Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: 2 April 2015
Applications Note

OXFORD

Subcellular location Prediction

Eukaryotic proteins' subcellular location prediction using Random Forests

Oliver Wesely (18057603) [1]

[1] Department of Computer Science, University College London, London, WC1E 6BT, Great Britain.

Abstract

Motivation: Subcellular location prediction is crucial to the understanding of protein function and is cheaper and faster than location finding with experiments. A variety of methods have been proposed in recent years to classify proteins and to extract different features to do so. This paper focuses on the method of n-Gram Descriptors' Frequency in combination with feature selection methods to train a model with the most significant features. The classification model used in this paper is a Random Forest Classifier.
Results: Using a Random Forest Classification Model with 500 trees achieved an accuracy level of about 0.65 and an average MCC value of about 0.5 on over all 5 fold cross validation subsets. The most important features in the Random Forest include the Secondary Structure (H), the isoelectric Point, the amino acid composition of D in the first 10% of a protein and the 2-gram 'EV'.
Contact: oliver.wesely.18@ucl.ac.uk

1 Introduction

Bioinformatics has been changing a lot since the last few decades, as it is gaining from computer science technology and the attention from computer scientists and biologists research community.
One of the fundamental goals in computational biology, cell biology, and proteomics is to identify the function of new proteins as it is one of the most fundamental element of any living organism. Typical proteins are comprised of 20 amino acids that carry out an important role in cell functions. Localization of proteins in appropriate compartments is vital for the function and integrity of the internal structure of the cell as protein's location is correlated with its molecular function, which is a key step to understand the biological functions of protein sequences. There have been different ways explored to acquire better prediction performance. Typically the development in this area follows two trends: sequence-based and database annotation-based.
Sequence based discriminative prediction attempts to extract increasingly more characteristic subsequence features from protein sequences and performs prediction based on these. This method can be further divided into three sub-categories: prediction based on amino acid composition; prediction via known targeting sequences and prediction based on other novel extracted features.
The second trend of prediction of protein subcellular location arises from the protein annotation databases. These annotation databases become increasingly more capable to supply reliable clues, such as motifs and gene ontology, therefore it becomes more powerful to use them. [6] [13] [5]

In this paper we use Random Forest predictors as our classification model, we carry out different sequence-based feature creation methods including a combination of n-Gram Descriptors and other physiochemical properties of the proteins. Furthermore we apply a feature selection method on all n-Gram Descriptors to get a reasonable number of features which are used to train the Random Forest. The Random Forest then undergoes hyperparameter optimization to find the optimal number of trees.
This paper is organized as follows: Section 2 describes the dataset we are working with. Section 3 will give an overview of the feature extraction methods we are using and will explain n-Gram Descriptors in more detail. In Section 3 we will also briefly explain Random Forests and the evaluation methods we are going to use. Section 4 should then present our results and some interpretations about them. Finally, we will end the paper with some further discussions and our conclusion in Section 5 and 6 respectively.

2 Dataset

The provided dataset is split in a training set and a test set including eukaryotic protein sequences located in all four major protein locations, as mentioned before. The training sample size is 9222 which are classified to their corresponding location as follows:

Cytosolic Proteins: 3004
Mitochondrial Proteins: 1299
Nuclear Proteins: 3314
Secreted Proteins: 1605

Each of these proteins is a sequence of consecutive amino acid residues.

The length of each of these sequences is extremely variable (between 9 and 13100 amino acids). The test set includes 20 amino acid sequences and is called the 'blind' test, for which the locations have to be predicted after fitting the model. All listed amino acids in table 1 are those which the provided proteins consist of.
However, only the 20 standard amino acid abbreviations will be considered within the prediction, as the remaining 4 in this list (B,U,Z,X) might not change our results significantly. As such the provided data was preprocessed before it is utilized in training and test.

