OXFORD UNIVERSITY PRESS | Bioinformatics

isGPT: An optimized model to identify sub-Golgi protein types using SVM and random forest based feature selection

Journal:	*Bioinformatics*
Manuscript ID	Draft
Category:	Original Paper
Date Submitted by the Author:	n/a
Complete List of Authors:	Rahman, M. Saifur; Bangladesh University of Engineering and Technology, Computer Science and Engineering Rahman, Md. Khaledur; Bangladesh University of Engineering and Technology, Computer Science and Engineering; United International University, Computer Science and Engineering Kaykobad, M.; Bangladesh University of Engineering and Technology, Computer Science and Engineering Rahman, M. Sohel; Bangladesh University of Engineering and Technology, Computer Science and Engineering
Keywords:	Bioinformatics, Feature selection, Machine learning, Proteins

Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

Subject Section

isGPT: An optimized model to identify sub-Golgi protein types using SVM and random forest based feature selection

M. Saifur Rahman[1], Md. Khaledur Rahman[1,2], M. Kaykobad[1] and M. Sohel Rahman[1,*]

[1]Department of CSE, BUET, ECE Building, West Palasi, Dhaka-1205, Bangladesh and
[2]Department of CSE, BUET, Shatmasjid Road, Dhaka-1209, Bangladesh.

[*]To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: The Golgi Apparatus (GA) is a key organelle arbitrating many proteins within the eukaryotic cell. There are two types of GA proteins, namely, *trans*-Golgi protein and *cis*-Golgi protein. Any dysfunction of GA proteins can result in congenital glycosylation disorders and may lead to neurodegenerative and inherited diseases. So, the exact classification of GA proteins may contribute to drug development.
Results: We focus on building a new computational model that not only introduces easy ways to extract features from protein sequences but also optimizes classification of *trans*-Golgi and *cis*-Golgi proteins. After feature extraction, we have employed random forest model to rank the features. Subsequently, we have applied Support Vector Machine (SVM) to classify the sub-Golgi proteins. As the benchmark dataset is significantly imbalanced, we have applied Synthetic Minority Over-sampling Technique (SMOTE) to the dataset and have conducted experiments on both balanced and imbalanced versions. Our method, isGPT, achieves accuracy values of 95.4%, 95.9% and 95.3% for 10-fold cross-validation test, jackknife test and independent test and respectively.
Availability: The source code of isGPT, along with relevant dataset and detailed experimental results can be found at `https://github.com/srautonu/isGPT`.
Contact: msrahman@cse.buet.ac.bd
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Golgi Apparatus (GA) is one of the most important protein in eukaryotic cell which is found as a key organelle in protein synthesis along with some other elements of the cell. GA has three elements, namely, *cis*-Golgi, medial and *trans*-Golgi. *cis*-Golgi is responsible for receiving proteins, while *trans*-Golgi releases the synthesized proteins. The function of medial is to synthesize the received proteins from *cis*-Golgi. Endoplasmic Reticulum (ER) builds proteins and send out to the cell through GA. A side of GA facing ER (*cis*-side) captures those proteins (also called cargo proteins) for synthesis and send those out via the other side of GA facing plasma membrane (*trans*-side).

Any functional deviation of GA may result in adaptable disorders during synthesis process in medial which may further contribute to inheritable and neurodegenerative diseases (e.g., Su *et al.*, 2010). It is necessary to identify any rambling and damage in a timely manner to better understand the problem of GA dysfunction. The current methods of treating patients having these diseases include neuroprotective therapies and anti-inflammation which do not provide permanent solutions (Elsberry and Rise, 2000). Interestingly, exact identification of sub-Golgi proteins can open new sight for scientists to recognize the dysfunctions subscribed by Golgi proteins (Ungar, 2009). Thus, sub-Golgi (*cis*-Golgi vs. *trans*-Golgi) protein classification is very important for effective drug-development.

