Amino Acids (2014) 46:1343–1351
DOI 10.1007/s00726-014-1697-z

ORIGINAL ARTICLE

Selection of relevant features from amino acids enables development of robust classifiers

Rishi Das Roy · Debasis Dash

Received: 3 July 2013 / Accepted: 14 February 2014 / Published online: 7 March 2014

Abstract Machine learning (ML) has been extensively applied to develop models and to understand high-throughput data of biological processes. However, new ML models, trained with novel experimental results, are required to build regularly for more precise predictions. ML methods can build models from numeric data, whereas biological data are generally textual (DNA, protein sequences) or images and needs feature calculation algorithms to generate quantitative features. Programming skills along with domain knowledge are required to develop these algorithms. Therefore, the process of knowledge discovery through ML is decelerated due to lack of generic tools to construct features and to build models directly from the data. Hence, we developed a schema that calculates about 5,000 features, selects relevant features and develops protein classifiers from the training data. To demonstrate the general applicability and robustness of our method, fungal adhesins and nuclear receptor proteins were used for building classifiers which outperformed existing classifiers when tested on independent data. Next, we built a classifier for mitochondrial proteins of *Plasmodium falciparum* which causes human malaria because the latest corresponding classifiers are not publically accessible. Our classifier attained 98.18 % accuracy and 0.95 Matthews correlation coefficient by fivefold cross-validation and outperformed existing classifiers on independent test set. We implemented this schema as user-friendly and open source application Pro-Gyan (http://code.google.com/p/pro-gyan/), to build and share executable classifiers without programming knowledge.

Keywords Protein sequence analysis · Feature extraction · Protein classifier design and evaluation · Mitochondrial protein · Machine learning

Abbreviations

ML	Machine learning
MP	Mitochondrial proteins
MCC	Matthews correlation coefficient
ANN	Artificial Neural Network
SVM	Support vector machine
FCA	Feature calculation algorithms
FSM	Feature selection methods
FCBF	Fast correlation-based feature
PF	*Plasmodium falciparum*
nrPfM165	Non-redundant training set
nrPfM205	Non-redundant test set
API	Application programming interface
AAC	Amino acid composition
AAPC	Amino acid pair composition
AC	Auto-correlation
CDTd	Composition, transition and distribution descriptors
PseAAC_T1	Pseudo amino acid composition Type1
PseAAC_T2	Pseudo amino acid composition Type2
SOD	Sequence-order-coupling descriptors
CD	Charge distribution

Work performed at: CSIR-Institute of Genomics and Integrative Biology.

Electronic supplementary material The online version of this article (doi:10.1007/s00726-014-1697-z) contains supplementary material, which is available to authorized users.

R. Das Roy · D. Dash (✉)
GN Ramachandran Knowledge Centre for Genome Informatics,
CSIR-Institute of Genomics and Integrative Biology, Mall Road,
Delhi 110007, India
e-mail: ddash@igib.res.in

R. Das Roy
e-mail: rishi.dasroy@gmail.com

IDP	Intrinsically disordered proteins
FWF	FoldIndex window-based features
MAE	Mean absolute error
TF	Top features
JRE	Java runtime environment
GUI	Graphics user interface
PSSM	Position-specific scoring matrix
FK-NN	Fuzzy K nearest neighbor
NR	Nuclear receptor
ROC	Receiver operating characteristic
Pgc	Pro-Gyan classifier

Introduction

High-throughput technologies generate huge amount of biological data that are noisy and heterogeneous. Machine learning (ML) has been successfully applied to infer hidden patterns from such complex data and build classifiers to characterize the unknowns (Arvey et al. 2012; BÃ¡nfai et al. 2012). This in turn saves resource-intensive and time-consuming experiments (Muggleton 2006). ML techniques such as artificial neural network (ANN), support vector machine (SVM), etc. which majorly accept data in numeric form cannot analyze the biological sequence data directly. Hence, the non-numeric data are converted into quantitative features by feature calculation algorithms (FCA). For a successful ML application, these features should be relevant to the biological phenomenon and therefore requires knowledge of domain experts and computational biologists. Once the sequence data are converted into features, ML methods are used to select the relevant features and to develop the classifiers. Further, to make the classifiers publically accessible to the users, web server or standalone applications are developed to easily annotate different biological entities such as genes, RNAs or proteins.

Proteins are the sequence of amino acids and the functional workhorse of cell. A large number of proteins across different species have been sequenced with the aid of high-throughput sequencing technologies and determining their functions is essential to envisage cellular functions. Since experimental methods to characterize these myriad of proteins is tedious and time-consuming, plenty of ML applications have been successfully applied to predict protein folds (Shamim et al. 2007), enzyme families (Cai et al. 2004), RNA-binding proteins (Han et al. 2004), nuclear receptor subfamilies (Wang et al. 2011), etc. However, regular update of existing or development of new classifiers is required to incorporate the new experimental results for more precise classifications. The existing applications cannot build new classifiers as they have been developed to annotate proteins only. However, there are few applications (Li et al. 2006; Shen and Chou 2008; Cao et al. 2013) which only provide the FCA used in these classifiers to generate a large number of features, but the methods to select the relevant features to develop robust classifiers and to build final ML applications directly from the training data set is not available as a single application. Overall, generating classifier from biological data is still decelerated by the human assisted processes. Hence, an automated application could be useful to perform this repetitive task of classifier development from experimentally data.

