Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: 24th March 2017
Manuscript Category

COMPGI10 Coursework

Predicting the subcellular location of eukaryotic proteins

Daniel Friar, 110008845

Abstract

We have developed a machine learning method to predict subcelluar location of eukaryotic proteins from features derived from raw amino acid sequences. A random forest model is fit to approximately 7,000 protein sequences which achieves an accuracy of 66.2% and a mean AUC of 0.702 on an unseen test set consisting of 2,300 sequences. Further, the model is analyzed in order to determine which sequence characteristics are important for predicting protein location, used to make predictions on an unlabelled dataset and extensions are proposed based on more sophisticated modelling approaches.
Contact: ucabdfr@ucl.ac.uk

1 Introduction and Approach

In recent years, machine learning approaches to predicting protein function have become increasingly popular due to large increases in available genomics data. Determining protein function can be difficult, particularly when there is no obvious homology with proteins whose function has previously been determined and as a result, computational methods for determining function directly from amino acid sequences are becoming increasingly important. A key step in determining protein function is the prediction of sub-cellular protein location (see the feature importance figure in (1)). In the following we present a machine learning approach to predicting protein location from amino acid sequences.

A dataset of 9,222 amino acid sequences was obtained, which was split into training and test sets (see the methods section below for more details). A large number of features were computed from the sequences which were later analysed in order to determine which were the most important predictors of protein location. Many features were created based on the proportion of each amino acid in different parts of the sequence, similarly (but with slightly fewer features) to the approach presented in (2).

While a deep neural network approach may be preferable purely for prediction purposes, in this case a random forest model was chosen due to it's relative simplicity, ease of interpretability and short training time. In particular this allowed us to more accurately assess the importance of the generated features. A valuable extension of the work presented may be to use the most significant features from the random forest model and feed the corresponding data into a more sophisticated model in order to obtain higher prediction accuracy.

In the following sections, results are presented for the model's performance on the test set, along with location predictions for 20 unlabelled amino acid sequences.

2 Methods

2.1 Dividing data

The full training dataset consisted of 9,222 amino acid sequences along with labels corresponding to each of the four sub-cellular location classes. A second "blind" dataset was provided, containing 20 amino acid sequences and no class labels.

A random sample of 25% of the full training set was split aside and used as a test set to evaluate final model performance. Cross-validation was performed on this training set to assess feature importance and choose hyperparameter values. This gave us:

- Training set: 6,916 examples
- Test set: 2,306 examples
- "Blind" (unlabelled) test set: 20 examples

Table 2.1 shows the distribution of classes for the training and test sets, indicating that there are approximately twice as many examples of cytosolic and nuclear proteins as the other two classes. While the class imbalance is not large enough to present a significant problem for the classifier, a possible extension may to re-sample from the training set in order to address the imbalance (a similar approach is presented in (3), involving producing synthetic examples from the minority class).

	Cytosolic	Mitochondria	Nuclear	Secreted
Train	2251	977	2495	1193
Test	753	322	819	412

Table 1. Class distribution per dataset.

2.2 Feature creation

The amino acid sequences were parsed and features were created and fed into the model. It's worth noting that a small number (just under 0.7%) of the given sequences contained "non-standard" amino acids B, J, O, U, X and Z, with the vast majority of these occurring in the secreted class. Amino acids O, U and X were removed from the sequences (after computing sequence length) and B, Z and J were replaced with N, Q and L respectively.

Initially, 72 features were created and feature importance was assessed throughout training in order to determine which features were most relevant for location prediction. The created features were as follows:

- Sequence length
- Amino acid composition - proportion of each amino acid in the chain (20 features)
- Proportion of each amino acid in the first 50 elements of the chain (20 features)
- Proportion of each amino acid in the last 50 elements of the chain (20 features)
- Molecular weight *
- Isoelectric point *
- Aromaticity *
- Hydrophobicity (calculated using the Kyte and Doolittle index)
- Secondary structure fraction - fraction of amino acids that tend to be in an alpha helix, turn or beta sheet (3 features) *
- Proportion of hydrophobic AAs, proportion of positively and negatively charged AAs (from the Willie Taylor Venn diagram) (3 features)
- Presence of Nuclear Localization Signals (NLS) (1 binary feature). Approximately 120 classic nuclear localization signals were obtained from the database presented in (5) and the prescence of these signals was checked in the amino acid sequences.

