NcPred for accurate nuclear protein prediction using *n-mer* statistics with various classification algorithms

Md Saiful Islam, Alaol Kabir, Kazi Sakib, and Md. Alamgir Hossain

Abstract Prediction of nuclear proteins is one of the major challenges in genome annotation. A method, NcPred is described, for predicting nuclear proteins with higher accuracy exploiting *n-mer* statistics with different classification algorithms namely Alternating Decision (AD) Tree, Best First (BF) Tree, Random Tree and Adaptive (Ada) Boost. On BaCello dataset [1], NcPred improves about 20% accuracy with Random Tree and about 10% sensitivity with Ada Boost for Animal proteins compared to existing techniques. It also increases the accuracy of Fungal protein prediction by 20% and recall by 4% with AD Tree. In case of Human protein, the accuracy is improved by about 25% and sensitivity about 10% with BF Tree. Performance analysis of NcPred clearly demonstrates its suitability over the contemporary in-silico nuclear protein classification research.

1 Introduction

Nucleus, popularly known as the control center of a cell, is the central unit of eukaryotic cells [2]. Unlike other organelles, its function is regulated by two genomes due to the presence of an explicit nuclear genome. It performs a plethora of biochemical reactions like oxidative phosphorylation, Krebs cycle, DNA replication, transcription, translation, etc. In addition nuclei are also involved in apoptosis and ionic homeostasis [3]. Because of their multidimensional utility, nuclear proteins are associated with several diseases, including Xeroderma pigmentosum, Fanconis anaemia, Bloom syndrome, Ataxia telangiectasia and Retinoblastoma [4] etc.

Md. S. Islam · A. Kabir
Institute of Information Technology, University of Dhaka, Bangladesh.
e-mail: mdsaifulislam@groupwise.swin.edu.au, alaol_kabir@yahoo.com

K. Sakib · Md. A. Hossain
Department of Computing, Bradford University, West Yorkshire, UK.
e-mail: $\{k.muheymin-us-sakib, M.A.Hossain1\}$@bradford.ac.uk

A majority of nuclear proteins are synthesized in cytoplasm from where those are transported inside nucleus. But a small number of nucleus-resident proteins are also synthesized inside nucleus. Proteins that are imported to nucleus contain a leader sequence at the N-terminus containing information needed to localize [5]. But this is not true always, as in many cases the leader sequence is altogether absent.

In the past, a number of methods were developed to predict proteins, indeed not exclusively for nuclear proteins [18]. The similarity search-based techniques fall under the first category in which the query sequence is searched against experimentally annotated proteins. Although the similarity-based method is very informative and considered to be the best, it becomes severely handicapped when no apparent homology is found [6].

Some of the methods are based on predicting signal sequences where sorting signals, present on the protein, are used. This category includes TargetP [7], SignalP [8]. Although these methods are quite popular, not all proteins have signals; for example, only around 25% of yeast nuclear proteins have matrixtargeting signals particularly at the N-terminus [9].

Methods also attempt to predict subcellular localization on the basis of sequence composition such as ESLpred (Subcellular Localization of Eukaryotic Proteins Prediction) [10], HSLpred [11], NNPSL [6], and LOCSVMPSI [12]. Although their overall performance is good, prediction accuracy of nuclear proteins is much lower than for proteins in other locations. It shows that nuclear protein localization is much more complex and hence warrants special attention.

This paper proposes a new technique called NcPred to improve the prediction accuracy of nuclear proteins with four different powerful machine learning algorithms namely AD Tree, BF Tree, Random Tree and Ada Boost. Rather than signals and subcellular localizations, NcPred exploits *n-mer* statistics presents in the sequence databases. Experimental evaluation shows the suitability of NcPred over the contemporary nuclear protein classification research.

2 Proposed Nuclear Protein Prediction (NcPred) Method

2.1 Modeling the Problem

The classification of nuclear proteins is a binary classification problem and the model developed here is a supervised learner. Formally, a set of protein sequences $S = \{s_1, s_2, ..., s_N\}$ and their labels $Y = \{y_1, y_2, ..., y_N\}$ are given ($y_i \in \{Nuclear, Non-nuclear\}$). We wish to determine the label of a newly arrived sequence, s_{new}.

$$S_{new} \overset{M}{\to} Y_{new} \tag{1}$$

Any model M performing this classification should be supervised since the labels of the given sequences are known. That is, each sequence in the database appears as a pair (s_i, y_i). To learn the model, the study exploits *n-mer* distribution statistics that present in the sequence databases rather than signals [7, 8] and subcellular localization [6, 10, 11, 12].

