Predicting the sub-cellular location of eukaryotic proteins

William Ferreira

April 10, 2015

Abstract

Protein sub-cellular location prediction is the task of predicting where a protein resides in a cell, in order to understand its function. This article presents a machine-learning approach for classifying protein sub-cellular location into one of four classes: cytosolic, secreted, nuclear and mitochondrial. The approach taken is to classify proteins using features derived from their amino acid sequences. The data-set used is a pre-labelled ensemble of 9,158 non-homologous eukaryotic protein sequences. Using a Random Forest classifier scores of approximately 65% are achieved for accuracy, precision, recall and F1.

1 Introduction

The data for the study consists of a pre-labelled ensemble of 9,222 non-homologous eukaryotic protein sequences in FASTA format. The sequences are labelled as belonging to one of four classes:

- cytosolic (within the cell itself, but not inside any organelles),
- secreted (transported out of the cell),
- nuclear (found/used within the cell's nucleus), and
- mitochondrial (transported to the cell's mitochondria)

Protein sequences in the FASTA format may contain non-standard amino acid codes which represent for example, a choice between 2 amino acids (e.g. B denotes Aspartic Acid or Asparagine), or that the amino acid is not specified at a given position (the letter X). The are 64 protein sequences in the ensemble which contain at least one of the non-standard amino acid codes: B, J, O, U, X and Z. These sequences are removed from the data-set since they can cause issues when calling standard software routines with the sequences, leaving a data-set of size 9,158 protein sequences. Finally, it is noted that there is a single outlier protein sequence Caenorhabditis elegans[1] (DIG1_CAEEL) with length 13,100, which is approximately 24 standard deviations from the mean sequence length for secreted proteins; therefore it is removed from the data set, leaving a final total of 9,157 sequences.

The class frequencies for the sequences, along with some summary properties, are shown in Table 1.

2 Methodology

In order to classify protein sequences into their sub-cellular locations, a Random Forest [1] classifier is trained using a set of features of the protein sequences. The features are a combination of physical and structural properties that can be derived using just the sequence of amino acids in a protein; the features, along with some of their properties, are described below.

[1] `http://www.genome.jp/dbget-bin/www_bget?sp:DIG1_CAEEL`

2.1 Features

Sequence length

The first feature used is the sequence length, i.e. the count of the number of amino acids in the protein. The summary statistics for the sequence lengths are given in Table 1. For the given data-set, it is clear that on average, mitochondrial and secreted proteins are shorter than nuclear and cytosolic proteins by ratio of about 1:2.

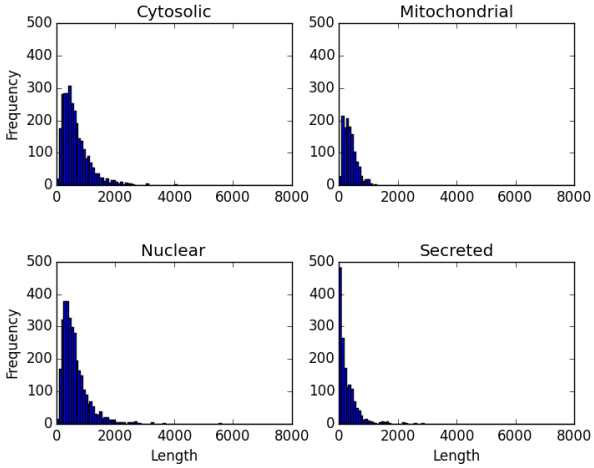

Figure 1: Distribution of protein sequence length, by label.

	Class size	Min	Max	Mean	Median	Sd
Cyto	2,998	19	7,393	665	523	548
Mito	1,296	36	2,628	376	330	247
Nuclear	3,312	35	5,596	624	496	501
Secreted	1,551	11	5,100	304	178	413

Table 1: Summary properties of sequence lengths.

Proportion of Amino Acids

The second feature used is the proportion of individual amino acids in each sequence type; the distribution is shown in Figure 2. There are no clear patterns in the distribution, but it is possible to determine that secreted proteins have proportionally higher levels of amino acids Glycine (G) and Cysteine (C) and lower levels of Leucine (L), then the other proteins types.

Molecular Weight

The third feature used is the molecular weight of the protein sequence. The distribution of molecular weight for each sequence type is shown in Figure 3. The relative molecular weights of the different protein classes are in rough proportion to their lengths.