*NB. Features followed by a * were computed using the BioPython module.*

2.3 Fitting model, feature selection and parameter tuning

A Random Forest model is fit to the data in order to make predictions for the classes. A grid search over various parameter values was performed in order to choose the optimal hyperparameter setting. This was done by selecting the parameters that gave the highest 5-fold cross validation accuracy on the training set. After optimizing parameters, the relative feature importance is calculated by considering the mean decrease in the Gini index for each parameter - see the plot in figure 1 which shows the relative importance of the top 50% of features. Including just the top 50% of features in the model did not lead to any significant decrease in cross-validation accuracy, and as a result the bottom 50% of features were dropped, leaving the 36 features shown in figure 1. The hyperparameter optimization was then repeated on this dataset with fewer features. Removing further features did lead to a decrease in cross-validation accuracy (using just the top 25% of features gave a decrease of approximately 2.5%) and as a result this was not done.

It's worth noting that over-fitting did not appear to be a problem throughout training, with the cross-validation accuracy giving very similar values to the training accuracy.

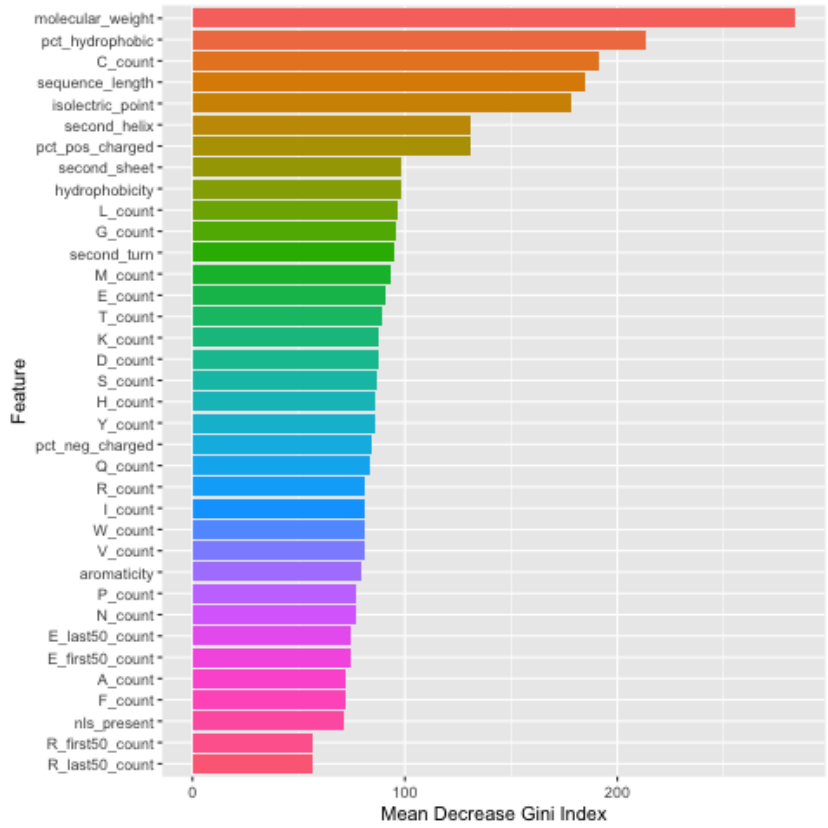

Fig. 1: Plot of relative feature importance in Random Forest model for the most important 50% of features.

The plots in figures 2 to 6 show the distributions of the four most important features over each of the classes (after which there is a significant fall-off in mean Gini index decrease as shown in figure 1), each of which show significant differences between the subcelluar locations, with the exception of cytosolic and nuclear which is discussed in more detail below. It's worth noting that in all cases a small number of outliers have been removed to make the plots more interpretable.