Here, we present an integrative general schema to perform basic ML steps to speed up the development of classifiers from training data for different kind of protein classification problems. The schema was designed to generate a common and large set of about 5,000 features for all classification problems. As irrelevant and immense number of features can diminish the performance of the classifiers due to the lesser number of sample size than features, we implemented hybrid of filter and wrapper methods for relevant feature selection (Saeys et al. 2007). We compared two different feature selection methods (FSMs): *F*-score (Chen and Lin 2006) and Fast correlation-based feature (FCBF) (Yu and Liu 2003) to evaluate and select the relevant features from the unbiased set of features (Fig. 1). The final outputs were executable and easily sharable SVM classifiers for different classification problems developed with this generic procedure.

To substantiate its general applicability, we first built classifiers for adhesions and nuclear receptors and compared with existing classifiers. The number of selected features was substantially lower than the features used by the existing classifier (Wang et al. 2011; Ramana and Gupta 2010). In each case, both the FSMs were successful to select the relevant features and achieved better results on independent test set.

Further, we chose to classify mitochondrial peptides (MPs) from *Plasmodium falciparum* (PF) which is the cause of malaria in human and claims millions of lives in developing countries (Murray et al. 2012). Mitochondria are the powerhouse of eukaryotes and so the MPs could be potential drug targets. There are several generic predictors such as TargetP (Emanuelsson et al. 2000), Wolf PSORT (Horton et al. 2007), MITOPRED (Guda et al. 2004), MitPred (Kumar et al. 2006), etc. for sub-cellular protein location prediction. Bender et al. showed organism-specific classifier PlasMit performs better than generic tools and achieved accuracy of 90 % (Bender et al. 2003). Consecutively several other classifiers (Chen et al. 2012; Verma et al. 2010; Jia et al. 2011) were reported with better accuracy but remained inaccessible (Chen et al. 2012; Jia et al. 2011) to use in recent studies (Oehring et al. 2012; Atkinson et al. 2012). So we built a new classifier for MPs which achieved 98.18 % accuracy with 0.95 Mathews correlation coefficient (MCC) by fivefold cross-validation.

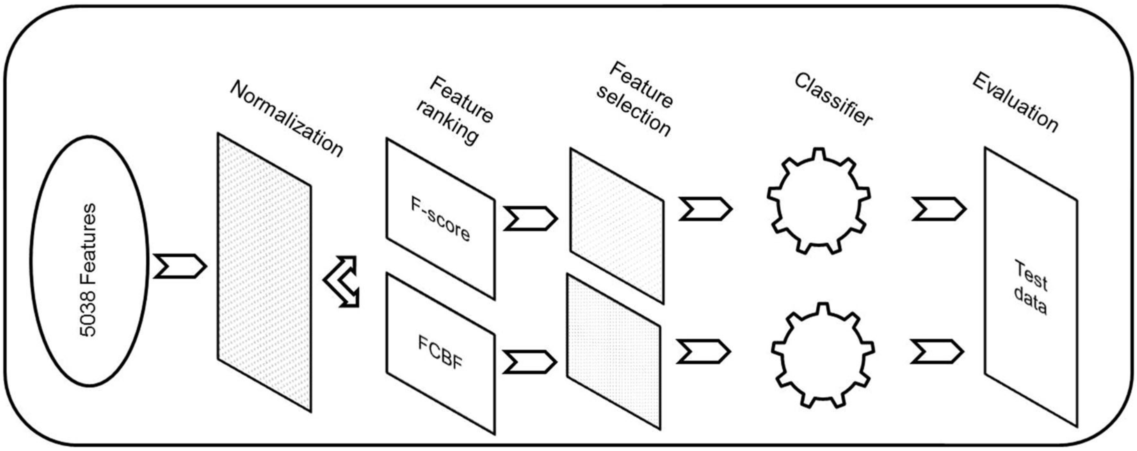

Fig. 1 Integrated schema from feature generation, normalization, selection, development of easily executable and accessible classifiers. Two feature-ranking methods (*F*-score and FCBF) were implemented for feature selection and build classifiers for each problem

In this study, we examined an integrated schema of algorithms to develop classifiers for different characteristics of proteins and showed its applicability. Two different FSMs were benchmarked against two different data sets. We also developed a new executable classifier to fulfill the absence of an updated classifier for MPs of *P. falciparum*. Overall, we developed three new classifiers which are better than the existing ones on independent test data. The above results motivated us to implement the schema as user-friendly software, Pro-Gyan (http://code.google.com/p/pro-gyan/). It accepts labeled protein sequences as training data and generates a large feature set. The relevant features are selected by FSM to build a classifier which can be shared and reused. The results and speed of Pro-Gyan will encourage experimental biologist to apply ML on their own proteomic data. Furthermore, it is an open source application and is integrated to another popular open source ML library, Weka. Hence, bioinformaticians and ML experts could easily examine and contribute to Pro-Gyan by introducing new FCA and ML algorithms. To the best of our knowledge, this is a unique application which can build classifiers directly from training data for proteins across different classification problems and makes it easily accessible for future predictions. Pro-Gyan is not just a classifier like other existing tools; rather, it is able to build new classifiers without programming assistance.