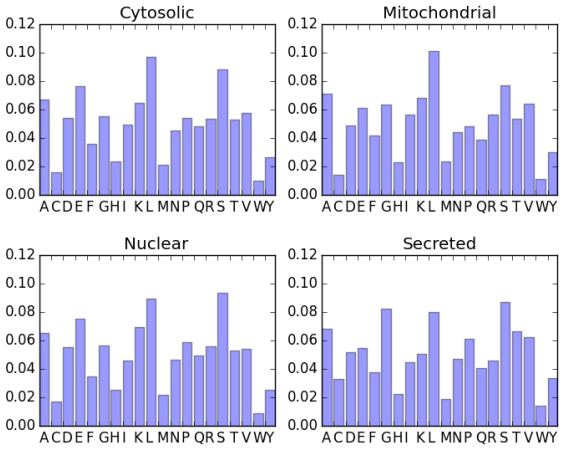

Figure 2: Distribution of amino acids for each class label, as a percentage of all amino acids in the sequence.

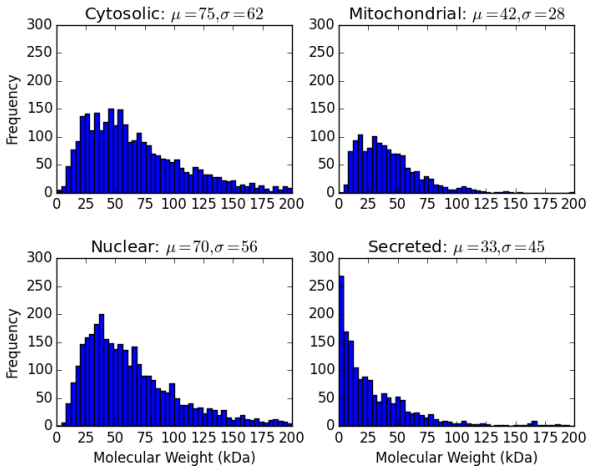

Figure 3: Distribution of protein molecular weight in kDa, by label, with mean and standard deviation.

Instability Index

The fourth feature used is the protein instability index [2]. The distribution of instability index for each sequence type is shown in Figure 4.

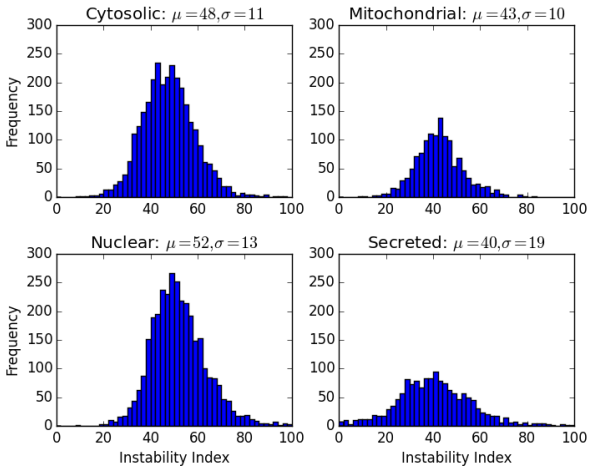

Figure 4: Distribution of protein instability index, by label, with mean and standard deviation.

Isoelectric Point

The fifth feature used is the protein isoelectric point (pI), the pH at which a particular molecule carries no net electrical charge. The distribution of isoelectric point for each sequence type is shown in Figure 5.

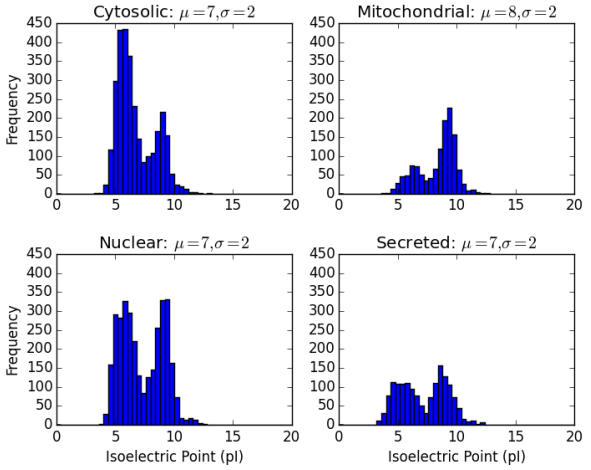

Figure 5: Distribution of protein isoelectric point, by label, with mean and standard deviation.

