BIOINFORMATICS

Vol. 00 no. 00 2018
Pages 1–5

Predicting the Subcellular Location of Eukaryotic Proteins

Nathalie von Huth

Department of Computer Science, UCL

nathalie.huth.14@ucl.ac.uk

ABSTRACT

The motivation of this task is to develop a simple method for classifying eukaryotic protein sequences into 4 different subcelluar location categories. The four subcellular locations are: cytosolic, secreted, nuclear and mitochondrial. Being able to classify such can give a guidance of to which function the protein has. By having the data set of over 3000 pre-defined genomes, it creates an opportunity to use machine learning concepts to tackle such a task.

This paper shows the process of identifying and combining different features that can help allocate a certain protein sequence into its subcellular location. A Multi-layer Perceptron (MLP) was used to classify the eukaryotic protein sequences. The MLP consisted of a hidden layer size of (200,10). It managed to classify the cytosolic location with a confidence metric of ≈ 0.83, but failed significantly on classifying the nuclear location, with a confidence metric of only ≈ 0.01. The mitochondrial achieved ≈ 0.56 and the secreted around ≈ 0.69.

1 INTRODUCTION

Along with the rise of generated protein sequences, there has been several achievements in the area of predicting subcellular locations based on the protein sequence information (6), (7). It is currently known that only between 40-60% of the genes in the human genome can be assigned a functional role. Long-established methods has been previously used to relate the protein function directly to the three-dimensional structure of the polypetide chain. The issue with this is that is it difficult to calculate. Another approach is to utilize the feature space of the sequences, which can be used to compare and weight different sequences against each other. Studies of cell biology often find it useful with in-depth information about the subcellular location of the proteins. It has shown to be particularly useful in proteomics and system biology, where it opens up for the possibility to get some information about the function of the protein. The study of subcellular localisation has shown to have great importance in understanding protein function, mechanisms involved in pathology of different diseases and new drug tests. Hence, a tool that could easily assign a function to a protein is very useful for many biologists. The computational approach of solving this problem is highly attractive over a traditional experimental method, specifically when dealing with high-throughput data of genome sequences. Such computational approaches often gives much higher accuracy at a much faster speed, automatically. This project analyses the protein sequence in detail and extracts important features that can help determine the subcellular location. The project implements several simple machine learning techniques to help identify which out of four subcellular locations the protein belongs to.

2 APPROACH

The task consists of different steps that helps identifying the different protein functions:

1. Feature engineering
2. Data cleaning and Data exploration
3. Feature selection
4. Model selection
5. Model optimisation

These steps requires further understanding of the different protein compositions, their features and how they might affect the protein function. Each of the steps above will be explained further in section 3. The four subcellular locations that will be predicted is:

- **Cytosolic** : when the protein is located within the cell itself, but not inside any organelles
- **Secreted** : proteins transported outside the cell
- **Nuclear** : proteins located within the cell's nucleus
- **Mitochondrial** : proteins transported to the cell's mitochondria

3 METHODS

3.1 Feature Engineering

Since no features were given in the data set initially, these needed to be extracted from the protein sequences. A total of 16 features were extracted from the sequence, some which turned out quite useful at a later stage for the protein prediction.

All the different features are described in Table 3.1. I will explain a few here more in detail. The feature 'GCcount' shows the presence of Guanine and Cytosine in the protein sequence. This indicates the percentage of nucleotide base pairs where Guanine is bonded to Cytosine. The sequence of nucleotides determines the transcription of RNA, which again determines the primary structure (order of amino-acids) of the protein after translation. The amino acids also contains several side chains which operates best in certain environments (the protein subcellular location). E.g if some of the outer side chains of the protein are hydrophilic, it can make it easier for the protein to dissolve in water and can operate within the blood (hemoglobin). Other proteins have hydrophobic side chains and have other functions such as fibrous structures (e.g. keratin in hair or intermediate filaments in many cells). The quantity of GC determines ultimately which amino acids will form the primary sructure of the protein. Another feature, 'isX', is describing whether there is a part of the sequence that is undefined. It might appear that sequences with more/less X's might more often belong to a certain subcellular location.