A	alanine	N	asparagine
B	aspartate/asparagine	P	proline
C	cystine	Q	glutamine
D	aspartate	R	arginine
E	glutamate	S	serine
F	phenylalanine	T	threonine
G	glycine	U	selenocysteine
H	histidine	V	valine
I	isoleucine	W	tryptophan
K	lysine	Y	tyrosine
L	leucine	Z	glutamate/glutamine
M	methionine	X	any

Table 1. Amino Acid Codes

3 Methods

3.1 Feature Extraction

The representation of the sequence of the twenty amino acids in the form of a number of numeric features is an ongoing problem in bioinformatics including machine learning methods. Many different feature extraction methods have been developed so far and can be used to predict the subcelluar location of a protein. The feature extraction in this paper will focus on the sequence encoding by the combination of n-Gram Descriptors' Frequency, which is a more general version of amino acid composition, which not only look at local but also on global interaction. Other important features which also will be included in the model are the length of a protein, the molecular weight of a protein, the aromaticity value of a protein, the instability index, the isoelectric point, the secondary structure and local amino acid composition especially near the start and the end of the sequence.

3.1.1 Combination of n-Gram Descriptors' Frequency

In this section we describe a sequence encoding method using combinations of different n-gram descriptors to extract valuable features from a protein. By applying this encoding method we can extract either global and local features from a protein. In this paper we will focus on descriptors of length 1, 2 and 3, as of descriptors with bigger length would increase the computational cost exponentially.
Set $X_i,\ i = 1, ..., n$ to be the set of protein descriptors of the length i amino acids. In our case example descriptors look as follows:

$$\begin{aligned} X_1 &= \{A, C, ..., Y\}, \\ X_2 &= \{AA, AC, ..., YY\}, \\ X_3 &= \{AAA, AAC, ..., YYY\} \end{aligned} \tag{1}$$

The general total number of protein descriptors is:

$$N_{Total} = \sum_{i=1}^{n} 20^i = \frac{20 - 20^{n-1}}{1 - 20}. \tag{2}$$

Hence, in our case of n=3 we get $N_{Total} = 20 + 20^2 + 20^3 = 8,420$ possible combinations of amino acids we will look at to find significant features.
In order to make it a bit clearer I am showing the procedure using an example protein sequence 'ACDGIKAD' which would evolve the following descriptors for this specific example:

$$\begin{aligned} X_1 &= \{A, C, D, G, I, K\}, \\ X_2 &= \{AC, CD, DG, GI, IK, KA, AD\}, \\ X_3 &= \{ACD, CDG, DGI, GIK, IKA, KAD\} \end{aligned} \tag{3}$$

.
Any protein descriptors with bigger length could be similarly attained. But using a descriptor length up to three will already give rise to 8,420 features which already is computational expensive to derive all of them. Though many of these features will have zero values and some may be redundant or irrelevant for the classification algorithm and will only affect the algorithms performance and computational time. Hence we will use a feature selection technique to remove all features of the original feature space which do not contribute in the representation of a sequence. To do so we evaluate the statistical significance of all features, more precisely the feature selection method will look at subsets of the original feature space and select the one including the best features to represent the protein sequences, which means that shows maximum accuracy. This technique will greatly reduce computational time in training the model and the chance of overfitting. The following steps are performed to select features capable of seperating different superfamilies, in our case four different subcellular locations.
Suppose Z_i is the i^{th} sequence superfamily including $k = 1, ..., N_i$ sequences, denoted by Z_i^k, representing the k^{th} sequence. The feature vector representing this sequence is $Z_i^k(j)$ with $j = 1, ..., 8420$.
We then have to calculate a few statistical values to be able to reduce our features space. For each superfamily we will calculate the mean vector:

$$\bar{Z}_i(j) = \frac{\sum_{k=1}^{N_i} Z_i^k(j)}{N_i}, j = 1, ..., 8420 \tag{4}$$

and the variance:

$$S_i^2(j) = \frac{\sum_{k=1}^{N_i} (\bar{Z}_i(j) - Z_i^k(j))^2}{N_i - 1}. \tag{5}$$