Secondary Structure Fraction

The sixth feature used is the protein secondary structure fraction. The secondary structure fraction is a 3-tuple (h, s, t) where h is the proportion of amino acids in the protein which are in helix (i.e. amino acids V, I, Y, F, W, L), s is the proportion of amino acids which are in sheet (i.e. amino acids N, P, G, S), and t is the proportion of amino acids in turn (i.e. amino acids E, M, A, L). A plot of secondary structure fractions for each class is given in Figure 6.

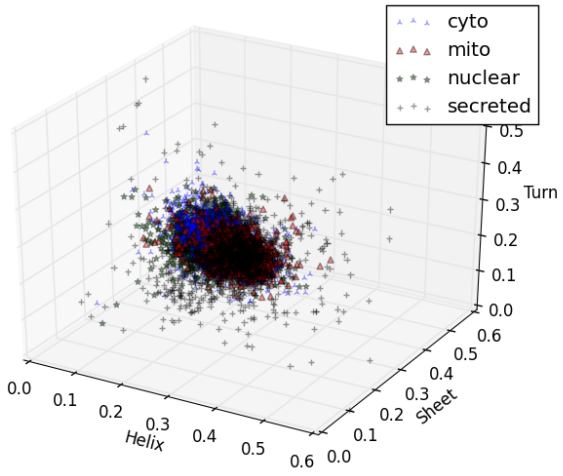

Figure 6: Plot of secondary structure fraction for protein sequences.

Grand Average of Hydropathy (GRAVY)

Another feature used is the Grand Average of Hydropathy (GRAVY) [3]. The GRAVY of a protein is calculated as the length normalised sum of the hydropathy values of each amino acid in the protein sequence. The distribution of the GRAVY for each sequence type is shown in Figure 7.

Sequence Start/End Point Distribution

A further feature used is the distribution of the 20 amino acids at the start and the end of the protein sequence. Rather than choose a fixed window for this feature, the distribution is calculated using a window length proportional to the length of the sequence[2]. The proportion is chosen by grid-search and 10-fold cross-validation.

Taylor Venn Diagram Categories

The final feature used is computed using the Taylor Venn Diagram for protein properties. The Taylor Venn Diagram categorises proteins into Hydropathic, Tiny, Small, Aromatic, Aliphatic, Polar and Charged (Positive or Negative). For each protein sequence, a count is made of the number of properties that are encountered for each amino acid in the sequence.

[2]window length min=5, max=250

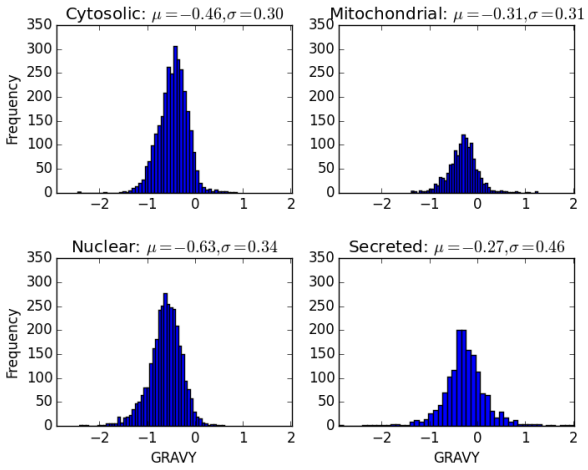

Figure 7: GRAVY, by label, with mean and standard deviation.

For example Proline (P) counts 1 for Small, whereas Tryptophan (W) counts 1 for Aromatic, 1 for Hydrophobic and 1 for Polar. The property count for each sequence is thus an eight element vector (only Positive is counted since it is mutually exclusive with Negative), which is normalised by the length of the sequence.

Collected together the features constitute a 75 element vector; the full set of features is summarised in Table 2.

Feature	**Size**	**Index**
Sequence length	1	1
Amino Acid Percentage	20	2-21
Instability Index	1	22
Isoelectric Point	1	23
Secondary Structure Fraction	3	24-26
GRAVY	1	27
Sequence Start Distribution	20	28-47
Sequence End Distribution	20	48-67
Taylor Venn Diagram	8	68-75
Total	75	

Table 2: Features and feature lengths

2.2 Random Forest Classifier

A Random Forest classifier constructs a large number of Decision Trees using random samples, drawn with replacement, from the training data, and by choosing a random subset of the features at which to split the data. Random Forests can produce very accurate classifications, by exploiting the low bias of Decision Trees, but also reducing their tendency to suffer from high variance; they can also be implemented in a highly parallel manner. The Decision Trees in the forest can be split until there is some minimum number of samples per

leaf node; this min. number of sample per leaf parameter, as well as the number of trees in the forest will be chosen by grid search.

