DeepGO: predicting protein functions form sequence and interactions using a deep ontology-aware classifier

Dreycey Albin (da39)

April 8, 2019

Scientific question

The scientific question and goal for this paper focuses on *developing a method to predict protein function from the protein's sequence*. The goal for the authors was to combine a cross-species protein interaction network and gene ontology (GO) classes in combination to predict the function of the proteins using a deep learning model.

Hypothesis tested

The hypothesis tested in this research is that *their method of integrating protein interaction networks, their vectorization of GO terms per protein, in combination with a deep learning model, may be used in combination to predict potential peotropic protein functions.* The major test was how their model compared to the other exsisting methods for predicting protein function form sequence: (1) BLASTp baseline test; (2) GoFDR; and (3) FFPred3. Using the results from these programs, they used a "protein centric maximum F-measure"(called Fmax below), a "AUC of a ROC Curve of sensitivity" for a given false positive rate, and a Mathews Correlation Coefficeint (MCC).

Methods/Results

For DeepGO, the objective was to create a model that allows the users to predict the function of a protien from the sequence of the protein. I would break their results and methodology into three categories: (1) The training and representation of the data; (2) The Convolutional Neural Network; and (3) The comparisons to BLASTp, GoFDR, and FFPred3. Namely, the authors utilize three subontologies form the GO: (1)Biological Processes (BP, 28647 classes); (2) Molecular Function (MF, 10161 classes); and (3) Cellular Components (CC, 3907 classes), thereby creating three different models (called DeepGOSeq). These were used in combination with protein interaction networks obtained from the STRING database to further improve the prediction accuracy (called DeepGO).

0.1 The training and representation of the data

After filtering the proteins for proteins with functional annotations, experimental evidence, and length, there were 60710 proteins left over. These proteins were randomly split into the training set (80 percent) and the validation set (20 percent). We these proteins they then map these to a binary label vector based on the presence or absence of different GO terms for that protein. They used this all three of the different ontologies, this was then turned into a trigram one-hot encoding. However, it looks like they then turned to dense embeddings shortly after to rid of the negative generalization effects of the one-hot embeddings (essentially a matrix with rows of length AA and columns equal to the number of embeddings). The first layer of the CNN is supposed to then learn this embedding. The training time for the ontologies was less than 3 hours.

0.2 Convolutional Neural Network Implementation

This deep learning model was implemented in the Kreas library with TensorFlow as backend. Here they use a 1-dimensional (1D) convolution over the protein seqence data. Through a convolution in the neural network, the output is a vector representing the learned features. Thereafter, max pooling is used to decrease the feature size since there will be redunant information in the vector output for the feature map. It should be noted that this is where the protein interaction information comes into the picture. The protien-protein interaction (PPI) networks were obtained from the STRING database, and thereafter connected with orthology relations from the EggNOG database. From here they generated a knowledge graph, in which they could combine with the output from the max pooling layer. This output is then considered to be higher level representation of protein sequences

which can then be used in the fully connected layers of for subsequent classification. This output is then passed to a fully connected network with 1024 neurons, before being passed to the structured neural network for further classification. This is the layer that classifies the proteins. Every class that has children in GO was given a merge layer, in which the max value of the class layer and that of the children, given the protein a term per node. The output of the model is the concatenation of the leaf nodes and the max internal nodes, giving the associated annotation for the protein. It should also be noted that the model is globally optimized using back propagation.

0.3 Results from model compared to BLASTp, GoFDR, and FFPred3

It can quickly be observed from the given table that DeepGO does not perform as well as BLAST for the MF and BP categories, but better at predicting the cellular locations of the proteins. This is mainly seen when looking at the Fmax score. In addition, when breaking the predictions down by organism, it is immediately appearent that model organisms had higher Fmax scores. The authors attribute this to the model organisms having a larger number of proteins within the database, allowing for the neural network to have a model moee suited for specific organisms that are well characterized. Testing DeepGO against GoFDR and FFPred3 yeiled a wide range of results. Namely, it can be seen that DeepGO outperformed GoFDR and FFPred3 at predicting a proteins BP and CC, but was outbeat by GoFDR for predicting MF.

Key implications of the results

Overall, the authors were able to convincingly show that a CNN was able to make a method able to predict the function of proteins, comparable to that of other top performing programs/methods. In addition, they brought a unique model to the table, by using a multi-modal data sources to create a model for predicting protein function from sequence.