Methods

Feature calculation algorithms

PROFEAT (Li et al. 2006) provides some conventional FCA which can be found utile for prediction of protein folds (Li et al. 2008; Shamim et al. 2007), enzyme families (Cai et al. 2004), RNA-binding proteins (Han et al. 2004), and many more, but it cannot process a request with more than 1,000 sequences and does not provide an application programming interface (API). So we developed our own library, which has FCA from PROFEAT: (1) amino acid composition (AAC), (2) auto-correlation (AC), (3) composition, transition and distribution descriptors (CTDd), (4) sequence-order descriptors (SOD) and (5) pseudo AAC Type1 (PseAAC_T1). The library was further extended by including (6) PseAAC Type2 (Chou 2005; Chou and Cai 2005) (PseAAC_T2), (7) amino acid pair composition (Shamim et al. 2007) (AAPC; with varying gap from 1 to 9 between amino acids), (8) global properties (Gasteiger et al. 2005), (9) charge distribution (CD) (Bum Ju and Keun Ho 2008). A brief explanation of these feature calculation algorithms is given in the supplementary text.

Together these contributed 4,857 features, which were calculated to extract the sequence pattern of different physicochemical properties such as hydrophobicity, average flexibility, residue accessible surface area in tripeptide, residue volume, normalized van der Waals volume, polarity, polarizability, charge, secondary structure, solvent accessibility, etc. (Supplementary Table 1).

FoldIndex window-based features

The importance of intrinsically disordered proteins (IDP) is described in different biological processes (Dunker et al. 2008; Singh and Dash 2008). FoldIndex (Prilusky et al. 2005) window-based features (FWF) are new set of features, which were implemented here to quantify the characteristics of IDPs. FoldIndex is a web server which predicts whether a protein will be intrinsically disordered or not, based on a score calculated from its sequence (Uversky et al. 2000). A positive or negative value of the score indicates that the protein is folded or unfolded, respectively. FoldIndex can also calculate window-based scores, which reveal the domains that are likely to be folded or unfolded. To find any hidden pattern from this domain information, 181 features (see supplementary method) including FoldIndex were calculated:

Classifier evaluation

There are different metrics to evaluate the performance of a classifier, such as:

$$\text{Accuracy (Acc)} = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN})$$

$$\text{Sensitivity or Recall (Sn)} = \text{TP}/(\text{TP} + \text{FN})$$

$$\text{Specificity (Sp)} = \text{TN}/(\text{FP} + \text{TN})$$

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP})$$

$$F\,\text{Measure} = 2 * \text{Precision} * \text{Recall}/(\text{Precision} + \text{Recall})$$

$$\text{Matthews correlation coefficient (MCC)} = (\text{TP} * \text{TN} - \text{FP} * \text{FN})/\sqrt{\{(\text{TP} + \text{FP}) * (\text{TN} + \text{FN}) * (\text{TP} + \text{FN}) * (\text{TN} + \text{FP})\}}$$

$$\text{Mean absolute error (MAE)} = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i|$$

where TP = true positive, TN = true negative, FP = false positive, FN = false negative, n = sample size, f_i = prediction and y_i = true value.

Accuracy could be misleading when the data is unbalanced. The sensitivity, specificity, precision and F measure can be influenced by interchanging the label (positive/negative) of the data. In our study, we selected MCC to evaluate the classifiers as it has no such dependencies.

Feature selection

We calculated total of 5,038 features (Supplementary Table 1, 4) for all training sets. All of these features were not equally significant for every classification problem. A classifier trained with relevant features produces more accurate prediction. Different FSMs (Saeys et al. 2007; Bum Ju and Keun Ho 2008) were developed and applied to filter out noisy features. But FSMs should be applied carefully (Smialowski et al. 2010) to minimize the chance of overfitting. Here, in this study, filter and wrapper-based FSMs were applied consecutively. The features were ranked according to their relevance separately by two different methods. Then, a subset of top features was selected to build two separate classifiers to achieve maximum possible MCC in reasonable time and compare the feature ranking methods (Fig. 1). The procedure is given below in details.

1. Feature ranking
   This step involves feature ranking by two matrices.
   (a) F-scores (Chen and Lin 2006): It measures the relevancy by comparing the distributions of a feature between the two classes

$$F(i) \equiv \frac{\left(\bar{f}_i^{(+)} - \bar{f}_i\right)^2 + \left(\bar{f}_i^{(-)} - \bar{f}_i\right)^2}{\frac{1}{n_+ - 1}\sum_{k=1}^{n_+}\left(f_{k,i}^{(+)} - \bar{f}_i^{(+)}\right)^2 + \frac{1}{n_- - 1}\sum_{k=1}^{n_-}\left(f_{k,i}^{(-)} - \bar{f}_i^{(-)}\right)^2}$$

   where, $\bar{f}_i^{(+)}$, $\bar{f}_i^{(-)}$ and $\bar{f}_i$ are the mean of the ith feature of positive, negative and whole data sets. $f_{k,i}^{(+)}$ and $f_{k,i}^{(-)}$ are the ith feature of kth sample from n_+ positive and n_- negative sample. F-scores were measured for all the features. A relevant feature would have higher F-score than the irrelevant features. The disadvantage of F-score is that it cannot measure the relation between two features.