A key observation from these plots is that the nuclear and cytosolic classes are reasonably similar in each of the five most important features (in particular in molecular weight and sequence length) and as a result these classes are the most difficult to distinguish (as can be seen in the confusion matrix presented in the results section). Initial analysis of these plots motivated the inclusion of the presence of NLS signals as a feature, however the feature importance plot indicates that this feature (*nls_present*) did not play a hugely significant role in the model. Approximately 36% of nuclear proteins in the training set contained one of these NLS signals (compared with an average of approximately 11% for the other classes),

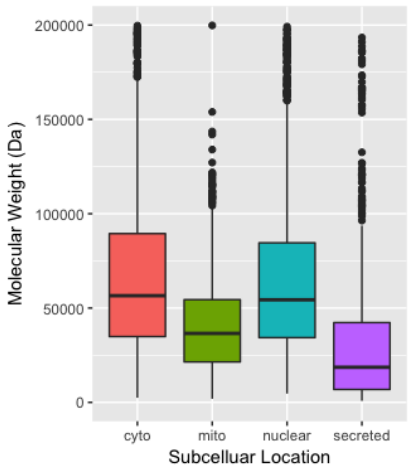

Fig. 2: Plot of molecular weight distribution over the four classes.

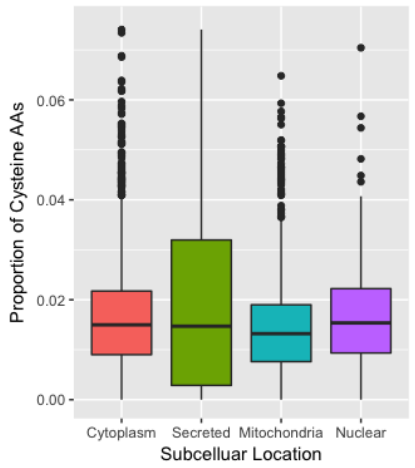

Fig. 5: Plot of proportion of cysteine AAs distribution over the four classes.

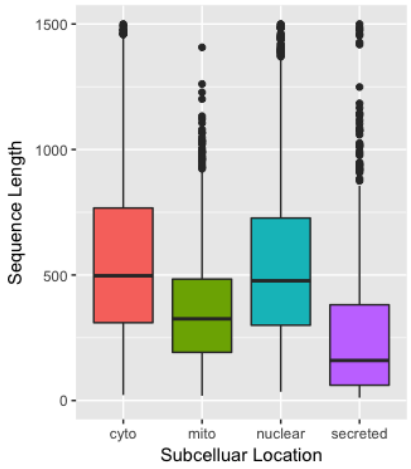

Fig. 3: Plot of sequence length distribution over the four classes.

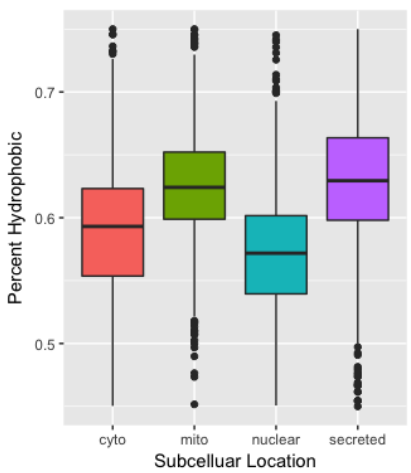

Fig. 4: Plot of % hydrophobic distribution over the four classes.

suggesting that a more detailed database containing more NLS

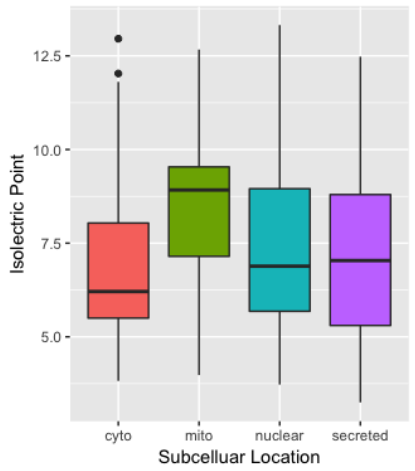

Fig. 6: Plot of molecular isoelectric point distribution over the four classes.

signals may increase this feature's impact and help to improve prediction performance.