To the best of our knowledge, very few tools have been developed in the literature for sub-Golgi protein classification. Nevertheless, researchers nowadays are focusing on this topic and trying to build efficient classification models. Ding et al. proposed a special method to extract features from pseudo amino acid composition to predict the Golgi-resident proteins (Ding *et al.*, 2011). They achieved an accuracy of 74.7% using jackknife cross-validation test. Ding et al. further continued their previous work and proposed a g-gap dipeptide based feature extraction technique (Ding *et al.*, 2013). They used analysis of variance (ANOVA) test to select relevant features and applied Support Vector Machine (SVM) (Boser *et al.*, 1992) as the learner. For this, they obtained an improved accuracy of 85.4% using jackknife cross-validation. Jiao et al. presented a model that computed Position Specific Scoring Matrix (PSSM) based on physicochemical values of the engaged amino acid residues (Jiao and Du, 2016a). They further extended their work by combining with Chou's PseAAC which achieved better accuracy (Jiao and Du, 2016b). The former achieved an accuracy of 86.9% whereas the latter, 91%.

All the previous methods have worked with a small, highly imbalanced dataset having only 150 GA proteins where the number of *trans*-Golgi proteins is significantly lower than that of *cis*-Golgi proteins. Yang et al. have recently worked on an updated benchmark dataset having 304 sub-Golgi proteins for training and 68 sub-Golgi proteins for testing the classification model (Yang *et al.*, 2016). They have further applied Synthetic Minority Over-sampling Technique (SMOTE) (Chawla *et al.*, 2002) to balance the dataset. They have conducted experiments on both the imbalanced and balanced (i.e., SMOTE'd) datasets and have demonstrated improved accuracy for the latter. For feature selection, they have used Random Forest (RF) (Breiman, 2001) based recursive feature elimination method and then have applied RF as the learning method. Their model shows an accuracy of 88.5%, 93.8% and 90.1% for jackknife cross-validation, independent testing and 10-fold cross-validation, respectively.

Very recently, Ahmad et al. have also conducted similar kind of experiments though their feature construction, feature selection and learning algorithms are different (Ahmad *et al.*, 2017). They have applied Fisher feature selection method to select relevant features and K-nearest neighbor algorithm as the learner. Ahmad et al. have reported an accuracy of 94.9%, 94.8% and 94.9% on the balanced benchmark dataset for jackknife cross-validation, independent testing and 10-fold cross-validation, respectively. A brief qualitative discussion on this model and results reported thereof in (Ahmad *et al.*, 2017) is presented at a later section.

Exploring previous studies, we note that there is still room for improvement because even a small improvement in accuracy is highly demanding in bioinformatics tools. Improved accuracy can also contribute to better drug-development which is maintained by sensible computer-aided design. In this paper, we first construct a heavy set of features based on three feature construction techniques and then apply Random Forest (RF) algorithm on the constructed feature set. We select relevant features based on the importance score provided by the RF model. Then, we apply SVM on the selected features for both classification and regression analyses. Our tool, named *isGPT*, is evaluated based on several well-established performance metrics and demonstrates superiority over existing methods.

2 Methods

2.1 Dataset

We have collected the training and testing benchmark datasets from Yang et al. (Yang *et al.*, 2016), which have also been used by Ahmad et al. (Ahmad *et al.*, 2017) recently to measure the performance of their tool. The training dataset[1]) contains 304 sub-Golgi protein sequences among which there are 87 sequences of *cis*-Golgi type and 217 sequences of *trans*-Golgi type. The testing dataset is used for independent testing and it contains 13 *cis*-Golgi protein sequences and 51 *trans*-Golgi protein sequences. Both the training and testing datasets are highly imbalanced as they contain 71.4% and 80% *trans*-Golgi protein sequences, respectively.