   (b) FCBF: Features relevant to the class variable and non-correlated with other features construct a good subset of independent features. FCBF (Yu and Liu 2003) was used to acquire a subset of non-redundant and ranked features. It measures the relevancy or correlation of two random variable by *Symmetric uncertainty*,

$$\text{SU}\,(f_i, C) = 2\left[\frac{IG(f_i|C)}{H(f_i) + H(C)}\right]$$

   where H(C) is entropy (information theory) of class variable C and IG ($f_i|C$) (Quinlan 1993), information gain, is a measure of uncertainty of C, which decreases in the presence of additional information provided by feature f_i. It scores 0 if two variables are independent of each other and 1 if one of the variables can explain the other completely. Subsequently, redundant features f_j are removed, if it is less relevant than f_i, i.e.,

$$\text{SU}(f_j, C) \leq \text{SU}(f_i, C)$$

   f_j is more correlated with f_i than C, i.e.;

$$\text{SU}(f_j, f_i) \geq \text{SU}(f_j, C)$$

   FCBF does not only calculate the relevancy of features, it also considers the relation between the features and filters out irrelevant ones.

2. Top features selection
   The above methods ranked the features independent of the classifier and did not provide the required number of top features (TFs) to build optimal classifier. Wrapper-based FSMs (Saeys et al. 2007) evaluate all possible subset of features which is a time-consuming process. Instead, we heuristically evaluated 2, 100, 200, 300 and 400 top features to build classifiers. Then, an algorithm was implemented to build classifiers with different number of TFs equally spread over a smaller search space around the best performing classifier compared by MCC (Fig. 2a). Our approach recursively executed until the length of search space reduces to a threshold which is equal or less than 10. The algorithm 1 summarizes the above procedure. To reduce the possibility of over-fitting, the dimension of search space is restricted to 400 TFs.

Algorithm 1: **recursive_search**

```
Input:  F // Ranked  features
        $TF_{min}$ //minimum number of top features to evaluate
        $TF_{max}$ //maximum number of top features to evaluate
Output: $MCC_{max}$ //maximum Mathew correlation coefficient (MCC).
        m //number of top features required to achieve $MCC_{max}$.

begin
$MCC_{max} = 0$;
$m = TF_{min}$;
$step = (TF_{max} - TF_{min})/5$; //distance between two points inside the search space
for $i = (TF_{min}+step)$ to $TF_{max}$, increment  by $step$, begin
        $MCC_{temp} = MCC(TF_i)$ // calculate MCC of SVM classifier build with $i$ top features.
        if($MCC_{temp} > MCC_{max}$)
                $MCC_{max} = MCC_{temp}$;
                $m = i$;
        end
end

//redefining the boundary of the new search space (triangles in Fig 2a)
$TF_{min} = m – step$;
$TF_{max} = m + step$;

If ( ($TF_{max}$ - $TF_{min}$) <= 10)  return [$MCC_{max}$, m]
        [$MCC_{temp}$, $m_{temp}$] = recursive_search(F,$TF_{min}$,$TF_{max}$) // next search

if($MCC_{temp} > MCC_{max}$)
        $MCC_{max} = MCC_{temp}$;
        $m = m_{temp}$;
end
return [$MCC_{max}$, m]
end
```

Machine learning

Features were normalized before training. SVM (Vapnik 1999) was used to build the classifiers by fivefold cross-validation to measure the performance. The classifiers were built by radial basis kernel function, and to find the best possible gamma (within 2^{-15}–2^{3}) and cost (within 2^{-5}–2^{15}), parameter grid search (equivalent to grid.py of LIBSVM) was executed with same splits for all cross-validation (Fig. 2b). For each classification problem, two classifiers, $\mathrm{SVM}^{F\text{-score}}$ and $\mathrm{SVM}^{\mathrm{FCBF}}$, were built corresponding to F-score and FCBF feature-ranking methods. LIBSVM (Chih-Chung and Chih-Jen 2011) and Weka (Hall et al. 2009) were used to build the classifiers.

Pro-Gyan to build and share classifiers

Pro-Gyan was developed to easily build and share protein classifiers. It is an open source desktop application written in Java. It can be executed on any operating system with Java runtime environment (JRE) six or above. The user needs to provide only the labeled training data in FASTA format through the easy-to-use graphic user interface (GUI). Multi-threaded Pro-Gyan can also be used to provide a quick estimate of the newly built classifier's performance. The final classifier should be built with single thread for consistent results. Pro-Gyan is available at http://code.google.com/p/pro-gyan/.