2.2 Selection of Features

One of the most important tasks in the classification is to select the appropriate features that can improve the model accuracy. In NcPred *n-mer* combinations are used to construct the feature vector. The overlapping concept has been brought in *n-mer* combinations to make it more accurate and to reduce the search space, *n-mers* are extracted directly from the existing sequences rather than permuting all amino acids. As shown in the experimental evaluation (Section 3), the cogency of *3* and *4-mer* techniques leads to better results because the frequency distribution of the feature set of lower or higher *mers* are not descriptive enough for the machine learning algorithms like AD Tree, BF Tree, Random Tree and Ada Boost.

To construct the desired feature vector, each *n-mer* is searched in both nuclear and non-nuclear protein databases to find its presence in every sequence. The frequency difference is calculated by subtracting non-nuclear protein *n-mers* from nuclear protein *n-mers* total frequency. On the basis of the frequency differences, top 64 *n-mer* combinations are considered to calculate Term Frequency (TF, tf_i), Inverse Document Frequency (IDF, idf_i) and TF-IDF ($(tf-idf)_{i,j}$) values. Since the selection of these *n-mer* combinations have been derived by their frequency distribution, there will be a little chance for a protein sequence not to have any of the top 64 *n-mers* considered to predict. For each of TF, IDF and TF-IDF, the Attribute Relationship File Format (ARFF) [13] is constructed to build the feature vectors. These terms are defined as follows:

$$tf_i = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{2}$$

where $n_{i,j}$ is the number of occurrences of the *n-mer* (t_i) in the sequence d_j, and the denominator is the sum of number of occurrences of all terms in the sequence d_j.

$$idf_i = log\frac{|D|}{|\{d : t_i \in d\}|} \tag{3}$$

with $|D|$ is the total number of sequences in the database and $|\{d : t_i \in d\}|$ is the number of sequences where *n-mer* t_i appears (that is, $n_{i,j} \neq 0$).

$$(tf-idf)_{i,j} = tf_{i,j} X idf_i \tag{4}$$

A high weight in (tf - idf) is reached by a high term frequency (in the given sequence) and a low sequence frequency of the term in the whole collection of sequences; the weights hence tend to filter out common terms.

2.3 Evaluation Metrices

For estimating the predictive accuracy on a given data set a strong statistical process, n-fold cross validation is used (for experiments, 10-fold cross validation available in WEKA is used). In this technique, the data sets are initially partitioned into n subsets. n-1 subsets are used for training and the rest is used for testing the model. The process is repeated n times and average rating is taken to evaluate the model. The standard parameters, namely Accuracy, Precision, Recall or Sensitivity and Specificity [14], that are routinely used in other prediction methods are adopted.

Assume that TP is the total number of truly positive samples, TN is the total number of truly negative samples, FP is the total number of samples that are identified by the classifier as positives but actually those are not and FN is the total number of samples that are identified as negatives but actually not. Then the above mentioned parameters can be calculated as follows.

Accuracy of a classifier is calculated by dividing the number of correctly classified samples by the total number of test samples and is defined as [14]:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} X100 \tag{5}$$

Precision measures the systems ability to present only relevant samples while recall measures systems ability to present all relevant samples. Precision also indicates the probability of correct prediction.

$$Precision = \frac{TP}{TP+FP} X100 \tag{6}$$

$$RecallorSensitivity = \frac{TP}{TP+FN} X100 \tag{7}$$

Specificity is calculated by dividing the number of true negative samples by the total number of samples that should be classified as negatives and is defined as [14] :

$$Specificity = \frac{TN}{TN+FP} X100 \tag{8}$$

We also calculated the Matthews Correlation Coefficient (MCC), the statistical parameter to assess the quality of prediction [15]. MCC = 1 is regarded as perfect, 0 for completely random and -1 as the worst possible prediction.

$$MCC = \frac{(TPXTN)-(FPXFN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{9}$$

3 Experimental Evaluation

To evaluate the performance of NcPred, two experiments are conducted on three different datasets (Blind I, II and III). We experimented with almost all machine learning algorithms available with WEKA but with AD, BF, Random trees and AdaBoost, encouraging results were noticed. Particularly, the performance of SVM was not as good as the reported algorithms.

The Human protein dataset is taken from Blind I which has 363 nuclear Animal proteins, earlier used in BaCello for benchmarking of different eukaryotic subcellular localization methods [1], Blind II has 122 nuclear and 57 nonnuclear Fungal proteins also used in BaCello [1], Blind III consists of 687 nuclear and 1526 nonnuclear Human proteins used in NucPred [10]. Weka 3.6.0 suite of machine learning software [13], written in Java, developed at the University of Waikato, is used to test the algorithms.