Feature	**Description**
seq_len	Length of the sequence
global_aa_comp	Global amino acid composition
isoelectric_point	Isoelectric point - the pH when the molecule has electrical charge in terms of the statistical mean
aromaticy	Aromaticity as defined in (2)
secondary_structure_fraction0	Calculate fraction of helix, turn and sheet, first parameter.
secondary_structure_fraction1	Calculate fraction of helix, turn and sheet, second parameter.
secondary_structure_fraction2	Calculate fraction of helix, turn and sheet, third parameter.
flexibility	Flexibility as defined in (3)
molecular_weight	Molecular weight of protein sequence.
isX	1 if X is present in the sequence, 0 otherwise.
instability_index	Instability Index as defined in (4)
gravy	Gravy as defined in (5)
local_aa_comp_firsthalf	Composition of amino acid in the first part of the sequence
local_aa_comp_secondhalf	Composition of amino acid in the second part of the sequence
seq_species	Species derived from the seq_id
GCcount	Count of guanine and cytosine in sequence, normalised by seq_len

Table 1. Features derived from the protein sequence.

3.2 Data Transformation and Exploration

To be able to understand the data, we need to transform and explore it. The transformation is needed in order to get the right format for the models to work with. The exploration seeks meaning of the features. This includes seeing correlations between features and discarding features without any contribution to the target values.

3.2.1 Data Transformation First, some of the data features includes categories containing strings. This is converted to numeric values using the LabelEncoder in scikit learn. The encodings are stored in a dictionary in case wanting to encode again at a later point or reversing the encoding. Secondly, since some of the sequences consists of the value 'X', some features (e.g instability_index, gravy, etc) are not computable. Hence, the mean of the columns of the values for that specific subcellular location is used to fill the empty spaces. The third part of the data transformation is standardising features with continuous values.

3.2.2 Data Exploration To get an idea of which data is relevant/irrelevant, Pearson's r is calculated to see the correlation between each of the features from data_train compared to the subcellular location. Figure 1 shows a histogram that displays the strength of the negative/positive correlation. If r is close to zero it is not correlating to the subcellular location, but if it is close to 1 or -1 it is correlating.

Another way to visualise the correlations is feature by feature. This can be shown in a correlation matrix. The matrix can give us some insight into key biology concepts, and we can see which features correlate to each other. This can be seen in Figure 2. We can see in the figure that for example the local amino acid composition of the first and second half is strongly correlated with the global amino acid composition, which is as expected. The molecular weight is also strongly correlated with the sequence length, which is also expected. One of the most interesting parts of the matrix is to see which features correlates with each individual subcellular location. One problem that can be seen at this stage is how the nucleus label doesn't seem to correspond to almost anything. This might cause problems later in the prediction stage.

3.3 Feature Selection

Even if both Pearson's r and the correlation matrix gives us an idea of how much a feature is related to the target value, the feature might have an indirect contribution. To test recursively whether some features have a relationship with the subcellular location we can perform feature selection techniques to get rankings of the importance of the features. Feature selection reduces overfitting, where less redundant data means less opportunity to make decisions based on noise. It improves accuracy such that there is less misleading data, hence modeling accuracy improves. It also reduces the training time - less data means that algorithms train faster. The models that are being trained for feature selection is:

- Stability (Randomized Lasso)
- RFE (Recursive Feature Extraction)
- Feature Importance (Extra Trees Classifier)

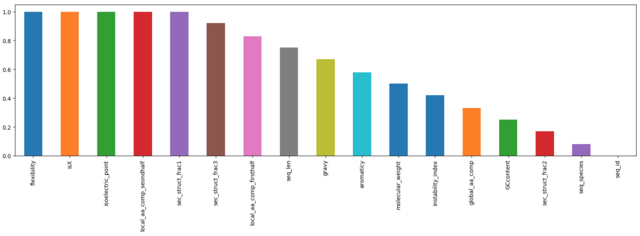


Fig. 3. Ranking of the importance of the feature by the RFE algorithm.

From the three techniques used above, the RFE seemed to make the most sense in regards to the data. Since some features are clearly more important than others, RFE was therefore used to cut out features that didn't contribute enough to the subcellular location. The ranking of the features by the RFE algorithm can be seen in Figure 3.