2.2 isGPT Model Construction Overview

There exists similarity in Amino Acid Composition (AAC) among *cis*-Golgi proteins and *trans*-Golgi proteins (Yang *et al.*, 2016). Thus, traditional computational methods using Basic Local Alignment Search Tool (BLAST) (Altschul *et al.*, 1997) is inefficient to distinguish between the two. Machine learning methods can be a wise alternative option, which we have pursued in this paper. We process all sub-Golgi protein sequences through isGPT feature extraction step. In this step, several position independent and position specific features are extracted from the training dataset. All the features are obtained directly from the sequences. Among the position independent features are n-grams and n-gapped dipeptide based features, which have widely been used in literature. All these features are combined together to make a hybrid feature space. As we already know that the benchmark dataset is significantly imbalanced, we conduct Synthetic Minority Oversampling Technique (SMOTE) to make a balanced dataset. In the feature selection step, features are ranked based on Random Forest (RF) based importance score and only a subset of the top-ranked features are selected based on 10-fold cross validation performance. Finally, Support Vector Machine (SVM) is applied on the selected features to compute the final predictor. A diagram of our model construction work-flow has been presented in a supplementary file.

2.3 Feature Extraction Techniques

The features we have extracted can largely be divided into two categories: position independent and position specific. Among the position independent features are Amino Acid Composition (AAC), Dipeptides (Dip), Tripeptides and n-Gapped-Dipeptides (nGDip). These features do not depend on any specific position in the amino acid sequence. These has been used in literature, albeit with a slight variation. The position specific feature, on the other hand, is something that is introduced in this paper. We describe each of these feature construction techniques below. In describing the feature types, we have followed the nomenclature from (S Bernardes, 2013).

Amino Acid Composition (AAC): Amino Acid Composition (AAC) of a protein sequence means the normalized frequencies of the 20 native amino acids. This can thus contribute 20 features in a feature vector. After counting the frequency of each amino acid residue, normalization is done by dividing by the length of the sequence.

We have calculated the average AAC in the training dataset for *cis*-Golgi and *trans*-Golgi proteins (see Fig. 1). We see that there is only slight difference in each amino acid ratio between *cis*-Golgi and *trans*-Golgi proteins. This indicates that AAC alone is unlikely to be able to categorize an unknown sub-Golgi protein sequence. As such, we have employed several other feature extraction techniques.

[1] Yang et al. constructed this dataset from Universal Protein Knowledgebase (UniprotKB) (Uniprot-Consortium,

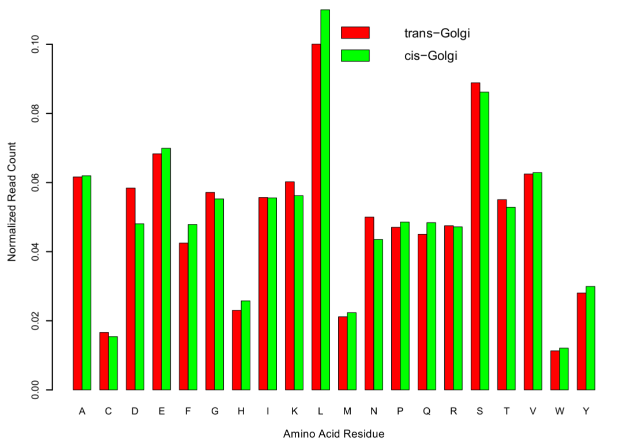

Fig. 1. Amino Acid Composition (AAC), on average, for the different sub-Golgi protein classes in the training dataset.

Dipeptides (Dip): The Dipeptides (Dip) feature type (also known as Dipeptide Composition (DPC)) computes the normalized frequency of adjacent amino acids within the sequence. This feature type provides into the feature vector some sequence-order information and has been successfully used in several protein related studies including the recent work of (Ahmad *et al.*, 2017). Dip thus contributes 400 features to our feature vector.