The repository of feature calculation algorithm is developed by utilizing Java reflection through spring

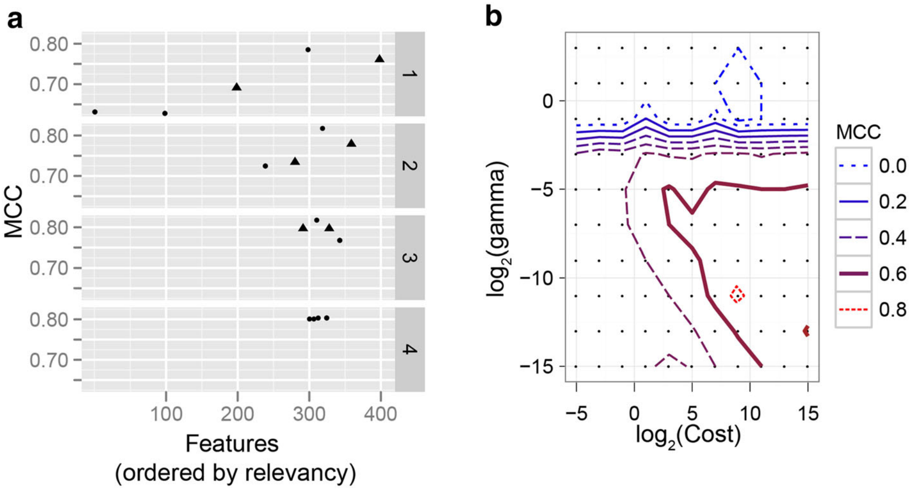

Fig. 2 **a** The features were ranked by their relevancy. Heuristically different numbers of top features were used to build classifiers. The classifiers were compared by their MCC score. A recursive search was then executed in a smaller search space (between the *triangles*) surrounding the best performing classifier. **b** Grid search guided by MCC score was performed to tune the cost and gamma parameter of radial basis function of SVM. A SVM classifier was built for each pair of cost and gamma by fivefold cross-validation (color figure online)

(http://www.springsource.org/) framework where each FCA is a Java object and is easy to maintain them. The SVM implementation provided by LIBSVM (Chih-Chung and Chih-Jen 2011) is used through Weka (Hall et al. 2009) which is an open source platform of ML algorithms. The library or software included in this application are all free and open source software.

Results

The schema presented here was designed as a generic application to solve binary protein classification problems. To substantiate this, we built and evaluated two different classifiers for adhesin and nuclear receptor proteins. The classifiers were trained and evaluated using published training and test set of FaaPred (Ramana and Gupta 2010) and NR-2L (Wang et al.), respectively. Further, we built classifier for MPs of *P. falciparum* and compared with multiple existing classifiers. Comparison with each predictor is discussed below. To build a classifier for MPs of *P. falciparum*, the training data set from PFMpred (Verma et al. 2010) (40 MPs and 135 non-MPs) and test data set (Jia et al. 2011) (108 MPs and 125 non-MPs) from GeneDB were used. In earlier studies, redundant sequences between test and training set were not removed due to smaller sample size (Verma et al. 2010). For a robust evaluation of the classifier, we removed the sequences which had 100 % identity to any other sequences in the training and test set using CD-HIT(Li and Godzik 2006). The final non-redundant training set (nrPfM165) had 40 MPs and 125 non-MPs. The classifiers were evaluated on a non-redundant test set (nrPfM205) of 90 MPs and 115 non-MPs.

Adhesin classifier

Classifiers built for fungal adhesin and adhesin-like proteins were compared with FaaPred (Ramana and Gupta). Ramana et al. developed multiple classifiers using different sets of feature in FaaPred. The best performing classifier (PSSM-a) was built on features based on position-specific scoring matrix (PSSM). PSSM was obtained by using PSI-BLAST against protein database. It is both time-consuming and database-dependent process.

The calculated features of present study were extracted from the protein sequence itself. FAAP$^{F\text{-score}}$ (MCC 0.81) selected 90 and FAAP$^{\mathrm{FCBF}}$ (MCC 0.88) selected 55 features as compared to 400 features of FaaPred (MCC 0.68) thus reducing the possibility of over-fitting (Table 1; Supplementary Table S2 and S3). Another reason of its success could be evaluations of classifiers with respect to MCC to treat biased data set like the ones presented here.

Nuclear receptor classifier

We also compared our method with NR-2L (Wang et al.), a multi-class classifier for nuclear receptors. It has two Fuzzy K nearest neighbor (FK-NN) classifiers, one for segregating nuclear receptor (NR) from non-nuclear receptor and second to further classify NRs in seven subfamilies. As our schema builds only binary classifiers, we built classifiers NRP$^{F\text{-score}}$ and NRP$^{\mathrm{FCBF}}$ to classify nuclear receptors which are comparable to first level of classifier of NR-2L. The performance of our classifiers were better than NR-2L (Table 1) with respect to number of features and MCC. NRP$^{\mathrm{FCBF}}$ used significantly less number of features compared to NR-2L.

Table 1 Comparison of our classifier with existing classifiers on their test data set

Classifier	Features[a]	Sn(%)	Sp(%)	Acc	MCC
Adhesin					
FaaPred	400	100	90.32	91.22	0.68
FAAP^{FCBF}	55	96.88	97.74	97.66	0.88
Nuclear receptor					
NR-2L	881	99.64	96.20	98.03	0.96
NRP^{FCBF}	224	99.30	99.80	99.53	0.99

[a] Selected feature vector length

There were very few common features, selected by *F*-score and FCBF for fungal adhesin proteins with respect to nuclear receptor proteins (Supplementary Fig S2). The detailed training and test information are given in supplementary Table S2 and S3. The list of selected features are given in supplementary Table S5.

