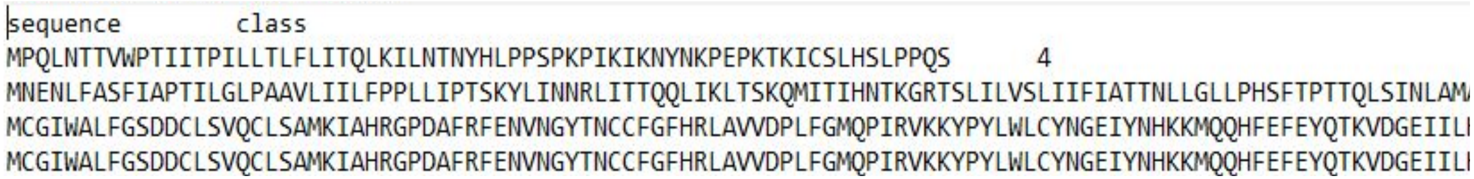
Genes function prediction

Katarzyna Foryszewska

Dataset and data preprocessing

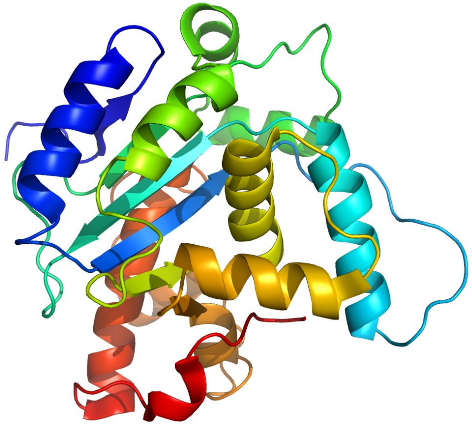
1. 4380 DNA sequences
1. Duplicates removal
2. Different variants of the same genes removal
3. Sequences indivisible by 3 removal
4. Expanding and balancing the dataset
5. Translation sequences of DNA to sequences of proteins
6. Proteins classification - assigning a class to every sequence
7. Two dataset: >15 000 and >70 000 sequences



```
sequence        class
MPQLNTTVWPTIITPILLTLFLITQLKILNTNYHLPPSPKPIKIKNYNKPEPKTKICSLHSLPPQS      4
MNENLFASFIAPTILGLPAAVLIILFPPLLIPTSKYLINNRLITTQQLIKLTSKQMITIHNTKGRTSLILVSLIIFIATTNLLGLLPHSFTPTTQLSINLAM
MCGIWALFGSDDCLSVQCLSAMKIAHRGPDAFRFENVNGYTNCCFGFHRLAVVDPLFGMQPIRVKKYPYLWLCYNGEIYNHKKMQQHFEFEYQTKVDGEIIL
MCGIWALFGSDDCLSVQCLSAMKIAHRGPDAFRFENVNGYTNCCFGFHRLAVVDPLFGMQPIRVKKYPYLWLCYNGEIYNHKKMQQHFEFEYQTKVDGEIIL
```

Protein classes:

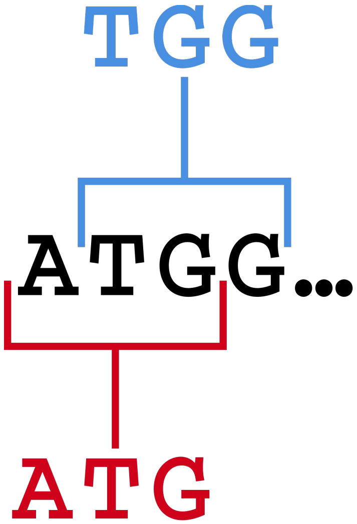
1. Transcription factors
2. Ion channel
3. Synthetase
4. Synthase
5. Tyrosine kinase
6. Tyrosine phosphatase
7. G protein coupled receptors