After that we will use these values to calculate the distance for each pair of superfamilies (say p and q) using the following metric:

$$v_{p,q}(j) = \frac{\bar{Z}_p(j)\bar{Z}_q(j)}{\sqrt{(S_p^2(j)/N_{Total}) + (S_q^2(j)/N_{Total})}}, \tag{6}$$

which is going to be a matrix of the size $6 \times N_{Total}$ as we have 6 pairs of superfamilies in this paper, as

$$\begin{aligned} (p, q) \in \{&(C, M), (C, N), (C, S), (M, N), (M, S), (N, S)\} \\ C &= \text{Cytosolic Proteins} \\ M &= \text{Mitochondrial Proteins} \\ N &= \text{Nuclear Proteins} \\ S &= \text{Secreted Proteins} \end{aligned} \tag{7}$$

and a cardinality of six. As the final metric we will take the minimum of each of these six distances as follows:

$$v(j) = \underset{p \neq q}{Min}\{v_{p,q}(j)\}, j = 1, ..., 8420. \quad (8)$$

As our last step we take those features which are capable to explain our superfamilies best and therefore correspond to the highest values of $v(j)$. The number of the features we want to use can be chosen and in this paper we will take the 20 most significant features to train our model.
As a short summary we first created n-gram descriptors including length n=1,2,3 to get 8,420 features and then used a feature selection method to get our final most important 20 features to train the model.[6]

3.1.2 Other extracted Features from protein sequences - Chemical Properties

Physiochemical properties of protein are intent by analogous properties of the amino acid in it. These different properties control the structure and therefore the function of proteins. It is obvious that proteins belonging to the same subcellular location must share some properties.

- Protein length: Number of Amino Acids forming a protein sequence.
- Molecular weight: The mass of a molecule which can be determined by multiplying the atomic mass of each atom with the number of given atoms. [7]
- Aromaticity value: This value represents the relative frequency of aromatic amino acids, which include phenylalanine, tryptophan and tyrosine and therefore is,

$$\sum_{j \in [F,W,Y]} x_{ij} \quad (9)$$

  for a given protein i. [12]
- Instability Index: In the primary structure and the intrinsic property of a protein can be shown an appearance of correlation between its sensitivity to in vivo degradation and certain dipeptides presence. The whole influence of these dipeptides contributes to instability or stability characteristics of proteins. [8]
- Isoelectric point: This point is where a amino acid does not carry any electrical charge in the statistical mean and where the pH of it is neutral. [7]
- Secondary structure: There are four levels of protein structure including primary, secondary, tertiary and quarternary structure. The protein secondary structure(PPS) includes three basic elements: -helices (H), β-strands (E) and coils (C). PPS plays an important in modeling protein structures because it represents the local conformation of amino acids into regular structures. [10]
- Local amino acid composition: In general the amino acid composition is a vector of size 20, which represents all standard amino acids as defined before. Assume this feature vector x_i for the i^{th} protein has labels y_j corresponding to all different 20 amino acids, therefore $j = 1, ..., 20$. Considering the composition as the number of occurrence in each protein, we get: $x_{ij} = count_i(j)$, which means counting all different amino acids j in a specific sequence i. [13] Local composition, in our case in the first and the last 10% of the amino acid sequence of a protein, means therefore only looking at this local parts of a protein sequence and calculate the amino acid composition in these parts separately.