Tripeptides: We have similarly applied the notion of Tripeptides to extract another 8000 features. Observe that all these feature types derive from the generalized form of *n-grams* feature type where frequency of n-length peptides is used as a feature vector. In our study, we extract a total of 8420 n-grams features, for n = 1, 2 and 3.

n-Gapped-Dipeptides (nGDip): In the n-Gapped-Dipeptides (nGDip) feature type (also known as the Gapped Di-peptide Composition (GDPC)), we count the frequency of amino acid dipeptides such that the amino acids are separated by n positions. The frequency is normalized, dividing by the total number of nGDip. (i.e. $L - n - 1$ for a sequence of length L). This can contribute 400 features to the feature vector. The motivation for this feature type stems from the belief that the gap between any two amino acids may carry significant information about the protein (Chang *et al.*, 2008). Notably, both Ding et al. Ding *et al.*, 2011, 2013 and Yang et al. Yang *et al.*, 2016 have also utilized this feature scheme in their works. Yang et al. called it *g-Gap Dipeptide Composition* and used $g = 3$ only. In our work, rather than considering one specific gap, we have used (GDPC) feature type for gaps of upto 15 positions. Thus we get a total of $15 \times 400 = 6000$ n-Gapped-Dipeptides features.

Note that, for some features, all the samples of sub-Golgi protein sequences may produce 0 frequency. Such features will naturally have no effect on learning model. We have carefully removed these from the feature vector. Subsequently the n-grams feature count reduced to 8365.

Position Specific Features (PSF): PSF identify whether specific n-grams occur in specific positions in the protein sequence. The value of each such feature in any sequence will therefore be either 0 or 1. For a sequence of length L, the feature space size can be as large as $L \times 20^n$. However, the actual size may be considerably smaller depending on the sample size. For small sample sizes, many of the features will not have discriminating scores and will not be part of the final feature vector.

Like in the case of position independent features, we wanted to consider n-grams for $n = 1$, 2 and 3 in case of PSF as well. However, the feature space became too large for the computing power and the memory at the disposal of the machines we have used. As such, rather than considering each position of the sequence, we are motivated by the concept of Split Amino Acid Composition (SAAC), which was also used by Ahmad et al. (Ahmad *et al.*, 2017). In SAAC, a protein sequence is split into three parts: 25 residues at the N-terminus, the center part and the 25 residues at the C-terminus. Each portion is handled separately for feature extraction. In our case, we construct the PSF only from the N-terminus part. However, even with this part, the feature space is still too large. Therefore, we considered only the first 10 positions of the N-terminus part. In this way, we constructed a total of 5056 position specific features. So, counting all types of features, we have extracted a total of $8365 + 6000 + 5056 = 19421$ features.

2.4 Feature Selection Techniques

It will be computationally expensive to work with such a large feature vector, both during the learning phase as well as the prediction phase. Besides, all features may not always be effective in the learning model (Saeys *et al.*, 2007). As such, we need to select a set of relevant features that will contribute to the learning model in improving accuracy.

Yang et al. (Yang *et al.*, 2016) used Random Forest-Recursive Feature Elimination (RF-RFE) which is a wrapper method. Ahmad et al. (Ahmad *et al.*, 2017), on the other hand, applied a filtering approach, using the fisher selection technique. In this paper, we have employed a composition of filter and wrapper approaches. A very brief discussion on different approaches is presented in the supplementary file.

Filtering Phase: In the filtering phase, we apply Random Forest (RF) on the entire feature set to generate a model. Through this model creation, the RF algorithm is able to set an importance score (*MeanDecreaseAccuracy* to each of the input features. This importance score indicates the global importance over all out-of-bag cross validated predictions and is more robust as it is averaged over all predictions for a given feature variable. The importance score is used to rank the features and subsequently filter our irrelevant features.

In Figure 2, we see that when we take all the features into account, the summation of importance score for n-Gapped-Dipeptides (nGDip) based features is quite high. However, for the n-grams based features as well as position specific features (PSF) this sum is in fact negative. Overall, only the top 2985 features have positive importance score. From feature 2986 up to feature 15980, the importance score of each feature is 0. Beyond that, the scores actually become negative. Therefore, we further examine the top 3000 features. When we select this feature subset, the total importance scores for all three feature types are positive. Among these 3000 features, there are 2105 nGDip features, 52 PSFs and 843 n-grams.

Wrapper Phase: Subsequently we apply the wrapper phase. Feature filtering does not consider inter-dependencies amongst the features. To account for it, instead of directly selecting the 3000 features, we further experimented with the top 3500, 3000, 2500, 2000 and 1500 features by training SVM regression models on the benchmark dataset as well as on the set balanced with SMOTE.