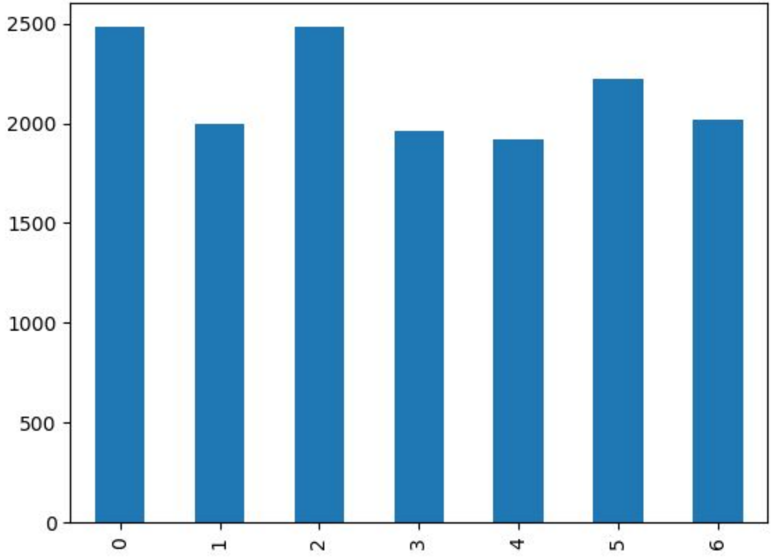
K-mers based method

1. Sequence to k-mers (lenght = 6)
2. Bag of Words, CountVectorizer
3. Repeated K-Fold Cross Validation (the best results)
4. Multinomial Naive Bayes Classifier (alpha = 0.1)

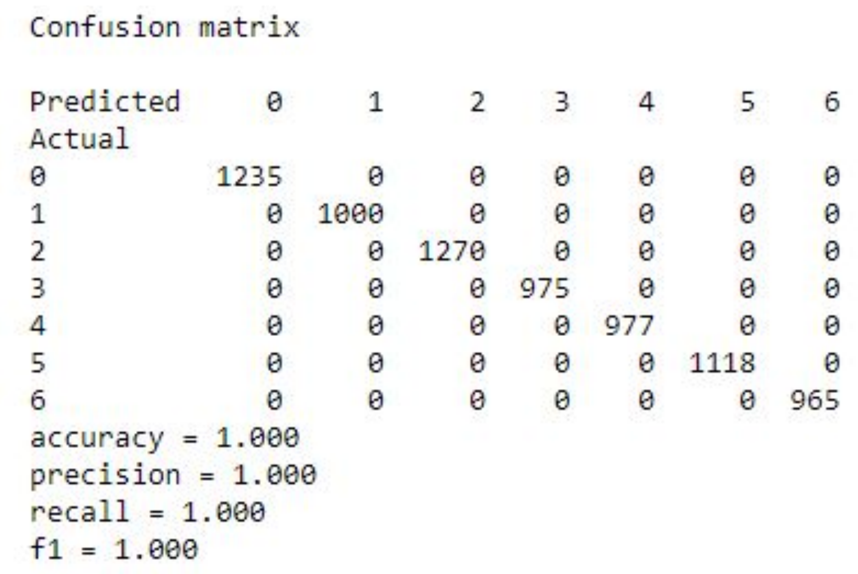
For this model results were already good enough for a dataset of 15 000 sequences.