3.2 Random Forest Classifier

In this paper we will use random forests as the classification method. Random forests are typically built by combining the predictions of several trees which were trained independently whereas its overall prediction is a combination of all of them through averaging. [4] In general a tree is a special type of graph which structures data within a collection of nodes connected via edges in a hierarchical fashion. The used nodes are divided into two groups: internal/split nodes and terminal/leaf nodes. All nodes have exactly one incoming edge, but the number of output edges is not certainly clarified, but could be fixed to two, for example in the binary tree, where each internal node has two outgoing edges. As within our example the classification method is used to output probabilities to any possible classes. Therefore the tree will now be called decision tree and given an input on the very top of the tree it will estimate unknown properties of the input variable by asking questions about its known properties/features, where the following question depends on the last answer and the final decision will then be made in a terminal node at the end of a path. Such a tree will then be trained with several training data to evolve its questions to get reasonable output decisions. Decision trees are characterized by components including the family of weak learners (test functions), energy model, the leaf predictors and the randomness influence of the prediction, which we will not explain into detail at this point.
A random decision forest, also called ensembles of trees, is an ensemble of randomly trained decision trees. The key aspect is that its component trees are all randomly different from one another. Which leads to uncorrelated individual tree predictions and therefore results in improved generalization and robustness of the model. All trees of a forest will be trained independently. Due to this fact we can easily achieve high computational efficiency by testing the trees in parallel. The final tree predictions are often just the simple average over all trees.
The key model parameters in a decision forest are:

- The forest size T (number of trees),
- The maximum allowed tree depth D,
- The amount of randomness and its type,
- The choice of weak learner model,
- The training objective function,
- The choice of features in practical applications.

All these factors affect the predictive accuracy, its generalization and its computational efficiency.
For instance, the testing accuracy increases monotonically with the forest size T, which we will look at as a hyperparameter in this paper. Thanks to all trees being different from each other, an increasing forest size T will produce much smoother decision boundaries. [3]
Several other reasons come up why we use a Random Forest Classification as our prediction model including that random forest do not overfit as more trees are added, but produce a limiting value of the generalization error. Regarding categorical variables random forests have one important advantage which is the difficulty of what to do with categoricals that have many values. In the two class problem the search for the best categorical split can be reduced to an $O(I)$ computation, but for more classes it is an $O(2^{I_1})$ computation. But using a random forest implementation this only involves selecting a random subset of categories and is therefore computationally cheaper.[2]

3.3 Evaluation

3.3.1 K-Fold Cross Validation

To evaluate our approach we are using K-fold cross validation, to be more precisely a 5-fold test, which divides each superfamily of the data into five subsets of approximately equal size. After that five different trainings and tests will be carried out using 4 subsets to train on and the remaining fifth subset to test the model on. The final performance measures will then be obtained by taking the mean over the results of all five outputs. [13]

3.3.2 Performance Measurements

We are using different measures to assess our approach performance. These include Precision (Pr) which represents the classifiers ability not to label

a negative sample as positive and therefore will also sometimes be called as the random error, the Sensitivity/Recall (Se) measure which reflects the performance of classifier interface residues and is the classifiers ability to find all positive samples, the F-measure (F) indicates the harmonic mean of Sensitivity and Precision, Accuracy (Ac) which reflects the prediction ability of the classifier for the test set and Matthews correlation coefficient (MCC) demonstrating the correlation between prediction results and real data:

$$Pr = \frac{TP}{TP + FP} \tag{10}$$

$$Se = \frac{TP}{TP + FN} \tag{11}$$

$$F = 2 \times \frac{Se \times Pr}{Se + Pr} \tag{12}$$

$$Ac = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN)(TP + FP)(TN + FP)(TN + FN)}} \tag{14}$$

where TP represents the number of correctly classified interface residues, FP the number of incorrectly classified non-interface residues, TN the number of correctly classified non-interface residues and FN the number of wrongly classified interface residues. [11]

Regarding the derivation of these values we will always use the macro-averaging method, which calculates metrics for each label and finds their unweighted mean in contrast to the micro-averaging method which just takes the overall average, whereas the macro method does not take the label imbalance into account.