3.4 Model Selection

There are many different machine learning models, but only some are relevant for a task of classification. I have used scikit

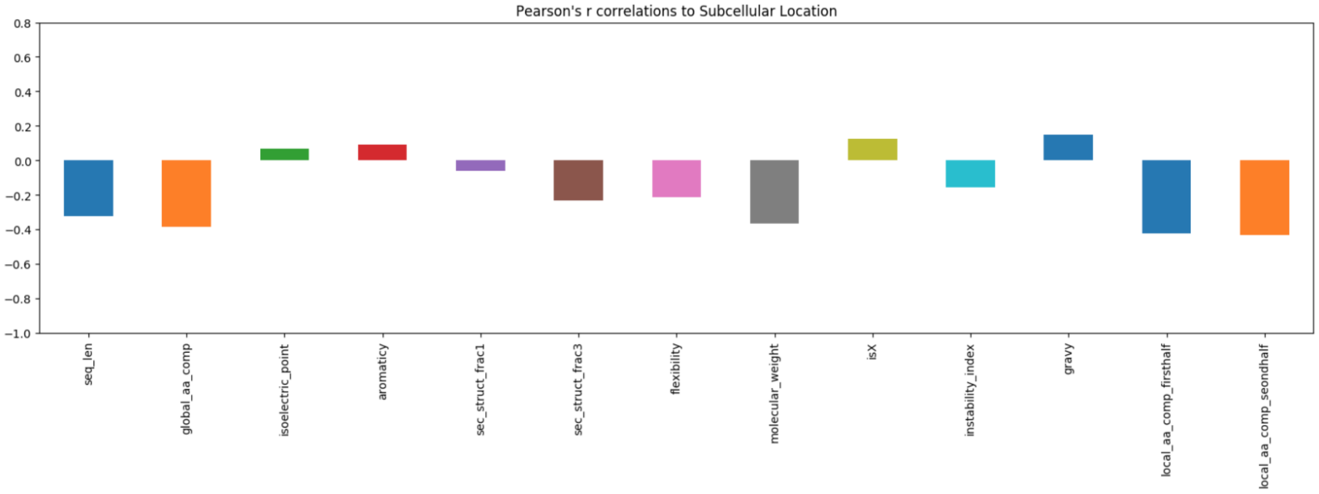


Fig. 1. Pearson's r with all the features against the subcellular location.