The model was then used to make predictions on the test set and various statistics calculated to assess the fit to the test data. Finally, the model was fit to the "blind" test set in order to predict the classes for the 20 unknown sequences.

3 Results and Analysis

3.1 Performance on test data

Using a winner-takes-all approach, i.e. choosing the class with the highest probability as the predicted class, an overall accuracy of 66.2% is achieved on the test set.

The confusion matrix for the four classes is given in table 2. This clearly indicates that nuclear and cytosolic proteins were the most often confused. As mentioned above, nuclear and cytosolic proteins have similar values for the key features that were used in the model which is likely the reason for this.

The ROC curves and corresponding AUC values are given in figure 7 and table 3 respectively, again illustrating that the model performed the worst on nuclear and cytosolic proteins but with reasonably good scores for the other two classes, in

	cyto	mito	nuclear	secreted
cyto	446	49	219	39
mito	56	213	40	13
nuclear	198	34	567	20
secreted	48	23	40	301

Table 2. Confusion matrix for test set.

particular the secreted class. The mean AUC score (weighted by the number of examples in each class in the test set) was 0.701.

Class	AUC score
cyto	0.629
mito	0.716
nuclear	0.702
secreted	0.795

Table 3. AUC score per class

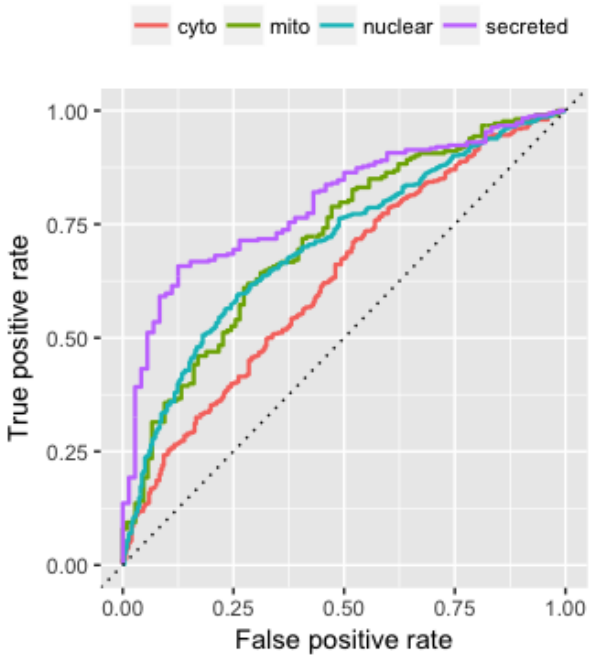

Fig. 7: ROC curves for the four classes.

The confusion matrix for the training set is shown in table 4 below, also showing that cytosolic and nuclear were the most easily confused classes.

	cyto	mito	nuclear	secreted
cyto	1374	141	639	97
mito	186	603	130	58
nuclear	589	112	1735	59
secreted	173	58	116	846

Table 4. Confusion matrix for training set.

While the mis-classifications for mitochondrial and secreted proteins are much more balanced over the 3 classes (compared to nuclear or cytosolic), the confusion matrices indicate that in both cases these proteins were more frequently misclassified as either nuclear or cytosolic. This is likely due to the previously mentioned class imbalance in the dataset, in which there are approximately twice as many examples of nuclear and cytosolic proteins as the other two classes. One way to address this issue is to simply obtain more examples of mitochondrial and secreted proteins or alternatively to re-sample from the training set prior to fitting the model (e.g. by including random secreted/mitochondrial examples multiple times).