Another method to visualise a classifier's performance is the Receiver Operating Characteristic (ROC) curve. However when we have to compare different classifications it is often not easy to compare ROC curves and therefore it is desirable to obtain single figures, which often is a cross-validated estimate of the classifier's overall accuracy, but we are going to use the area under the ROC curve (AUC) as a single value classifier performance measurement. [1]

The ROC and AUC analysis is a standard practice for the design of two-class pattern recognition systems. Advantages include decision boundary adaptation to imbalanced missallocation costs, the ability to fix some classification errors, and performance evaluation in imprecise, ill-defined conditions where costs, or prior probabilities may vary. To be able to apply ROC and AUC analysis to our multi-class classification problem we have to use a general version of ROC curves which uses pairwise ROC analysis and then approximates the multi-class ROC curve.[9]

4 Results

First we evaluate all n-Gram Descriptors of length 1,2,3 and all other physiochemical features for the whole data set. After that we used a feature extraction method, explained in the last section, to get the 20 most significant n-Gram Descriptor features out of a total number of 8,420. Within these most important features one feature stands out with a high importance value, which is the 2-Gram Descriptor 'KR'. The 10 most important n-Gram Descriptors are:

$$\{LEQ, KRK, LR, EL, KK, RKK, KKK, LE, RK, KR\}.$$

To predict the location of an unknown protein we want to avoid overestimating performance, therefore redundancy between test set and training set were reduced by using a 5-fold cross validation test. In 5-fold cross validation we divide the available data in 5 subsets, analysis was then performed on each possible combination using 4 subsets to train our random forest and one as test data. We then validate our models with the test data via deriving several performance measurements we already explained in the last section. To do so we have to know that predicted class probabilities of a Random Forest of an input sample are computed as the mean predicted class probabilities of the trees in the forest itself, which can be used as a measure of confidence of the prediction. The class probability of a single tree is the fraction of samples of the same class in a leaf. The final predicted class is the one with the highest mean probability estimate. One other thing we want to do in this paper is to optimize the number of trees used in our random forest. Therefore we treat the number of trees T as a hyperparameter and will try to find a reasonable value of T by analysing the performance of the models. We therefore implemented a test which runs 20 times for each of the following T values: 1, 10, 50, 250, 500 and for all 5 folds of the cross validation. Estimating these models we can show the results using boxplots and validate our models via Precision, Sensitivity, F-measure, Accuracy and MCC, which can be found in figures 1, 2, 3, 4 and 5 respectively.

In all of the 5 boxplots we can basically see the same pattern, while all measurements are pretty low when using only 1 tree in the classification model increasing it to 50 instead would already give a pretty good model. Although the difference between models using a tree number of 50, 250 and 500 is very little. The model average of the measurements for the three highest T values is a Precision of about 0.67, Sensitivity 0.64, F-measure 0.66, Accuravy 0.65 and a MCC of about 0.5.

To get even more detailed results we will run each of the three models one more time using only one 5-fold cross validation split and show the results using pairwise ROC curves and combining them to a Multi-class ROC curve. To be able to compare these curves we will evaluate the area under each curve and will interpret a higher value as a higher performance. As you can see in figure 6, 7 and 8 we get again, as expected, similar results using different number of trees. In general we can say that Cytosolic Proteins are classified the worst in our model, whereas Secreted Proteins the best, as by the AUC value. Looking at the macro-average ROC and it area under the curve value we can see a slight better value using 500 trees.

We can also have a look at the Performance Measurement values of these 3 implementations in Table 1 which underline that T=500 performs best having an accuracy of about 0.69 and MCC of 0.56.

One further measurement we provide are all three Confusion matrices, also known as error matrices, to show the model estimations in more detail. Within each confusion matrix the number of correct and incorrect predictions are summarized and broken down by each class. Each row of the matrix corresponds to a predicted class, whereas each column corresponds to the actual class of the protein. In these matrices we can see a confusion between cytosolic and nuclear proteins in both directions with a miss-classification number of about 150-180. Looking at mitochondrial proteins a significant chunk was estimated as cytosolic proteins, and same for secreted proteins where a big amount of proteins was estimated as cytosolic. All in all we can say that our model with 500 trees is a little better than the others.