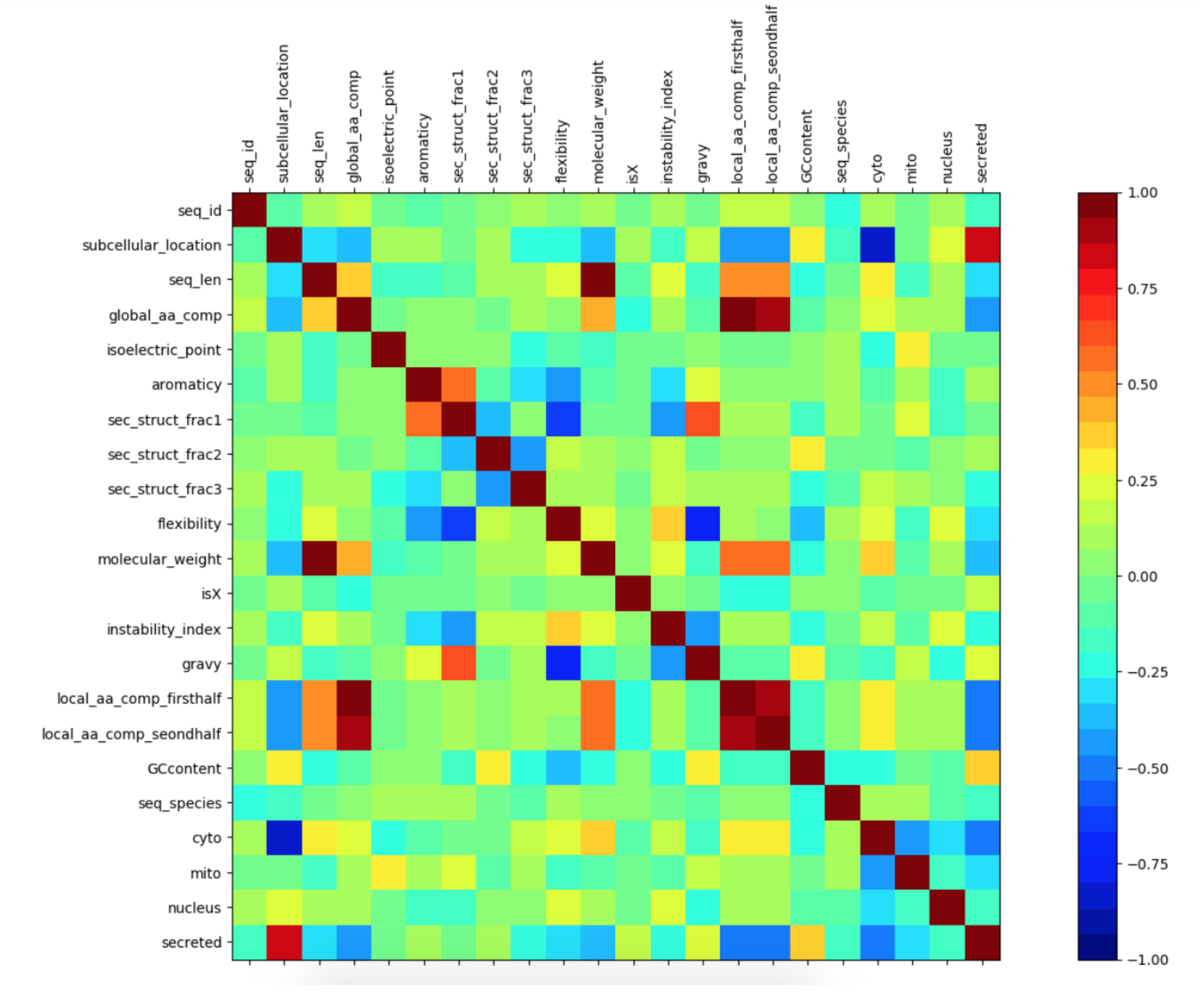


Fig. 2. Correlation matrix between all the features

learn's cheat-sheet (`http://scikit-learn.org/stable/tutorial/machine_learning_map/`) as a guideline of which specific models to use. From background readings I decided to use these models:

- Support Vector Classifier
- Gaussian Naive Bayes
- KNN
- Ada Boost
- Logistic Regression
- Multi-layer Perceptron (MLP)

The reason I use for example Ada Boost is that it fits a sequence of weak learners (i.e. desicion trees) repeatedly into a combination that makes it very powerful. A very simple neural network (MLP) was also used as it has shown in previous research that this can be good for when dealing with proteins.

Before training the different models which we will perform the prediction on, a test set will be put aside in order to get evaluations on data that the model has not seen before. Moreover, in order to get correct evaluation results from the model we need to train it on different types of splits of the data set. This is called cross validation (CV), where I use k-fold CV to split into k smaller sets. By using cross validation, we are able to objectively compare the models in terms of their respective fractions of misclassifications. With the CV we can compute both CV accuracy score and the area under the ROC curve (AUC) to get an idea of which model performs better. Figure 4 compares the six different models, with the results having being cross-validated with k=3 splits. From the model we can see that the AUC score for all the models stays very low and at the same level, while the accuracy spikes for some (MLP, Logistic Regression).

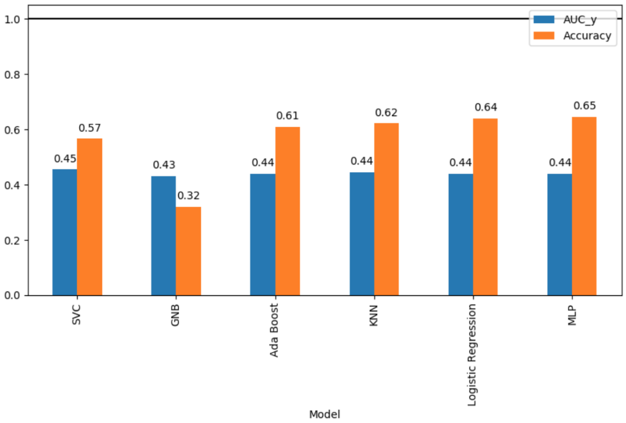


Fig. 4. Comparison of the different machine learning models in terms of accuracy score and AUC score. They are all cross-validates with k=3 splits.

By comparing the results, the best model turned out to be the Multi-layer Perceptron and was picked to study more in depth.

