and not containing homeodomain consensus sequence. The model was again trained to distinguish novel features of the two categories (homeodomain and no homeodomain) and classify a novel sequence accurately. When the network was prompted to define the feature most activating nodes for proteins in the consensus containing class, the model predicted a sequence highly similar to the theoretically accepted homeodomain consensus. This computational model supports the use of LSTM’s in proteomics and should be considered for use in a wide array of research dataset types to analyze, sort, and predict copious amounts of information.

LSTM — Proteins — Machine Learning — Tensorflow

Introduction

As bioinformatics data grows exponentially, our need for collecting, sorting, and analyzing it becomes an imperative aspect of this research, and methods that incorporate machine learning, a class of techniques where artificial networks learn from data, will be extremely important. Long Short Term Memory (LSTM), a type of recurrent neural network first described in [4], is a relatively innovative approach being used in speech recognition [1]. With a few manipulations, a novel deep learning neural network model is proposed here for a new application: protein function given primary amino acid sequence. The predicted class of a protein can be used in part with wet-lab research to establish the protein’s function as well as narrow down important sequence fragments for that function. The pragmatic use of machine learning before traditional genetic engineering and synthetic biology approaches can save multitudes of time,
resources, and money by giving the expert a probable starting place in their work. It would, furthermore, also allow for rapid analysis of big data when traditional testing may not be a readily available or profitable option. As one of the first attempts at applying LSTMs to protein sequences, the results presented here illustrate the potential value they — and other similar models — could add to the field of bioinformatics.

Artificial Neural Networks

An artificial neural network (ANN) performs mathematical procedures in an organized and seemingly redundant manner. Such networks are composed of nodes, connections, and layers, where each layer contains a specific and often arbitrary number of nodes, which individually perform a weighted sum on the inputs they are given from the previous layer. The most rudimentary and oldest form of ANN is known as a multilayer perceptron or fully-connected network. As the name indicates, each node in the network is connected to and extracts valuable features from every node in the previous layer.

The LSTM used 3 layers. The first layer is comprised of 300 nodes, each assessing the input data and forming a class prediction. The second layer, 300 nodes, assesses the input as well as the previous layer’s output before forming the prediction. The third layer, 200 nodes, assesses the previous two layer’s output in a connected format (meaning each node within this output layer makes a prediction based on the neighboring node’s output) [Fig. 2]. Once all the nodes have completed their designated algorithms to predict outputs of the proteins, an overall probability prediction is established. A more detailed model of the LSTM network used in this work is demonstrated by figure 3. This model requires ubiquitous data pertaining to a certain category and data not in said category. The data used was always the primary amino acid sequence of a protein in the form of universal one letter identifiers. A dictionary was created by a command code which converted letter identifiers of amino acid to unique numbers and then back to letters when necessary. To make each protein a consistent length, each protein was viewed as the collection of its subset smaller snippet sequences, allowing for uniform processing of the dataset. To evaluate the data this way for each protein was viewed with a sort of sliding window of variable sequence length that moves along the protein and creates the input sequence subset for each node to predict which class that snippet of the protein belongs to.

Three useful phases that can be identified during the computational procedure. In the first phase, the model is trained by assessing a large portion of the dataset with the given corresponding category (80%) before the rest of the data (20%) is assessed with hidden category. In training mode, the model uses the second batch of data to make and correct the network’s predictions about the data. The testing mode is the second phase in which a novel (not seen in training) sequences is presented to the trained
model to predict which category the entire protein belongs to. The third phase can be termed analysis as the model shows the composition of the sequence most related to the specified class. Since the model is end-to-end, the analysis is determined by the trained model in a fashion that finds the most conserved feature of the dataset. A combination of python, TensorFlow, and TFlearn are the coding language while Jupyter Notebook software housed the live code. Tensor-board was used to visualize the current progress of the model as it computed and to collect graphical results. The model generally only took a few hours to train, while testing and analysis was completed in a matter of minutes.

![LSTM Layers](image)

**Figure 2. LSTM Layers** This diagram shows a detailed model of the LSTM network used in this work.

There are a few variables within the model that must be determined to maintain consistency and optimal parameters of the network. The most meaningful variable to be specified is the sequence length. This tells the network what amino acid viewing size to look through when looking at each protein in the dataset. This variable is also highly useful to manipulate in different types of datasets for different probing purposes, as seen in the two models presented here. Secondly, the network layers and training parameters involve: number of epochs (number of times to loop through the dataset), validation set (amount of data held back in the initial training phase to be used for validation in the final training phase), batch size (number to train on each iteration), and snapshot step (how often to test validation set). The final variable needed is the number of layers to be used and the number of nodes in each layer.