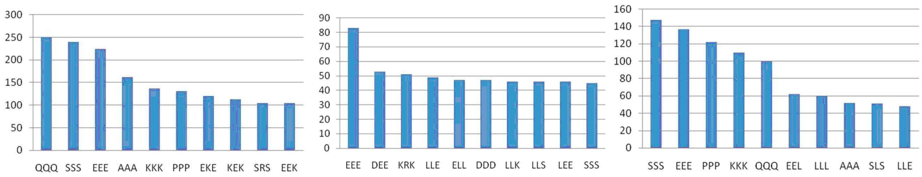

Fig. 1 Top 10 3-mers in Animal, Fungal and Human proteins

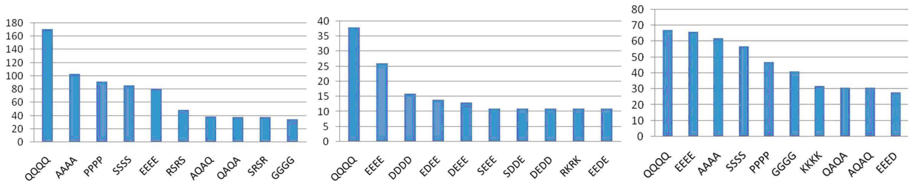

Fig. 2 Top 10 4-mers in Animal, Fungal and Human proteins

Experiment 1: 64 discriminating *3-mer* features of the three given species are obtained (Figure 1 represents the top 10 discriminating *3-mers* in different species). Then TF, IDF and TF-IDF are calculated, trained and tested. Table 1 shows the outcome of Animal, Fungal and Human dataset on TF, IDF and TF-IDF where 93.3% accuracy with AD Tree, 97.9% precision with BF Tree, 100% recall/sensitivity and specificity with AD Tree have been achieved.

Experiment 2: Again 64 discriminating *4-mer* features of the three given species are obtained (Figure 2 represents the top 10 discriminating *4-mers* in different species). TF, IDF and TF-IDF are calculated, trained and tested. Table 2 shows the accuracy, precision, recall/sensitivity and specificity of the Animal, Fungal and Human dataset on TF, IDF and TF-IDF where maximum 93.8% accuracy and 93.0% precision with Random Tree, 97.8% recall/sensitivity and 97.1% specificity with ADA Boost have been achieved.

Table 1 Highest parametric values achieved by AD Tree (ADT), BF Tree (BFT), Random Tree (RT) or ADA Boost (ADAB) in case of TF, IDF, TF-IDF of *3-mer* combinations

	Blind I(%)			Blind II(%)			Blind III(%)		
	TF	IDF	TF-IDF	TF	IDF	TF-IDF	TF	IDF	TF-IDF
Acc	93.3 (ADAB)	90.5 (BFT)	81.1 (ADT)	93.3 (ADT)	86.7 (RT)	86.7 (RT)	88.8 (ADT)	85.2 (BFT)	87.7 (BFT)
Pre	96.3 (ADAB)	87.5 (BFT)	78.0 (ADT)	88.1 (ADT)	84.6 (RT)	88.6 (RT)	97.3 (ADT)	93.2 (BFT)	97.9 (BFT)
Sen	91.9 (RT)	94.6 (ADT)	89.2 (ADAB)	100 (ADT)	91.9 (ADAB)	94.6 (ADAB)	89.4 (RT)	79.3 (RT)	83.8 (RT)
Spe	93.6 (ADAB)	94.1 (BFT)	86.2 (ADAB)	100 (ADT)	88.9 (RT)	92.9 (ADAB)	88.8 (RT)	79.7 (BFT)	83.1 (RT)

The high-percentage of accuracy, precision, recall/sensitivity and specificity clearly indicates that features obtained from the frequency distribution of *n-mers* in the database sequences are capable of discriminating nuclear proteins from non-nuclear protein with higher accuracy.

In a similar classification task, Hutchinson used differential hexamer technique for identifying vertebrate promoter on 29 test sequences where he correctly distinguished 18 proteins as true positive whereas 11 were false positive, which gave him a sensitivity of 62.1% [16]. The result shows an improvement by about 9% when considering the sequences of length above 10,000 [16]. On the other hand, for identifying cis-regulatory motifs in Drosophila, Chan and Kibler used 6-mer distribution technique and achieved a sensitivity and specificity of 38.68% and 93.77% respectively [17]. Interestingly, the sensitivity and specificity outcome is also significantly enhanced by the proposed method.