One more interesting analysis can be done using a feature importance analysis to show which features are the most relevant within the random forest model. Starting with the most relevant on, in our three estimations the 10 most important features we get within our models are:

- Secondary Structure (H)
- Isoelectric Point
- 'D' - AAC first 10%
- Instability Index
- Molecular Weight
- Sequence Length
- Aromaticity

- Secondary Structure (E)
- 2-gram 'EV'
- 'C' - AAC first 10%

4.1 "Blind Test" Proteins

As our final validation we are using the best parameter settings to train a Random Forest Classifier on our whole training data set to classify the following 20 proteins, which have unknown subcellular locations, and provide our class predictions and the corresponding confidence:

SEQ677 Cyto Confidence 36%	SEQ231 Secr Confidence 34%
SEQ871 Nucl Confidence 35%	SEQ388 Nucl Confidence 71%
SEQ122 Nucl Confidence 68%	SEQ758 Nucl Confidence 90%
SEQ333 Nucl Confidence 46%	SEQ937 Cyto Confidence 60%
SEQ351 Cyto Confidence 42%	SEQ202 Mito Confidence 70%
SEQ608 Mito Confidence 77%	SEQ402 Mito Confidence 64%
SEQ433 Secr Confidence 38%	SEQ821 Secr Confidence 64%
SEQ322 Nucl Confidence 89%	SEQ982 Nucl Confidence 81%
SEQ951 Cyto Confidence 42%	SEQ173 Cyto Confidence 46%
SEQ862 Mito Confidence 67%	SEQ224 Cyto Confidence 46%

Table 1. Performance Measurements

	T = 50	T = 250	T = 500
Accuracy	0.6531	0.6873	0.6916
Precision	0.6967	0.7225	0.7327
Sensitivity	0.6433	0.6769	0.6789
F-measure	0.6644	0.6954	0.7002
MCC	0.5084	0.5576	0.5633

Table 2. Confusion Matrix - 50 Trees

True \ Est.	Cyto	Mito	Nucl	Secr
Cyto	378	22	185	16
Mito	70	151	28	11
Nucl	175	22	462	4
Secr	55	20	32	214

Table 3. Confusion Matrix - 250 Trees

True \ Est.	Cyto	Mito	Nucl	Secr
Cyto	399	25	162	15
Mito	63	161	25	11
Nucl	152	21	486	4
Secr	51	17	31	222

Table 4. Confusion Matrix - 500 Trees

True \ Est.	Cyto	Mito	Nucl	Secr
Cyto	405	23	161	12
Mito	71	156	23	10
Nucl	151	19	489	4
Secr	51	12	32	226

5 Discussion

The above results show that a combination of specific physiochemical and n-Gram Descriptors' leads to a powerful classification model, as evidenced by the different performance measurements reported. Although there are other approaches to find an appropriate solution to this problem by applying machine learning methods. In this study four major subcellular protein locations were taken into account. After getting a high amount of features by using n-Gram descriptors we significantly reduced the size due to a statistical metric. After a hyperparameter search in our Random Forest including different measurements to evaluate the results we found the optimal parameter setting by looking at the ROC values, AUC values and the confusion matrices, which is T=500. Further analysis showed that most of the physiochemical features, two of the local features at the beginning of a protein chain and the 2-gram 'EV" are the 10 most important features in our random forest model.

6 Conclusion

Although we get proper results using the explained methods we do not get to actual accuracy benchmark levels. To reach that goal we suggest to use n-Gram descriptors of even higher length in combination with GPU to make it computationally not to expensive and use a bigger feature number used for the model. I would also try different classification models as of LSTMs or Neural Networks to get higher confidence about our predictions.

References

[1]Andrew P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *PATTERN RECOGNITION*, 30(7):1145–1159, 1997.

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

[3]Antonio Criminisi, Jamie Shotton, and Ender Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Found. Trends. Comput. Graph. Vis.*, 7:81–227, 2012.

[4]Misha Denil, David Matheson, and Nando De Freitas. Narrowing the gap: Random forests in theory and in practice. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32, pp. 665–673, 2014.

