Genes function prediction

Genes function prediction is the process of determining the biological role of a gene based on its sequence and/or expression data. It is a key step in understanding the genetic basis of health and disease, and is essential for the discovery of new drug targets and the development of personalized medicine. There are several different methods for gene function prediction: homology-based, expression-based, experimental, computational and bioinformatics methods. This project is using the last one of them.

At the beginning, models were supposed to predict functions of genes of chromosome 3. For this purpose, a file containing sequences of genes was downloaded from NCBI website. The next step was duplicates removal, different variants of the same genes removal, sequences indivisible by 3 removal, translation DNA sequences to protein sequences. The hardest part of this project was sequences clustering. Sequences were assigned to 7 different groups of proteins (enzymes, receptors, membrane proteins, transport involved proteins, DNA binding and transcription involved proteins, translation involved proteins, metabolism regulation involved proteins). Information about functions of the proteins were taken from GeneCards, UniProtKB, NCBI. The dataset needed to be balanced, because different genes occur in our genome with different frequency. For this purpose, the amount of sequences was multiplied. However, results obtained for this dataset weren't good enough. It is suspected that the reason is too much diversity of proteins in one group. The number of proteins groups should be much higher. Due to the fact, sequences clustering takes a lot of time and it was limited, decision about downloading the ready dataset was made. I this dataset proteins were also divided into 7 groups (transcription factors, ion channels, synthetase, synthase, tyrosine kinase, tyrosine phosphatase, g protein coupled receptors). The dataset turned to be too small, so the amount of sequences was multiplied as in a previous case. The size of the first dataset was 15 000 sequences and the second one - 70 000 sequences. First was used in k-mer based model, the second one - in RNN.

A k-mer based model uses sequences of k nucleotides (k-mers) as the basic unit of analysis. K-mers are short, contiguous sequences of nucleotides that occur within a larger DNA or RNA sequence. One of the main applications of this kind of model in bioinformatics is the prediction of gene function. By analyzing the k-mer content of a gene or set of genes, researchers can infer information about the function of those genes.

In this case, the length of k-mers was 6. To count their occurrence in text the Bag of

Words and CountVectorizer were used. To evaluate the model and test its performance Repeated K-Fold Cross-Validation was used. As a classifier Multinomial Naive Bayes was chosen. For this model very good results were obtained with dataset size of 15 000 sequences.

A recurrent neural networks (RNNs) are called "recurrent" because they have a feedback connection, or a "memory", that allows them to retain information from previous inputs in the sequence. RNNs are a type of neural networks that can process sequences of inputs, such as sequences of words in natural language processing or sequences of sensor data in time series analysis. That is why they have been used in various bioinformatics tasks, including gene function prediction.

One-hot encoding was applied to the integer representation of sequences. Model was trained with 10 epochs. All of the model's parameters were changed many times. The best results were obtained for 128 for amino acids sequences and 32 for DNA sequences. Unfortunately, in both of cases f1 score was smaller that 0.5, what means that the model is performing poorly in terms of precision and recall. There is a problem with a high number of false positives and false negatives. False positives (also known as a "Type I error") occur when a model predicts that an example belongs to a certain class, but it actually does not, false negatives (also known as a "Type II error") occur in an opposite situation. The cause is unknown. Maybe, an architecture of neural network doesn't fit the problem? Maybe a dataset is still too small…