A case study: classifier for mitochondrial proteins of PF

Training on N terminal of proteins

Here, in this study, we built classifiers with FCBF selected features from N terminal of training sequences as this region are known to be enriched with positively charged amino acids and distinguishable structural properties in mitochondrial proteins (Emanuelsson et al. 2001; Hammen and Weiner 1998). We used different lengths of N terminal such as 24, 31, 42 (Bender et al. 2003) suggested by Bender et al. as well as 50, 60, 70, 80, 90 and 100 (Table 2). The MCC of the classifiers increased with increase in N terminal length till 50. We found PF_Mito^{FCBF} built with 50 amino acid length N terminal performed the best with an accuracy of 98.29 % (MCC 0.95) and lowest MAE (mean absolute error). The corresponding receiver operating characteristic (ROC) curve is shown in supplementary Fig S3. The classifier was built with 52 selected features and 8 out of top 10 features were associated with positively charged amino acids and structural properties. In earlier classifiers, structural properties of N terminal was not evaluated to develop them. The FoldIndex of N terminal shows MPs are more disordered than non-MPs (one sided Mann–Whitney test: p value <0.1e−07; Supplementary Fig S4).

Comparison with other classifiers

We compared training performance metric of PF_Mito^{FCBF} with different classifiers and found it outperformed others and was as good as the method of Jia et al. with respect to MCC (Table 3). For rigorous evaluation, we performed jackknife cross-validation and our method scored MCC 1. However, Jia et al. did not evaluate their classifier using a test set that was not used in training as suggested (Smialowski et al. 2010). A set of 24 MPs (Verma et al. 2010) was earlier used for independent evaluation of the classifiers (Verma et al. 2010; Bender et al. 2003). However, lack of negative data set could not determine the specificity of these methods. For robust evaluation and comparison with the publicly available classifiers, we used nrPfM205 test set built with 90 MPs and 115 non-MPs of *P. falciparum*. The general methods TargetP (Emanuelsson et al. 2000) and WoLF PSORT (Horton et al. 2007) achieved higher specificity, whereas organ-specific classifiers PlasMit and PFMpred performed better with respect to sensitivity (Table 4). The PF_Mito^{FCBF} classifier scored highest MCC (0.54) and accuracy (77.07 %) on this non-redundant test set by counterbalancing sensitivity and specificity.

Table 2 Classifiers built on different N terminal length of nrPfM165 training set

N terminal Length	Features[a]	MAE	Sn(%)	Sp(%)	Acc(%)	MCC
24	49	0.08	95.00	97.60	96.97	0.92
31	49	0.06	97.50	98.40	98.18	0.95
42	35	0.06	100	97.60	98.18	0.95
50	52	0.05	95.00	99.20	98.18	0.95
60	47	0.09	92.50	96.00	95.15	0.87
70	52	0.1	87.50	98.40	95.76	0.88
80	54	0.07	90.00	99.20	96.97	0.92
90	52	0.07	97.50	96.80	96.97	0.92
100	52	0.09	92.50	96.80	95.76	0.89

[a] Selected feature vector length

Table 3 Comparison of different methods on training set

Method	Sn(%)	Sp(%)	Acc(%)	MCC
PlasMit	94.00	89.00	90	0.74
PFMpred	97.5	91.04	92.00	0.81
Jia et al.	97.50	97.30	98.80	0.97
Chen et al.	100	89.63	92.00	0.82
PF_Mito^{FCBF}	95.00	99.20	98.18	0.95

The integrated schema was converted into software, Pro-Gyan (Supplementary Fig S5). It is a user-friendly, platform-independent stand-alone tool, which can be easily used by biologists (Supplementary User manual). The graphic user interface directly accepts the labeled input (Supplementary Fig S6) from the users and can create classifiers. Pro-Gyan reports the different performance metrics (Supplementary Fig S7), ROC curve (Supplementary Fig S8)

Table 4 Evaluation of specificity and other metrics on independent test data (PfM205)

Method	(MP 90/ non-MP 115)	Sn(%)	Sp(%)	Acc(%)	MCC
TargetP	15/115	17.78	100	63.90	0.33
WoLF PSORT	28/111	31.11	96.52	67.80	0.37
MitPred[b] (SVM)	54/23	60.00	20.00	37.56	−0.22
MitPred[b] (Pfam + SVM)	25/115	27.78	100	68.29	0.42
PlasMit	53/101	58.89	87.83	75.12	0.49
PFMpred[a]	45/92	50.00	80.00	66.83	0.32
PF_Mito[FCBF]	51/107	56.67	93.04	77.07	0.54

[a] PFMpred SAAC classifier was used due to unavailability of SAAC + PSSM classifier

[b] Both methods of MitPred (HMM based) were evaluated

and histogram of the selected features (Supplementary Fig S9). The newly built classifiers can further be evaluated with test data for which Pro-Gyan again generates prediction results along with performance metrics and ROC curve. The classifiers can be shared with other researchers in "pgc" (Pro-Gyan classifier) format which consists of feature normalizing information, selected features and SVM files. An interested user can import this classifier (Supplementary Fig S10) in his own copy of Pro-Gyan and annotate the unknown proteins and get the results in tabular format (Supplementary Fig S11).