In Figure 3, we have reported the Receiver Operating Characteristics (ROC) curves from these experiments, as obtained using 10-fold cross validation. Since the dataset is imbalanced, ROC-Curve alone is not able to identify the significance of selected features. In fact it has been argued in the literature that for imbalanced datasets, Precision Recall Curve (PR-Curve) is of more significance than ROC-Curve (Davis and Goadrich, 2006). Thus, we also report PR-Curve in Figure 3.

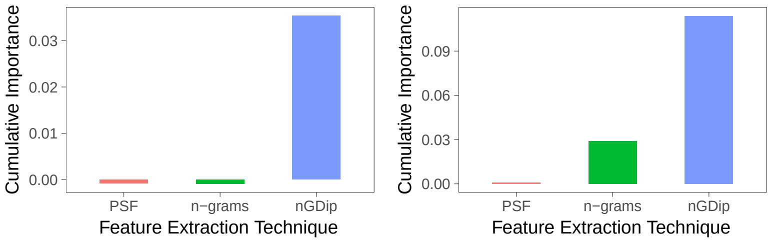

Fig. 2. Categorized feature importance based on MeanDecreaseAccuracy of Random Forest model. The aggregate MeanDecreaseAccuracy is better for top 3000 features (right) compared to all features (left). PSF: Position Specific Features, n-grams: Combination of AAC, dipeptide and tripeptide composition features, nGDip: n-Gapped Dipeptides.

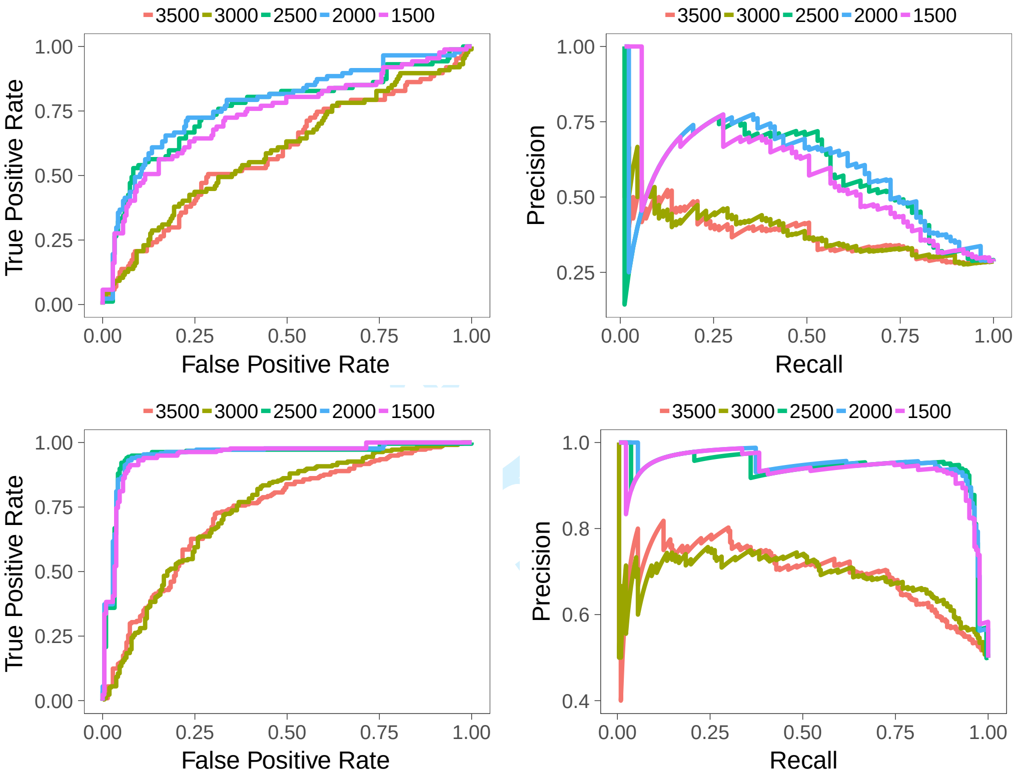

Fig. 3. ROC-Curves (left) and PR-Curves (right): The curves are generated by regression analysis with 10-fold cross validation, using top 3500, 3000, 2500, 2000 and 1500 features, respectively. The benchmark (imbalanced) dataset was used to generate the top curves. For the bottom curves, the dataset was balanced using SMOTE.