The training data is split 90% into a development set and 10% into a hold-out test set. The test set is chosen in a stratified way to reflect the original distribution of the data. A grid search is then performed using 10-fold cross-validation on the development set, over the parameter space of: the number of estimators (trees); the min. number of samples per leaf, and the sequence start/end distribution window size. The grid search returns the best value of the parameters, chosen by the highest F1 score, defined as:

$$\mathrm{F1} = 2 \times \frac{\mathrm{precision} \times \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

The Random Forest classifier is then re-estimated using the training data and the best parameters found by the grid search, and used to estimate the labels for the test set.

3 Results

The confusion matrix for the test set is shown in Table 3. These are obtained using the best parameter estimates of min. number of samples per leaf = 10, sequence start/end distribution window size = 0.01 (i.e. 1%) and the number of estimators (trees) = 400, as obtained from the grid search.

	cyto	**mito**	**nuclear**	**secreted**
cyto	349	36	196	19
mito	50	164	30	15
nuclear	179	18	457	9
secreted	41	19	29	222

Table 3: Confusion matrix resulting from classifying test set, with scores precision: 0.6569, recall: 0.6507, accuracy: 0.6507, F1: 0.6524

The Random Forest classifier produces scores of precision: 0.6569, recall: 0.6507, accuracy: 0.6507, F1: 0.6524 on the test set. The confusion matrix highlights that there appear to be issues in distinguishing between cytosolic and nuclear proteins. For the feature characteristics plotted above, cytosolic and nuclear proteins appear very similar, and this is worth further investigation. The top 50% (38 features) of feature importances are shown in Figure 8 and are calculated from the drop in entropy achieved (the information gain) when splitting the tree on the feature, averaged across all the trees in the forest. Of the top 50% of features, Table 4 shows the top ten feature importances by name.

By running a one vs all classifier it is also possible to estimate the importance of features for distinguishing each class against the rest. The best parameters from the grid search are used to estimate a Random Forest classifier in a one vs all manner, and the feature importances are extracted for each class label; the results are shown in Figure 9. The top 5 features, by importance, for each class are show in Table 5.

4 Conclusion

This article has shown that it is possible to use Random Forests, a well-known and computationally efficient machine learning classifier, to obtain accuracy of approximately 65% when classifying protein sub-cellular location into one of cytosolic, mitochondrial, nuclear and secreted, using just the protein amino acid sequence information. Of the 75 features constructed for the classifier only a subset appear to show any significance; it would be

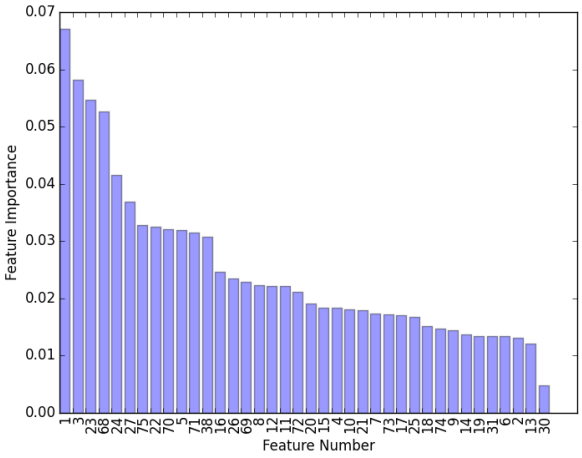

Figure 8: Top 50% of Feature Importances.

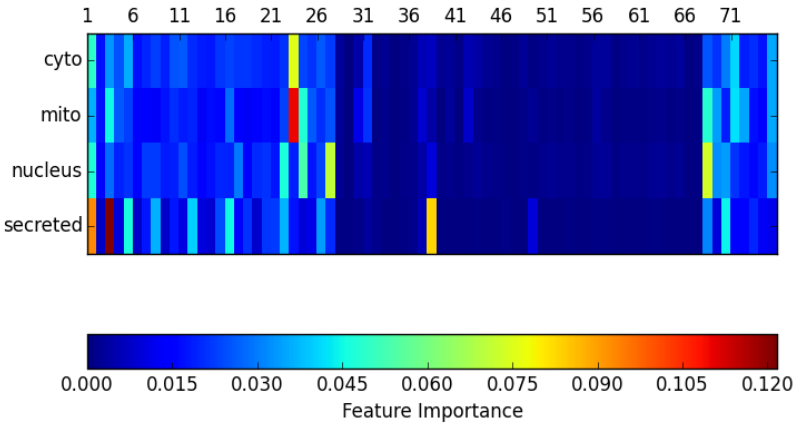

Figure 9: Feature importances from one vs all classification.