[5]Geetha Govindan and Achuthsankar Nair. Composition, transition and distribution (ctd) âŁ" a dynamic feature for predictions based on hierarchical structure of cellular sorting. *Proceedings - 2011 Annual IEEE India Conference: Engineering Sustainable Solutions, INDICON-2011*, 12 2011.

[6]Muhammad Javed Iqbal, Ibrahima Faye, Samir Brahim, and Abas Md Said. Efficient feature selection and classification of protein sequence data in bioinformatics. *TheScientificWorldJournal*, 2014:173869, 06 2014.

[7]Shalini kaushik et. al. Prediction of protein subcellular localization of human protein using j48, random forest and best first tree techniques. *JOURNAL OF ADVANCED APPLIED SCIENTIFIC RESEARCH*, 1(12), 2017.

[8]Madhusudan W. Pandit Kunchur Guruprasad, B.V. Bhasker Reddy. 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), 1990.

[9]Thomas C. W. Landgrebe and Robert P. W. Duin. Approximating the multiclass roc by pairwise analysis. *Pattern Recogn. Lett.*, 28(13):1747–1758, October 2007.

[10]Ho SY Hsu WL Lin HN, Sung TY. Improving protein secondary structure prediction based on short subsequences with local structure similarity. *BMC Genomics*, 11(4), 2010.

[11]Bingqiang Liu, Xiaoying Wang, Cheng Chen, Bin Yu, Anjun Ma, and Qin Ma. ProteinâŁ"protein interaction sites prediction by ensemble random forests with synthetic minority oversampling technique. 12 2018.

[12] Gautier C. Lobry, J. R. Hydrophobicity, expressivity and aromaticity are the major trends of amino-acid usage in 999 escherichia coli chromosome-encoded genes. *Nucleic acids research*, 22(15), 1994.

[13]W. Yang, B. Lu, and Y. Yang. A comparative study on feature extraction from protein sequences for subcellular localization prediction. In *2006 IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology*, pp. 1–8, Sep. 2006.