The close the ROC curve is to the top-left corner of the graph, the better is the performance of the model. On the other hand, the PR curve should be as close to the top-right corner as possible. Therefore, from the curves of Figure 3, it is clear that the performance with 3500 or 3000 features is much inferior compared to the other feature subsets. The curves further demonstrate the importance of balancing the dataset. Perhaps a better articulation of these points are in Table 1, where the area under the ROC and PR curves for the different settings are recorded.

Subsequently, we have further examined the feature space comprising of the top-ranked 1500 to 2800 features. First, we ran SVM with 10-fold cross validation using the top 2800 features. The C parameter of the regularization term in the *Lagrange formulation* of SVM was varied from the set 0.3, 1, 3, 10, 30, 100. Thus 6 different models were constructed and we evaluated their performance. Then, from the feature set, we eliminated the least ranked 50 features, recomputed 6 more models the same way and measured their performance. We repeated this process until the feature subset size was reduced to 1500. Thus a total of 162 models were evaluated. We finally selected the combination of C and feature subset that yield the best performance. This wrapping step was applied independently both in classification and regression analysis with the native (imbalanced) dataset as well as the set balanced with SMOTE.

Table 1. Area under ROC and PR curves for different number of top-ranked features selected.

Number of	Without SMOTE		With SMOTE	
Features	AUCROC	AUPR	AUCROC	AUPR
3500	0.55	0.33	0.73	0.68
3000	0.59	0.37	0.74	0.68
2500	0.75	0.53	0.95	0.95
2000	0.78	0.60	0.95	0.95
1500	0.75	0.57	0.95	0.95

2.5 Evaluation Metrics

Several testing methods exist that can assess the quality of the learning model while it is being trained as well as after the training has been completed. For instance, jackknife cross-validation, 10-fold cross-validation test, independent test etc. have been applied to bioinformatics tools in the literature to evaluate the performance, credibility and consistency. To evaluate isGPT, we have used all three testing methods mentioned above. It is extremely important to choose a good set of performance metrics to assess the predictive performance of a trained model. In this paper, we have used accuracy, precision, recall, sensitivity, specificity and Matthew's Correlation Coefficient (MCC), which are well-established performance metrics. These metrics are calculated using a confusion matrix which can be generated based on true classes and predicted classes. Additionally, we have analyzed Area Under Receiver Operating Characteristic Curve (ROC-Curve) and Area Under Precision-Recall Curve (AUPR-Curve). Details of these methods and metrics are provided in a supplementary file.

2.6 Experimental Setup and Packages

We have conducted experiments using R language (version 3.2.1) on three different machines, namely, a Desktop computer with Intel Core i3 CPU @ 1.90GHz x 4, Ubuntu 15.10 64-bit OS and 4 GB RAM, a Desktop computer with Intel Core i7 CPU @ 3.30GHz x 4, Windows 7, 64-bit OS and 8 GB RAM and a server machine with Intel Xeon CPU E5-4617 0 @ 2.90GHz x 6, Ubuntu 13.04 64-bit OS, 15 MB L3 cache and 64 GB RAM. To construct the isGPT model, we have used Random Forest (RF) and SVM machine learning algorithms. These are available respectively from the R packeges **randomForest** and **e1071**. Default parameter settings was used for the Random Forest. For SVM, the cost parameter was explored from the set 0.01, 0.03, 0.10, 0.30, 1, 3, 10, 30, 100. We found that the value of 0.3 provided the best generalization. Hence this parameter was used in the final model, while all other parameters were set to default values.