Existing ESLpred [10] and LOCSVMpsi [12] methods have focused on subcellular localization. These methods have been developed for the prediction of nuclear, cytoplasmic, mitochondrial and extracellular proteins. Prediction of nuclear proteins using these methods achieved 35.8% and 38.7% of accuracy on Blind I dataset respectively as shown in [18]. But the proposed NcPred achieves prediction accuracy of 93.8% for nuclear proteins on Blind I data set.

NpPred achieved the closest efficiency to NcPred. it showed 74.3% and 72.7% accuracy on the prediction of nuclear proteins on Blind I and II dataset. On both occasions, NcPred showed accuracy of 93.8% and 93.3% respectively. The method also achieves MCC of .79 which justifies its applicability. A summary of different nuclear protein prediction methods including NcPred has been given in Table 3.

Table 2 Highest parametric values achieved by AD Tree (ADT), BF Tree (BFT), Random Tree (RT) or ADA Boost (ADAB) in case of TF, IDF, TF-IDF of *4-mer* combinations

	Blind I(%)			Blind II(%)			Blind III(%)		
	TF	IDF	TF-IDF	TF	IDF	TF-IDF	TF	IDF	TF-IDF
Acc	83.5 (ADT)	93.8 (RT)	88.9 (ADT)	91.0 (BFT)	86.5 (ADT)	84.9 (ADT)	89.4 (BFT)	85.4 (BFT)	90.9 (BFT)
Pre	83.8 (RT)	93.0 (RT)	88.9 (RT)	88.5 (BFT)	84.6 (RT)	82.0 (ADT)	87.4 (BFT)	81.6 (RT)	89.3 (BFT)
Sen	91.7 (ADAB)	97.8 (ADAB)	95.0 (ADAB)	94.3 (BFT)	95.1 (ADT)	89.3 (ADT)	92.7 (BFT)	93.3 (BFT)	93.3 (BFT)
Spe	89.2 (ADAB)	97.1 (ADAB)	93.6 (ADAB)	93.9 (BFT)	94.1 (ADT)	88.3 (ADT)	91.9 (BFT)	91.7 (BFT)	92.7 (BFT)

Table 3 Summary of different nuclear protein prediction methods [18] including NcPred.

	Blind I dataset (Animal Proteins)		Blind II dataset (Fungal Proteins)		Blind III dataset (Human Proteins)	
	Sensitivity	Accuracy	Sensitivity	Accuracy	Sensitivity	Accuracy
BacelLo	66.1%	56.1%	66.4%	71.3%	61.0%	67.0%
Loctree	62.2%	49.5%	66.4%	66.9%	63.0%	59.0%
Psort II	70.2%	43.0%	71.1%	44.2%	70.0%	47.0%
SubLoc	67.8%	37.2%	70.5%	38.4%	-	-
ESLpred	79.1%	35.8%	84.4%	37.5%	-	-
LOCSVMpsi	80.2%	38.7%	88.5%	51.0%	-	-
pTARGET	73.3%	64.2%	62.3%	63.5%	-	-
NpPred	87.3%	74.3%	93.4%	72.7%	83.0%	63.0%
NcPred	97.8%	93.8%	97.3%	93.3%	93.3%	90.9%

4 Conclusion

In this study, NcPred has been developed as a tool for classifying the nuclear proteins from the non-nuclear one and verified its suitability in three different data sets consisting of Animal, Fungal and Human proteins. Unlike other methods, NcPred depends on the *n-mer* distribution in the relevant sequences rather than similarity search and subcellular localization. This enables to gain the advantage of higher accuracy and sensitivity achieved by NcPred. The improved accuracy of nuclear protein prediction rate in Animal, Fungal and Human proteins using the proposed approach has validated the use of *n-mers* frequency distribution technique to discriminate between nuclear and non-nuclear proteins. As supported by the extensive experimental results, the proposed approach would be an enormously useful and a proficient tool to meet the demands of the molecular biologists.

The parameters for these algorithms were not optimized, instead default settings were used for experimentation. Currently we are bending to optimize the different parameters for these reported algorithms and a hybrid approach is our future research direction.