3.5 Model Optimisation

The model can now be trained and tweaked further to improve the accuracy of the model. Some of the parameters used was to use the Adam Optimiser as the solver for the weight. The logistic function was used as the activation function between the neurons and follows this equation:

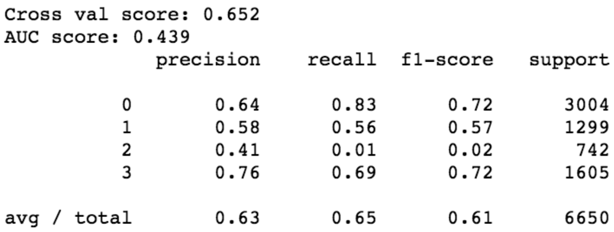
$$f(x) = \frac{1}{1+e^{-x}}$$

The learning rate is set to $\alpha = 0.00001$ and with a hidden layer size of (100,10). The classification report can be seen in Figure 5. The report shows us the precision, recall and F1-score for the four different subcellular locations (0='cyto', 1='mito', 2='nucleus', 3='secreted'). The precision and recall are defined as follows, where $tp = truepositive$, $fp = falsepositive$ and $fn = falsenegative$ in terms of classification.

$$Precision = \frac{tp}{tp+fp}$$

$$Recall = \frac{tp}{tp+fn}$$

The precision seem to be relatively good for all the subcellular locations, but the recall seems to fail when it comes to classifying the nucleus. This means that most of the 'nucleus' instances that were misclassified to another subcellular location. The combination of the precision and recall is showed in the F1-score and also demonstrate the declination of classifying 'nucleus'. The support shows us how many of the samples belonged to the different groups, where there is considerably less samples avaialble for the 'nucleus'.



```
Cross val score: 0.652
AUC score: 0.439
             precision    recall  f1-score   support

          0       0.64      0.83      0.72      3004
          1       0.58      0.56      0.57      1299
          2       0.41      0.01      0.02       742
          3       0.76      0.69      0.72      1605

avg / total       0.63      0.65      0.61      6650
```

Fig. 5. Classification Report of the Multi-layer Perceptron model.

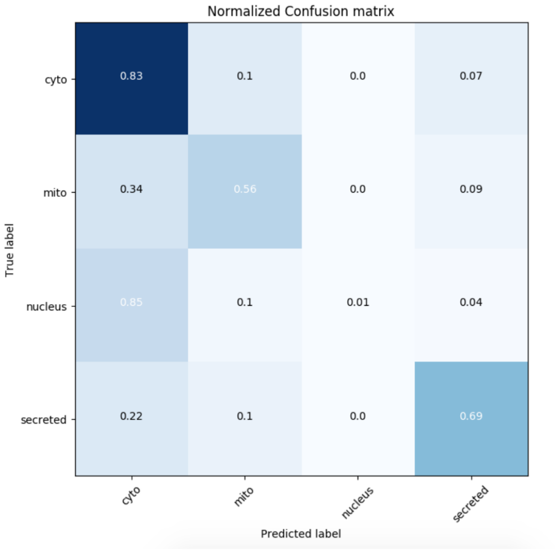


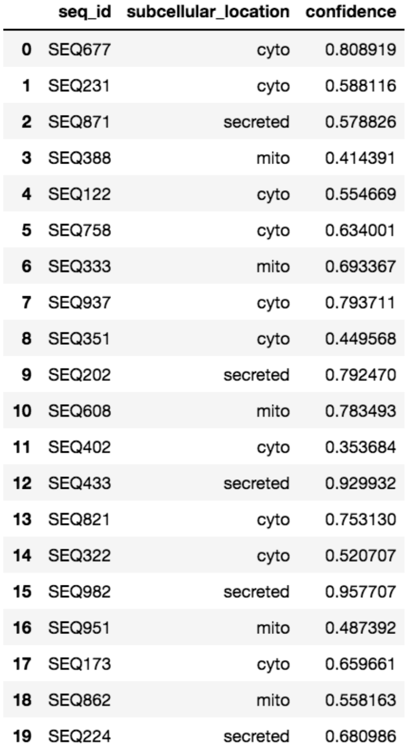
Fig. 6. Confusion matrix of the Multi-layer Perceptron classifications.