Different Plots

Fig. 1 Tree number optimization - Precision

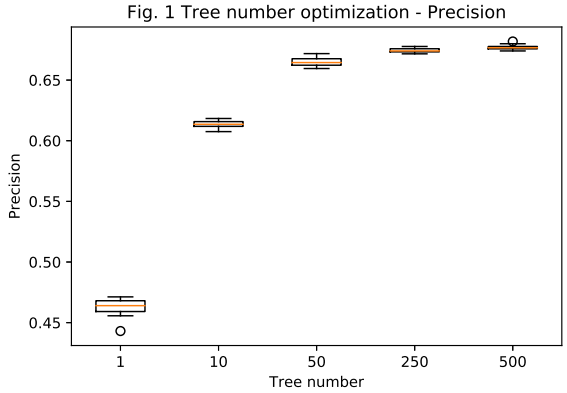

Fig. 2 Tree number optimization - Sensitivity

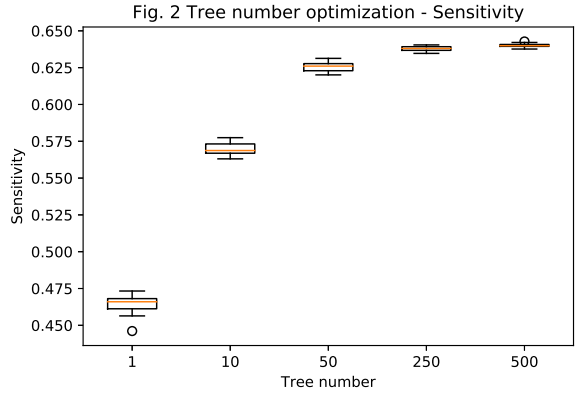

Fig. 3 Tree number optimization - F-measure

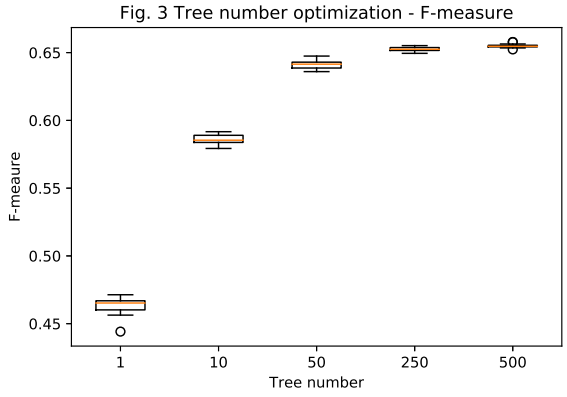

Fig. 4 Tree number optimization - Accuracy

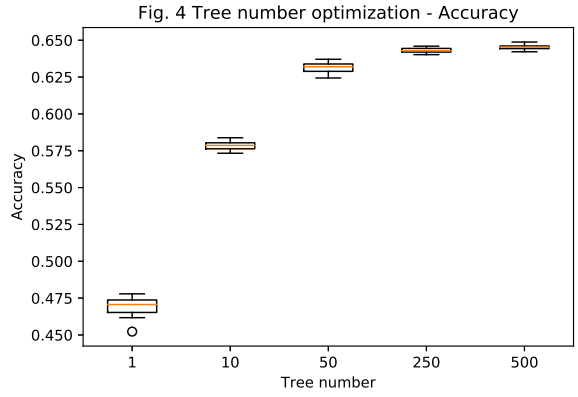

Fig. 5 Tree number optimization - MCC

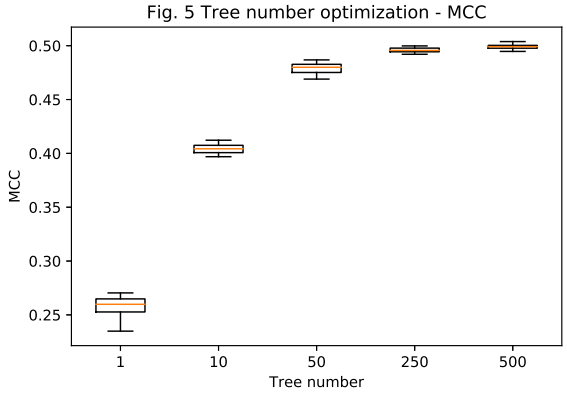

Fig. 6 Multi-class ROC curves - 50 Trees

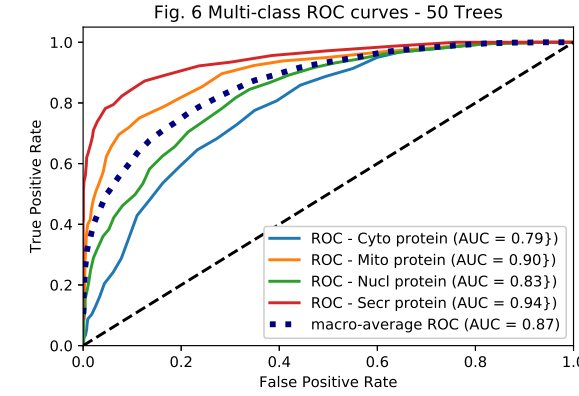

Fig. 7 Multi-class ROC curves - 250 Trees

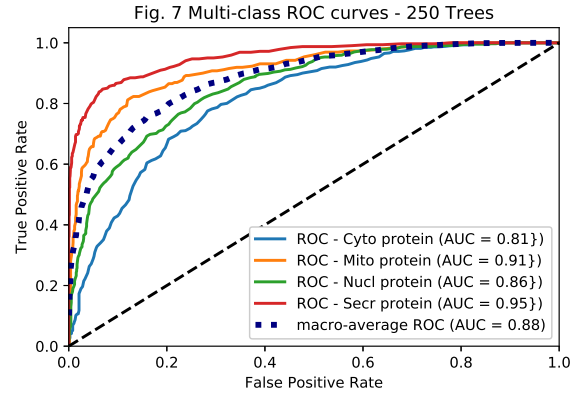

Fig. 8 Multi-class ROC curves - 500 Trees