References

1. Pierleoni, A., Martelli, P., Fariselli, P., Casadio, R.: Bacello a balanced subcellular localization predictor. Bioinformatics. **22(14)**, 408–416 (2006).
2. Kumar, M., Verma, R., Raghvan, S.: Prediction of mitochondrial proteins using support vector machine and hidden markov model. Int. J. of Biol. Chem. **28(19)**, 5357–5363 (2006).
3. Jassem, W., Fuggle, S., Rela, M., Koo, D., Heaton, N.: The role of mitochondria in ischemia/reperfusion injury. Transplantation, **73(4)**, 493–499 (2002).
4. Ganesh, A., Kenue, R., Mitra, S.: Retinoblastoma and the 13q deletion syndrome. J. of Ped. Ophth. & Strab., **38(4)**, 247–250 (2001).
5. Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., Walter, P.: Molecular Biology of Cell. Fourth Edition. New York, USA: Garland Science (2000).
6. Reinhardt, A., Hubbard, T.: Using neural networks for prediction of the subcellular location of proteins. Nuc. Acids Res. **26(9)**, 2230–2236 (1998).
7. Emanuelson, O., Nielsen, H., Brunak, S., Heijne, G.: Predicting subcellular localization of proteins based on their n-terminal amino acid sequence. J. of Mole. Bio. **330(4)**, 1005–1016 (2000).
8. Bannai, H., Tamada, Y., Maruyama, O., Nakai, K., Miyano, S.: Extensive feature detection of n-terminal protein sorting signals. Bioinformatics, **18(2)**, 335–338 (2002).
9. Marcotte, E., Xenarios, I., Bliek, A., Eisenberg, D.: Localizing proteins in the cell from their phylogenetic profiles. Proc. of Nat. Aca. of Sci., **97(12)**, 115–120 (2000).
10. Bhasin M., Raghava, G.: ESLpred: SVM based method for subcellular localization of eukaryotic proteins using dipeptide composition and PSI-BLAST. Nuc. Acids Res., 414–419 (2004).
11. Garg, A., Bhasin, M., Raghva, G.: Support vector machine based method for subcellular localization of human proteins using amino acid compositions, their order and similarity search. J. of Bio. Chem., **280(14)**, 427–433 (2005).
12. Xie, D., Li, A., Wang, M., Fan, Z., Feng, H.: LOCSVMPSI: a web server for subcellular localization of eukaryotic proteins using SVM and profile of PSI-BLAST. Nuc. Acids Res., **33**, w105–w110 (2005).
13. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: The WEKA data mining software: an update. ACM SIGKDD Explorations News. **11(1)**, 10–18 (2009).
14. Makhoul, J., Kubala, F., Schwartz, R., Weischedel, R.: Performance measures for information extraction. In Proc. of DARPA Broadcast News Workshop, pp. 249–252 (1999).
15. Mathews, B.: Comparison of the predicted and observed secondary structure of t4 phase lysozyme. Bio. et bioph. acta. **405(2)**, 442–451 (1975).
16. Hutchinson, G.: The prediction of vertebrate promoter regions using differential hexamer frequency analysis. Bioinformatics, **12(5)**, 391–398 (1996).
17. Chan B., Kibler, D.: Using hexamers to predict cis-regulatory motifs in drosophila. BMC Bioinformatics, **6**, 262 (2005).
18. Kumar M., Raghava, G.: Prediction of nuclear proteins using svm and HMM models. BMC Bioinformatics. **10(22)** (2009).

5 Reply to Comments

Firstly, the authors would like to thank the reviewers for their prolific comments to enrich the technicality of this paper. We have clearly addressed all the issues that the reviewers made and we also amended our paper accordingly. In the following we included our reply to our reviewers comments.

Reviewer 1: Comments: The paper is suitable for the PACBB conference but I have two mayor concerns about it: - What configuration was used for each tested algorithm?. Have the authors optimized the parameters of the classifiers? - Weka comes with a lot of classification techniques but the authors only tested AD, BF, Random trees and the well-know AdaBoost. What about other classifiers like SVM?
Our Reply:

- We experimented with almost all ML algorithms available with WEKA but got best results only with AD, BF, Random trees and the well-known AdaBoost and we also mentioned it at Section: 3, Page: 5).
- Particularly, the performance of SVM is not as good as the reported algorithms (and we also mentioned it at Section: 3, Page: 5).
- It should be noted that the reported algorithms also vary in themselves for different performance metrics.
- We did not optimize the parameters for these algorithms. Default settings are experimented in our study. Currently we are bending ourselves to optimize the different parameters for these reported algorithms and a hybrid approach is our future research direction. So we clearly added it in our Conclusion section (please see Section 4, Page: 8)

Reviewer 2: Comments: The paper is interesting and well written, but it lacks of a clear explanation about the testing protocol used for the experimentation carried out. Have the authors used a k-fold cross validation. Specifying this point is compulsory in order to evaluate the suitability of the proposed technique.

Our Reply:

- To evaluate the performance of the reported algorithms we adopted 10-fold cross fold validation technique available in WEKA.
- We have included this in the Methodology Section, please see Section 2, Page: 4.