3.2 Predictions on new data

The trained model was used to predict the class for the “blind" dataset, which contained 20 unlabelled amino acid sequences. The predictions and associated confidence scores (probabilities) for each of the 20 sequences are presented in table 5 below, along with the sequence identifier that identifies the protein.

	Sequence Identifier	Class Prediction	Confidence
1	SEQ122	nuclear	0.71
2	SEQ173	cyto	0.60
3	SEQ202	mito	0.70
4	SEQ224	cyto	0.38
5	SEQ231	secreted	0.63
6	SEQ322	nuclear	0.93
7	SEQ333	cyto	0.37
8	SEQ351	cyto	0.56
9	SEQ388	nuclear	0.72
10	SEQ402	mito	0.70
11	SEQ433	secreted	0.82
12	SEQ608	mito	0.79
13	SEQ677	cyto	0.42
14	SEQ758	nuclear	0.75
15	SEQ821	secreted	0.73
16	SEQ862	cyto	0.35
17	SEQ871	nuclear	0.31
18	SEQ937	cyto	0.68
19	SEQ951	cyto	0.44
20	SEQ982	nuclear	0.85

4 Conclusion and extensions

The work presented illustrates that a simple and easily extensible machine learning model can obtain reasonable success in predicting subcelluar location of proteins. While performance on mitochondrial and secreted proteins was good, nuclear and cytosolic proteins were often confused, indicating that the inclusion of further features may be necessary in order to distinguish between these two classes. Including a larger database of NLS signals may also help to address this issue, and this is something that could be seamlessly integrated into the existing model.

Since nuclear and cytosolic proteins made up the majority of the dataset, including more examples from the other two

classes (or using re-sampling as mentioned in section 2) may also help to improve prediction accuracy.

A further possible extension to the work presented may be to use the features that were identified as important by the random forest model and fit alternative machine learning models which may achieve better predictive performance (for example a multi-layer feed-forward neural net or a kernel SVM). This is likely to improve predictive performance at the cost of reduced interpretability.

References

[1]Jensen LJ, Gupta R, Blom N, Devos D, Tamames J, Kesmir C, Nielsen H, Staerfeldt HH, Rapacki K, Workman C, Andersen CA, Knudsen S, Krogh A, Valencia A, Brunak S. (2002) Prediction of human protein function from post-translational modifications and localization features., *Elsevier Science Ltd*.

[2]Arvind Singh Mer and Miguel A Andrade-Navarro (2013) A novel approach for protein subcellular location prediction using amino acid exposure, *Biomed Central*.

[3]Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, W. Philip Kegelmeyer (2002) SMOTE: Synthetic Minority Over-sampling Technique, *Journal of Artificial Intelligence Research*.

[4]Allison Lange, Ryan E. Mills, Christopher J. Lange, Murray Stewart, Scott E. Devine and Anita H. Corbett. Classical Nuclear Localization Signals: Definition, Function, and Interaction with Importin $\alpha*$, *Journal of Biological Chemistry*.

[5]Simarjeet Negi, Sanjit Pandey, Satish M. Srinivasan, Akram Mohammed, Chittibabu Guda. LocSigDB: a database of protein localization signals, *The Journal of Biological Databases and Curation*.

[6]Haibo He, Edwardo A. Garcia. Learning from Imbalanced Data, *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 21, NO. 9, SEPTEMBER 2009*.

[7]Kuo-Chen Chou, Hong-Bin Shen. Recent progress in protein subcellular location prediction, *Analytical Biochemistry, 2007*.

[BioPython]

1 Source code for project

Please see the repo here for project source code: *https://github.com/dannyfriar/bioinformatics_coursework*

Brief explanations of what the relevant files do:

- *prepare_Data/prepare_data.py* loads the Fasta files and creates features
- *prepare_Data/exploratory_plots.R* produces the boxplots and visualizes relationships between features
- *Models/random_forest.py* does the hyperparamter optimization
- *Models/random_forest.R* runs the final model and performs analysis on results