As discussed earlier, RF model has been used for feature selection, while SVM is used to learn the model. Since our training set is relatively small, we have used linear kernel function in SVM to avoid overfitting. All code have been written in R language where we have used some available R packages. We have also used **ROCR**, **caTools** R packages for performance evaluation of our model. For balancing the dataset, we used an implementation of SMOTE from *Weka 3 Data Mining Software* (Hall *et al.*, 2009; Frank *et al.*, 2016).

3 Results and Discussion

In this section, we describe several experiments and analyze their results. Some previous studies in a separate area of bioinformatics showed that regression analysis may perform better than classification models (Doench *et al.*, 2016; Rahman and Rahman, 2017). As such, we have compared results from both the regression analysis and the binary classification. We have also compared the results of our proposed technique with state-of-the-art methods.

3.1 Impact of Feature Extraction Techniques

To analyze the efficacy of the different feature extraction techniques, we take a closer look at the top 2500 features. In this subset, there are 1772 nGDip features, 34 PSF features and 694 n-grams features. With the SMOTE-balanced dataset, we trained 3 different SVM regression models using each of these 3 subsets of features. In another model, we trained with all the 2500 features. In Figure 4, the accuracy and MCC values from these 4 models area compared in the left side graph. The nGDip feature extraction technique is a clear winner over the other two, while the combination of all performs slightly better than that.

Table 3. Optimal parameters for classification and regression models of isGPT, based on 10-fold and jackknife cross-validation results

Model Type	Number of Features	C	Threshold
C w/ S	2800	10	N/A
C	2050	1	N/A
R w/ S	2800	1	0.58
R	2250	10	0.44

Note, however, that in the above comparison, the size of the feature vectors was widely different. Therefore, we conducted another experiment where we trained 3 different SVM regression models using top 2500 features of the 3 individual feature extraction techniques. We compare the performance of these models to the combined model in the right side graph of Figure 4. The superiority of combined feature space over the individual feature spaces hold in this setting as well. PSF, n-grams and nGDip feature extraction techniques individually achieve accuracy values of 93%, 93% and 94%, respectively. When the combined feature space is used instead, the accuracy increases to 95%. Similarly the MCC increases from respective individual values of 0.86, 0.88, 0.86 to 0.91 for the combined feature space.

3.2 Impact of data imbalance in isGPT learning model

As mentioned in the ‘Data’ section, the benchmark dataset is highly imbalanced. Both Yang et al. (Yang *et al.*, 2016) and Ahmad et al. (Ahmad *et al.*, 2017) reported that the dataset balanced using Synthetic Minority Over Sampling Technique (SMOTE) performs better than imbalanced dataset to classify the sub-Golgi proteins. To be consistent with their approach, we too have applied SMOTE to balance the data by increasing the number *Cis*-Golgi data points to 217. To examine the impact of balancing, we have conducted experiments both before and after balancing and then compared the results. In both cases, we have run regression as well as classification models. As discussed earlier, we examined 162 different models in each experiment by varying the regularization parameter C and the feature vector size and measured performance using 10-fold cross validation. The models yielding the best accuracy were further validated using Jackknife cross validation. Subsequently, the best models, as determined by the jackknife accuracy, were applied to the separate test dataset for independent testing. In Table 2, we have summarized the results from our experiments. We have highlighted the best results in bold faced font. The impact of data balancing is clearly evident. Also, we see that the regression model performs better than the classification model. So, we have subsequently compared the results from the regression model on the SMOTE-balanced dataset with previous studies.

In Table 3, we have recorded the optimal parameters for each model. These include the number of features and the C constant of the regularization term in SVM. For regression models, the class discriminating threshold is also recorded. These values are obtained based on the 10-fold and jackknife cross validation results.

3.3 Comparison between isGPT and Existing Techniques

To show the effectiveness and efficiency of isGPT, we need to compare the results of isGPT with the existing methods. In Table 4, we have reported a comparison of results among isGPT regression model and previous methods. The results reported for isGPT as well as for Yang *et al.*, 2016 and Ahmad *et al.*, 2017 are for the same benchmark dataset, after balancing was performed using SMOTE. The work in Ding *et al.*, 2011 and Ding *et al.*, 2013 uses an earlier dataset of smaller size.