**Binding Model**

Data for computation in this model was hand sourced from NCBI for proteins known from literature to bind and those known from literature not to bind to the target drug. In this case, Artemisinin was arbitrarily chosen as the target. A text file was created containing the primary sequence of each protein with a line break to separate proteins in the set. While 125 binding proteins were found and 130 non-binding proteins were found, the model only used 125 of each different category, needed for the uniformity during training mode. This model was developed first and used as the set-up for the second computation.

Since binding mechanisms are mostly unknown and highly variable for the Artemisinin target drug, a range was created from 3-150 amino acid long sequence length to be tested. The model had a loop command that told the network to run the procedures with sequence length of 3, then 4, then 5, and so on until a sequence length of 150 was met. The range 3-150 was chosen because we reasoned from literature that for a binding site to exist, it would most likely to be less than three amino acids short.
150 amino acids long was the highest tested sequence length to reduce testing time. As
the sequence length increases, the combinations of amino acids in that snippet becomes
exponentially large and requires more and more computing time. When the network
predicted the portion of a protein in the dataset was likely to bind, a value from zero to
two was giving for node one and for node zero. Node one indicates binding. Node zero
indicates not binding. During the analysis phase, the model what sequence asked what
activated node 1 the most with a variable sequence length.

Consensus Model

Data for this computation in the model was imported directly from Uniprot’s website
for protein database for proteins containing homeodomain sequences among various
species. Classes were defined as containing homeodomain sequence and not containing
homeodomain sequence. homeodomain was chosen as the key word to establish class
from the Uniprot data because the literature consensus was regarded as 60 amino acid
long, highly conserved consensus sequence [3]. Again, as data was imported, only
primary sequence was retained and proteins were distinguished by line break. Since the
amount of homeodomain proteins was much larger by the thousands, a command code
was utilized to determine the number of proteins to be pulled from Uniprot and used as
data. In this case 1500 proteins were used for the dataset.

Since the theoretically accepted length of the homeodomain consensus sequence is 60
amino acid longs, the sequence length variable to be tested was set in a range of 60-100.
The command coding to include this range of testing jumped by sets of 5. The network
first tested sequence length of 60, then 65, then 70, all the way to 100. We extended the
sequence length above the theoretical length to include for the model not getting all of
the homeodomain in its viewing window when looking at a protein form the dataset
(since it is a sliding window). If a homeodomain is predicted by the network to be
present, node 1 (consensus containing) will have a value between zero and one that is
higher than node 0 (not consensus containing). Like the previous data, during the
analysis phase, the model what sequence asked what activated node 1 the most with a
variable sequence length.

Results

Initial results were visualized with Tensorboard software, which plots the model’s
output as it runs, very useful for monitoring the efficiency of the code and determining
optimal variable parameters. While the training phase is important and elucidates the
fundamentals of neural networks, the most notable results come from the training and
analysis phases. The later being perhaps the most crucial in evaluating the multitude of
uses for machine learning LSTM’s. It should be noted that interpretation of the results,
especially regarding the analysis phase, requires a scientist knowledgeable in the
dataset’s field for the most conserved sequence in one field of data may not mean the
same as in another field of data.

Artemisinin Binding Prediction

As the binding model was ran with the range of varying sequence length, the results
were plotted on Tensorflow. The range from 3-150 was used to train the model. After
the model was trained several times, the network was given three novel sequences
separate from the original dataset. TCTP was used as a positive control sequence since
literature validates it binding ability to Artemisinin [2]. A broken TCTP sequence was
used for the negative control since the known literature binding site was manually
changed into a unrelated sequence. Finally the last sequence was engineered to model the TCTP with other related functions a genetic engineer would recommend and need for wet-lab purposes [Fig. 3].

**Figure 3. Novel Sequences for Class Prediction**

These are the novel sequence compositions of the proteins determined binding or non-binding by the model.