Feature Index	Feature
1	Sequence length
3	Percentage of Cysteine in overall sequence
23	Isoelectric Point
68	Hydrophobicity count (Taylor Venn Diagram)
24	Helix
27	GRAVY
75	Aliphatic count (Taylor Venn Diagram)
22	Instability Index
70	Charge count (Taylor Venn Diagram)
5	Percentage of Glutamic Acid in overall sequence

Table 4: Top 10 feature importances, with description.

	Cyto	Mito	Nuclear	Secreted
1	Isoelectric Point (23)	Isoelectric Point (23)	Hydrophobicity (68)	%age Cysteine (3)
2	Sequence length (1)	Hydrophobicity (68)	GRAVY (27)	Sequence length (1)
3	Negative charge count (71)	Helix (24)	Helix (24)	Methionine level at start (38)
4	%age Glutamic Acid (5)	%age Cysteine (3)	Sequence length (1)	%age Glutamic Acid (5)
5	Aliphatic count (75)	Negative charge count (71)	Instability Index (22)	%age Arginine (16)

Table 5: Top 5 features per class label

interesting to understand the biological significance of these features for each sub-cellular location. Another avenue of study would be to examine and compare the performance of logistic regression or support vector machines to the Random Forest classifier, or use the subset of features highlighted as significant by the Random Forest as inputs to these other classifiers. Another possibility is to engineer more features. For example, by viewing amino acids as letters in an alphabet, the occurrences of n-gram sequence of amino acids could be considered as a feature; this is done in [4] however as the authors point out, the dimensionality of the n-gram sequence is exponential in n and thus the feature space can become very large.

A Blind Predictions

The predictions for the blind test proteins are given in Table 6. The confidence figure is the predicted class probability of the sequence, computed as the mean predicted class probability across all the trees in the forest.

B The Code

The code[3] for the classifier is written in Python 3 using the `BioPython`[4] and `sklearn`[5] packages. The main modules are:

- `feature_transforms.py` - the features described above, implemented as `sklearn` feature transforms,

[3]The code is available separately.
[4]`http://biopython.org/wiki/Main_Page`
[5]`http://scikit-learn.org/stable/`

Sequence ID	Predicted Label	Confidence
SEQ677	cyto	54%
SEQ231	secreted	43%
SEQ871	nucleus	38%
SEQ388	secreted	35%
SEQ122	cyto	35%
SEQ758	nucleus	63%
SEQ333	mito	48%
SEQ937	cyto	59%
SEQ351	cyto	51%
SEQ202	mito	31%
SEQ608	mito	45%
SEQ402	nucleus	40%
SEQ433	secreted	67%
SEQ821	cyto	33%
SEQ322	nucleus	77%
SEQ982	nucleus	73%
SEQ951	cyto	37%
SEQ173	cyto	48%
SEQ862	mito	34%
SEQ224	cyto	42%

Table 6: Blind protein sequence predictions, with confidence level.

- `predictor.py` - implementation of the Random Forest predictor using the `sklearn RandomForestClassifier`
- `grid_search.py` - grid search for best fit parameters using 10-fold cross-validation
- `blind_predict.py` - builds a classifier using the parameters found by grid search and estimates the labels of the blind protein sequences. The results are written to the file `blind_predictions.txt`, and the file `blind_all.txt`, which contains a probability distribution over the labels for each element in the blind sequence.
- `one_vs_all.py` - performs a one vs all classification using a Random Forest, with the best parameters as found by `grid_search.py`.

To run the grid search, issue the command: `python grid_search.py`. To run the blind predictions, issue the command: `python blind_predict.py`.

References

[1] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001.

[2] Kunchur Guruprasad, BV Bhasker Reddy, and Madhusudan W Pandit. 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*, 4(2):155–161, 1990.

[3] Jack Kyte and Russell F Doolittle. A simple method for displaying the hydropathic character of a protein. *Journal of molecular biology*, 157(1):105–132, 1982.

[4] Wen-Yun Yang, Bao-Liang Lu, and Yang Yang. A comparative study on feature extraction from protein sequences for subcellular localization prediction. In *Computational Intelligence and Bioinformatics and Computational Biology, 2006. CIBCB'06. 2006 IEEE Symposium on*, pages 1–8. IEEE, 2006.