Lastly, we can analyse the confidence of the predictions of the different subcellular locations through a confusion matrix as shown in Figure 6. The model seems to be very confident on predicting ’cyto’, while misclassifying ’nucleus’ almost all of the time. Most of the time when the model is classifying something wrong, it’s taken to be ’cyto’. The diagonal in the confusion matrix corresponds to the confidence of each of the classifications of the subcellular locations.

4 RESULTS & DISCUSSION

The final predictions of the blind data set is shown in Figure 7 along with the confidence estimates. The confidence estimate is a method of a (soft) classifier outputting the probability of the instance being in the specific class.

One of the major problems that can be seen in this task of classifying eukaryotic protein sequences is the task of better distinguishing between the different subcellular locations. The proteins found in the nucleus were almost impossible for any of the machine learning algorithms to recognise and was almost always classified as a cytosolic location. There might be several reason for this misclassification. The most obvious observation is from the classification report in Figure 5, where we can see that there are much less samples available for the nuclear subcellular locations. Compared to the amount of samples from the other groups, it might make it difficult to find patterns in the data and it might confuse it for the other groups. We can also see that the AUC scores of all the models (see Figure 4) are extremely low, indicating that there are many false positives calculated by the model. This is reflected in the correlation matrix in Figure 2. Most of the time when the model is classifying something wrong, it’s taken to be a Cytosolic location. This might tell us that ’cyto’ might have a lot of similar properties that are present in the other groups. There can be better approaches to have features that are able to differentiate ’cyto’ from the other locations. For future work it might be an idea to have a more complicated neural network to automatically identify the features in the data set, rather than performing feature engineering.



	seq_id	subcellular_location	confidence
0	SEQ677	cyto	0.808919
1	SEQ231	cyto	0.588116
2	SEQ871	secreted	0.578826
3	SEQ388	mito	0.414391
4	SEQ122	cyto	0.554669
5	SEQ758	cyto	0.634001
6	SEQ333	mito	0.693367
7	SEQ937	cyto	0.793711
8	SEQ351	cyto	0.449568
9	SEQ202	secreted	0.792470
10	SEQ608	mito	0.783493
11	SEQ402	cyto	0.353684
12	SEQ433	secreted	0.929932
13	SEQ821	cyto	0.753130
14	SEQ322	cyto	0.520707
15	SEQ982	secreted	0.957707
16	SEQ951	mito	0.487392
17	SEQ173	cyto	0.659661
18	SEQ862	mito	0.558163
19	SEQ224	secreted	0.680986

Fig. 7. Blind test results.

5 CONCLUSION

Most of the time when the model is classifying something wrong, it’s taken to be a Cytosolic location. This might tell us that ’cyto’ might have a lot of similar properties that are present in the other groups. There can be better approaches to have features that are able to differentiate ’cyto’ from the other locations. For future work it might be an idea to have a more complicated neural network to automatically identify the features in the data set, rather than performing feature engineering.

REFERENCES

[1]Bofelli,F., Name2, Name3 (2003) Article title, *Journal Name*, **199**, 133-154.

[2]Lobry, J. R., & Gautier, C. (1994). Hydrophobicity, expressivity and aromaticity are the major trends of amino-acid usage in 999 Escherichia coli chromosome-encoded genes. Nucleic Acids Research, 22(15), 3174-3180.

[3]Vihinen, M., Torkkila, E., & Riikonen, P. (1994). Accuracy of protein flexibility predictions. Proteins: Structure, Function, and Bioinformatics, 19(2), 141-149.

[4]Guruprasad, K., Reddy, B. B., & Pandit, M. W. (1990). Correlation between stability of a protein and its dipeptide composition: a novel approach for predicting in vivo stability of a protein from its primary sequence. Protein Engineering, Design and Selection, 4(2), 155-161.

[5]Kyte, J., & Doolittle, R. F. (1982). A simple method for displaying the hydropathic character of a protein. Journal of molecular biology, 157(1), 105-132.

[6]Nakashima, H., & Nishikawa, K. (1994). Discrimination of intracellular and extracellular proteins using amino acid composition and residue-pair frequencies. Journal of molecular biology, 238(1), 54-61.

[7]Cedano, J., Aloy, P., Perez-Pons, J. A., & Querol, E. (1997). Relation between amino acid composition and cellular location of proteins1. Journal of molecular biology, 266(3), 594-600.