These sequences were then used to test the model to confirm it could predict the TCTP known to bind molecule would bind and the broken TCTP would not bind, as well as determine if a human proposed sequence would bind or not. The model completed this task with good accuracy [Fig. 4]. For protein sequences that are predicted by the model to bind, node 1 had a higher validation/accuracy score. For sequences not predicted to bind (by the model), node zero will had a higher score. The results are shown per sequence snippet analyzed for the specified sequence length variable, thus creating a line graph corresponding each snippet of sequence to a prediction value (from 0-1) of both node 1 and node 0. For the broken TCTP protein, node 0 was consistently higher than node 1, indicating the model predicts this protein to be more likely non-binding. The TCTP protein was predicted to bind as indicated by node 1 consistently being more activated than node 0. The novel engineered sequence was predicted to completely bind with no probability of non-binding.

Analysis phase determined which sequence composition activated node 1, binding, the most for a specified sequence length. Since the model is end-to-end, this sequence is what the model found the most conserved feature among the binding class. It was noticed that this sequence predicted by the model is very similar to the TCTP proposed binding site [Fig.5]. Which helped determine the composition for the engineered protein aforementioned and could be the reason why the model predicted complete binding.

**Homeodomain Consensus Prediction.**

As the previous model showed the model could determine a shared characteristic notorious in the class of the data, this model was derived to show proof of concept. When the model was prompted in the analysis phase for the most sequence that activated node one the highest with sequence length of 60 amino acids, the model
Figure 4. Class Prediction of Novel Sequences
These sequences are 1. non-binding broken TCTP protein, 2. accepted binding TCTP protein from literature, 3. theoretically engineered binding protein, class predictions by the model for sequence length 50.

Figure 5. Predicted Binding Sequence
This sequence compares the predicted sequence at 60 amino acid specified sequence length for binding by the model to a known binding site found in TCTP.

predicted a sequence extremely similar to the theoretically accepted sequence [Fig 8]. The variability in the predicted sequence is expected since a consensus sequence, even highly conserved, will be variable. These results reinforce the models ability to analyze a dataset and determine important areas of interest within a certain class. In this case, the model predicted consensus sequence by analyzing a group of similar proteins compared to a proteins not in the indicated group.

Discussion
Results indicated successful use of a Long Short Term neural network on protein data for protein class prediction. The two models asked two different things of the machine model, binding probability and consensus sequence availability. Both models utilized protein’s primary sequence as the data set and completed the same three tasks successfully. Task one required the model to learn the differences in classes for each of the classes. Task two showed the models competency in learning the classes by being
Figure 6. Predicted vs. Theoretical Homeodomain Consensus Sequence
The first sequence is the theoretical amino acid homeodomain consensus sequence. The second sequence is the predicted sequence the model showed most conserved among proteins containing the homeodomain sequence.

able to predict the appropriate class for a sequence it had not been trained with. Task three demonstrated the usefulness of the model by finding a sequence composition that was most related to the specified class. The end-to-end approach of analysis allows for the model to distinguish a conserved feature. It is the responsibility of the scientist to determine the way in which that feature corresponds to the class. The efficiency of the proposed LSTM is very valuable due to its accuracy, expediency, and variability.

Further applications of these types of computation models include integrating various diverse types of datasets and computational training using deep LSTM networks. Any type of dataset should be able to be assessed by the model as long as general categories and a relationship between them and the data-points can be established. Efficiency may be determined by the abundance of data in the set and the training ability of the network, especially if more than one category will be evaluated. The nature of this software implies that the more data in the set, the increased likelihood of accuracy due to the networks increased ability for training. Further testing will need to be done on the limits of the software, such as the need to organize large databases into readable formats. This work is the first of many data-sets to test and perfect the model for further to demonstrate its efficiency and flexibility in a new data field, such as proteomics.

Data Access
Data sets can be viewed by querying Uniprot.org.

Acknowledgments
The authors of this text would like to acknowledge Elan Barenholtze (PhD) for use of his lab’s servers, Duito Esiobu (PhD) and Douglas Holmes for suggestion of Artemisinin topic and genetic engineering input, Mirjana Pavlovic (PhD) for revisions and guidance, and FAU for funding the MPCR lab equipment.

References

2. Characterization of the Artemisinin Binding Site for Translationally Controlled Tumor Protein (TCTP), Bioconjugate Chemistry, 27, 12, 2828-2833, 2016. "http://dx.doi.org/10.1021/acs.bioconjchem.6b00556"

"https://doi.org/10.1007/978-90-481-9069-0"

"GersF.A, SchmidhuberJ,CumminsF. Learning to forget : Continual prediction with LSTM. 1999."