Dataset for k-mers based method

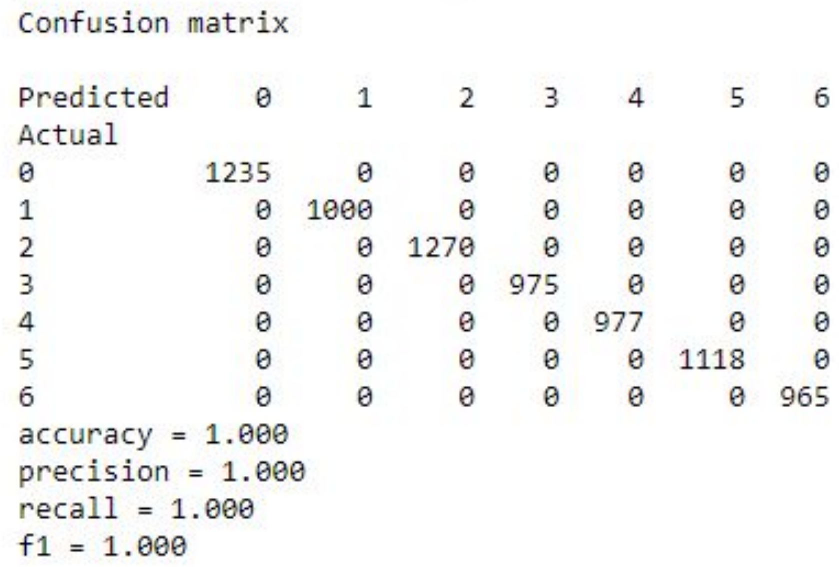


Confusion matrix for k-mers based method - DNA



```
Confusion matrix

Predicted     0     1     2    3    4     5    6
Actual
0          1235     0     0    0    0     0    0
1             0  1000     0    0    0     0    0
2             0     0  1270    0    0     0    0
3             0     0     0  975    0     0    0
4             0     0     0    0  977     0    0
5             0     0     0    0    0  1118    0
6             0     0     0    0    0     0  965
accuracy = 1.000
precision = 1.000
recall = 1.000
f1 = 1.000
```

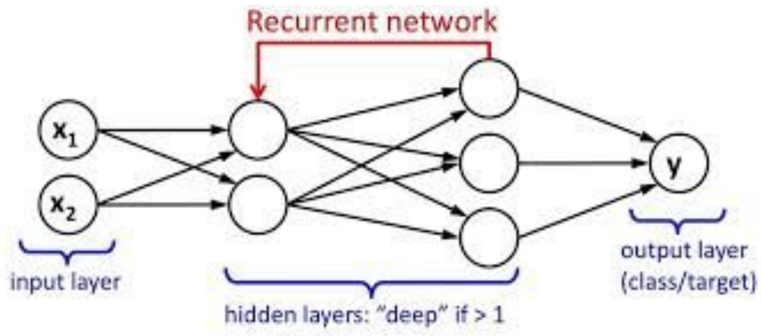
Confusion matrix for k-mers based method - proteins



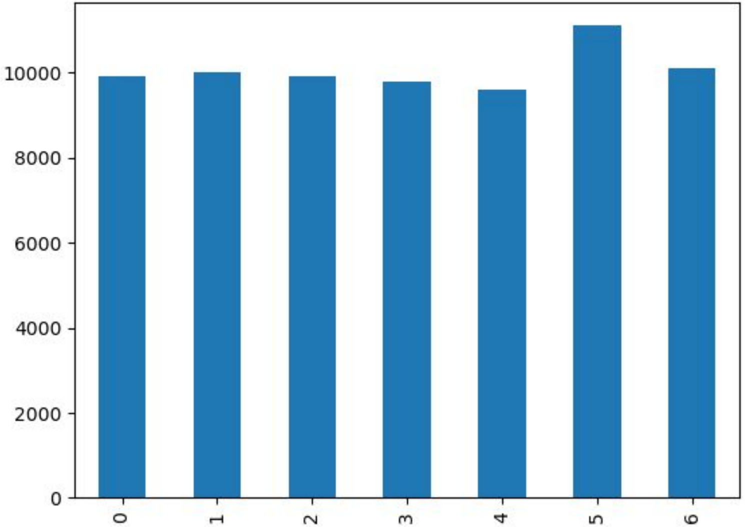
```
Confusion matrix

Predicted     0     1     2    3    4     5    6
Actual
0          1235     0     0    0    0     0    0
1             0  1000     0    0    0     0    0
2             0     0  1270    0    0     0    0
3             0     0     0  975    0     0    0
4             0     0     0    0  977     0    0
5             0     0     0    0    0  1118    0
6             0     0     0    0    0     0  965
accuracy = 1.000
precision = 1.000
recall = 1.000
f1 = 1.000
```

Recurrent Neural Network for DNA sequences

1. Dataset of more than 70 000 sequences
2. One Hot Encoding
3. Hidden size: 24
4. Final f1 score for DNA: 0.17



Dataset for RNN



Recurrent Neural Network for protein sequences

1. Dataset of more than 70 000 sequences
2. One Hot Encoding
3. Hidden size: 128
4. Final f1 score for proteins < 0.15

Conclusions

1. Dataset was extended to more than 70 000 sequences. All of the possible parameters were changed many times and result was always the same.
2. Results are much better for k-mers based method. It means, that it’s not easy to train neural network on DNA or protein sequences, but…
3. Maybe architecture of used neural network is improper for this specific case?

Thanks for your attention!