Discussion

We calculated a large and diverse set of features covering compositional, physicochemical, structural, sequential patterns from proteins to solve diverse set of classification problems. This large feature set gives an opportunity to evaluate them and select relevant ones to develop robust classifiers. Selection of relevant features was carried out by two different feature-ranking algorithms. We compared our results with published classifiers based on different ML algorithms such as SVM, FK-NN, ANN and increment of diversity (ID). The result shows the integrative approach has general capability to build classifiers from protein sequences by selecting informative features from a large feature pool. Our approach extensively used computational power and large number of features. The wrapper-based feature selection is generally criticized for brute-force, time-consuming and over-fitting on training set. These limitations were reduced by selecting the features according to their relevancy. Further, the classifiers were evaluated on independent test set as suggested (Smialowski et al. 2010). Finally, three classifiers for fungal adhesin (FAAP.pgc), nuclear receptor (NRP.pgc) and MPs of PF (PF_MITO.pgc) were developed and available with Pro-Gyan for future evaluation and application.

Conclusion

We have developed an application, Pro-Gyan, which enables the user to develop and share new protein classifiers without any a priori ML or programming knowledge. Publically available protein databases are continuously being updated through new experimental studies. Existing classifiers should also be updated regularly with this new data for more reliable predictions, and those predictions should be subsequently validated. Furthermore, sequencing technologies have been advancing faster, thereby infusing more unannotated sequences. This necessitates development of newer classifiers which are not dependent on computer experts and programmers. A fast automated application, Pro-Gyan will accelerate the process of knowledge discovery. Pro-Gyan-built classifiers are easily distributable and will remain available for future studies. However, it cannot check redundancy of data. So the training set should be prepared carefully. It is easy to add new feature calculation algorithms to open source tools such as Pro-Gyan by computational biologists or include other functionalities of Weka developed by ML experts. The results show that classifiers built with a generic approach could outperform specialized classifiers and additionally lead to knowledge discovery. Hence, it could be concluded that Pro-Gyan-like application for DNA, RNA sequences, cell-images, and many more non-numeric data types could be equally useful to enhance our knowledge in future.

Acknowledgments RDR acknowledges HP and CSIR Genesis project (BSC-0121) for funding. DD acknowledges CSIR Genesis project (BSC-0121) for funding through IGIB. The authors also like to thank Ritwick Pal for providing the image of the main window of Pro-Gyan.

Conflict of interest The authors declare that they have no conflict of interest.

References

Arvey A, Agius P, Noble WS, Leslie C (2012) Sequence and chromatin determinants of cell-type—specific transcription factor binding. Genome Res 22(9):1723–1734

Atkinson GC, Kuzmenko A, Kamenski P, Vysokikh MY, Lakunina V, Tankov S, Smirnova E, Soosaar A, Tenson T, Hauryliuk V (2012) Evolutionary and genetic analyses of mitochondrial translation initiation factors identify the missing mitochondrial IF3 in *S. cerevisiae*. Nucleic Acids Res 40(13):6122–6134

BÃ¡nfai B, Jia H, Khatun J, Wood E, Risk B, Gundling WE, Kundaje A, Gunawardena HP, Yu Y, Xie L, Krajewski K, Strahl BD, Chen X, Bickel P, Giddings MC, Brown JB, Lipovich L (2012)

Long noncoding RNAs are rarely translated in two human cell lines. Genome Res 22(9):1646–1657

Bender A, van Dooren GG, Ralph SA, McFadden GI, Schneider G (2003) Properties and prediction of mitochondrial transit peptides from *Plasmodium falciparum*. Mol Biochem Parasitol 132(2):59–66

Bum Ju L, Keun Ho R (2008) Feature extraction from protein sequences and classification of enzyme function. In: International conference on biomedical engineering and informatics, 2008. BMEI 2008, 27–30 May 2008, pp 138–142

Cai CZ, Han LY, Ji ZL, Chen YZ (2004) Enzyme family classification by support vector machines. Proteins: Struct, Funct, Bioinf 55(1):66–76

Cao DS, Xu QS, Liang YZ (2013) Propy: a tool to generate various modes of Chou's PseAAC. Bioinformatics 29(7):960–962

Chen YW, Lin CJ (2006) Combining SVMs with various feature selection strategies. In: Guyon I, Nikravesh M, Gunn S, Zadeh L (eds) Feature extraction, vol 207., Studies in fuzziness and soft computingSpringer, Berlin, pp 315–324

Chen YL, Li QZ, Zhang LQ (2012) Using increment of diversity to predict mitochondrial proteins of malaria parasite: integrating pseudo-amino acid composition and structural alphabet. Amino Acids 42(4):1309–1316

Chih-Chung C, Chih-Jen L (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol 2(3):1–27. doi:10.1145/1961189.1961199

Chou KC (2005) Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes. Bioinformatics 21(1):10–19

Chou KC, Cai YD (2005) Prediction of membrane protein types by incorporating amphipathic effects. J Chem Inf Model 45(2):407–413. doi:10.1021/ci049686v10.1021/ci049686v

Dunker AK, Silman I, Uversky VN, Sussman JL (2008) Function and structure of inherently disordered proteins. Curr Opin Struct Biol 18(6):756–764

Emanuelsson O, Nielsen H, S Brunak, von Heijne G (2000) Predicting subcellular localization of proteins based on their N-terminal amino acid sequence. J Mol Biol 300(4):1005–1016

Emanuelsson O, von Heijne G, Schneider G (2001) Analysis and prediction of mitochondrial targeting peptides. Methods Cell Biol 65:175–187

Gasteiger E, Hoogland C, Gattiker A, Duvaud S, Wilkins MR, Appel RD, Bairoch A (2005) Protein identification and analysis tools on the ExPASy server. In: Walker JM (ed) The proteomics protocols handbook. Humana press Inc., New York, pp 571–607

Guda C, Fahy E, Subramaniam S (2004) MITOPRED: a genome-scale method for prediction of nucleus-encoded mitochondrial proteins. Bioinformatics 20(11):1785–1794

Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. ACM SIGKDD Explor Newsl 11(1):10–18

Hammen PK, Weiner H (1998) Mitochondrial leader sequences: structural similarities and sequence differences. J Exp Zool 282(1–2):280–283

Han LY, Cai CZ, Lo SL, Chung MCM, Chen YZ (2004) Prediction of RNA-binding proteins from primary sequence by a support vector machine approach. RNA 10(3):355–368

Horton P, Park K-J, Obayashi T, Fujita N, Harada H, Adams-Collier CJ, Nakai K (2007) WoLF PSORT: protein localization predictor. Nucleic Acids Res 35(suppl 2):W585–W587

Jia C, Liu T, Chang AK, Zhai Y (2011) Prediction of mitochondrial proteins of malaria parasite using bi-profile Bayes feature extraction. Biochimie 93(4):778–782

Kumar M, Verma R, Raghava GPS (2006) Prediction of mitochondrial proteins using support vector machine and hidden Markov model. J Biol Chem 281(9):5357–5363

Li W, Godzik A (2006) Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics 22(13):1658–1659

Li ZR, Lin HH, Han LY, Jiang L, Chen X, Chen YZ (2006) PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. Nucleic Acids Res 34(suppl 2):W32–W37

Li ZC, Zhou XB, Lin YR, Zou XY (2008) Prediction of protein structure class by coupling improved genetic algorithm and support vector machine. Amino Acids 35(3):581–590

Muggleton SH (2006) 2020 Computing: exceeding human limits. Nature 440(7083):409–410

Murray CJL, Rosenfeld LC, Lim SS, Andrews KG, Foreman KJ, Haring D, Fullman N, Naghavi M, Lozano R, Lopez AD (2012) Global malaria mortality between 1980 and 2010: a systematic analysis. Lancet 379(9814):413–431

Oehring SC, Woodcroft BJ, Moes S, Wetzel J, Dietz O, Pulfer A, Dekiwadia C, Maeser P, Flueck C, Witmer K (2012) Organellar proteomics reveals hundreds of novel nuclear proteins in the malaria parasite *Plasmodium falciparum*. Genome Biol 13(11):R108

Prilusky J, Felder CE, Zeev-Ben-Mordehai T, Rydberg EH, Man O, Beckmann JS, Silman I, Sussman JL (2005) FoldIndex©: a simple tool to predict whether a given protein sequence is intrinsically unfolded. Bioinformatics 21(16):3435–3438

Quinlan JR (1993) C4 5: programs for machine learning. Morgan Kaufmann, Burlington, Massachusetts, United States

Ramana J, Gupta D (2010) Faapred: a SVM-based prediction method for fungal adhesins and adhesin-like proteins. PLoS ONE 5(3):e9695

Saeys Y, Inza I, Larrañaga P (2007) A review of feature selection techniques in bioinformatics. Bioinformatics 23(19):2507–2517

Shamim MTA, Anwaruddin M, Nagarajaram HA (2007) Support vector machine-based classification of protein folds using the structural properties of amino acid residues and amino acid residue pairs. Bioinformatics 23(24):3320–3327

Shen H-B, Chou K-C (2008) PseAAC: a flexible web server for generating various kinds of protein pseudo amino acid composition. Anal Biochem 373(2):386–388

Singh GP, Dash D (2008) How expression level influences the disorderness of proteins, vol 371. Elsevier, Amsterdam

Smialowski P, Frishman D, Kramer S (2010) Pitfalls of supervised feature selection. Bioinformatics 26(3):440–443. doi:10.1093/bioinformatics/btp621

Uversky VN, Gillespie JR, Fink AL (2000) Why are "natively unfolded" proteins unstructured under physiologic conditions? Proteins 41(3):415–427

Vapnik V (1999) The nature of statistical learning theory, 2nd edn. Springer, Heidelberg

Verma R, Varshney G, Raghava GPS (2010) Prediction of mitochondrial proteins of malaria parasite using split amino acid composition and PSSM profile. Amino Acids 39(1):101–110

Wang P, Xiao X, Chou K-C (2011) NR-2L: a two-level predictor for identifying nuclear receptor subfamilies based on sequence-derived features. PLoS ONE 6(8):e23505

Yu L, Liu H (2003) Feature selection for high-dimensional data: a fast correlation-based filter solution. In: Machine learning-international workshop then conference, 2003, p 856