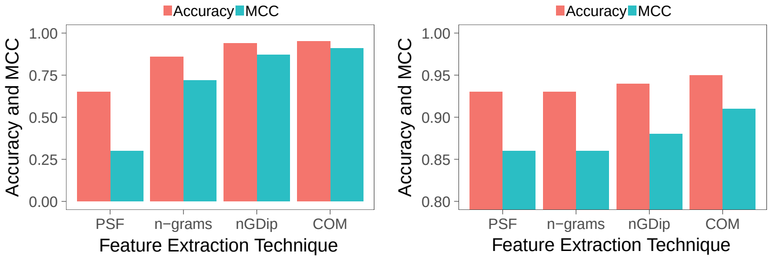

Fig. 4. Accuracy and Matthew's Correlation Coefficient (MCC) of different feature extraction techniques. The results are obtained from 10-fold cross validation of SVM regression model trained on the benchmark dataset balanced using SMOTE. PSF: Position Specific Features, n-grams: Combination of AAC, dipeptide and tripeptide composition. nGDip: n-Gapped-Dipeptides. COM: Combination of all the feature extraction techniques. The left figure is generated using the features of specific feature space that are within top 2500 positions in the combined space. In the right figure, for each feature space, corresponding top 2500 features are used.

Table 2. Comparison of classification and regression models of isGPT. In the Type column, 'C' and 'R' are used to represent classifcation and regression respectively. The 'w/ S' prefix is added if the model was computed on the dataset balanced with SMOTE. Acc: Accuracy, Sn: Sensitivity, Sp: Specificity, MCC: Matthew's Correlation Coefficient.

Type	10-fold Cross-Validation				Jackknife Cross-Validation				Independent Test			
	Acc	Sn	Sp	MCC	Acc	Sn	Sp	MCC	Acc	Sn	Sp	MCC
C w/ S	94.7	**95.9**	93.6	0.89	94.9	**95.9**	94.0	0.90	93.8	69.2	**100**	0.80
C	80.3	46.0	94.0	0.48	80.6	48.3	93.6	0.49	**95.3**	76.9	**100**	**0.85**
R w/ S	**95.4**	95.4	**95.4**	**0.91**	**95.9**	**95.9**	**95.9**	**0.92**	**95.3**	**84.6**	98.0	**0.85**
R	80.9	48.2	94.0	0.50	81.9	56.3	92.2	0.53	92.2	76.9	96.1	0.75

Note that we have highlighted the best results in bold faced font. We have reported results for jackknife cross-validation, independent test and 10-fold cross-validation. 10-fold cross-validation results are absent in (Ding *et al.*, 2011); both independent testing results and 10-fold cross-validation results are absent for the web-server based predictor of (Ding *et al.*, 2013). As such, we have marked the corresponding cells in the table by '-' symbol.

We see that isGPT achieves an accuracy of 95.9%, 95.3% and 95.4% for jackknife cross-validation, independent testing and 10-fold cross-validation, respectively. In comparison, the previous best method (Ahmad et al. (Ahmad *et al.*, 2017)) respectively achieved an accuracy of 94.9%, 94.8% and 94.9%. So, in all cases, isGPT shows improved performance.

In terms of MCC, isGPT demonstrates superiority in jacknife and 10-fold cross-validation - compare isGPT's respective scores of 0.92 and 0.91 to the previous best: 0.90 and 0.90. In case of independent testing, the MCC score of isGPT is slightly behind than that of (Ahmad *et al.*, 2017). However, we believe that the latter value might have been erroneously reported. This is elaborated in the 'Discussion' section.

Overall, it is evident that isGPT performs better than all previous methods.

3.4 Discussion

In this section, we make a discussion on the results we have obtained, previous results as well as key differentiation between our work and state-of-the-art. During our comparative analysis with earlier work, we attempted to check the consistency of earlier results. As we know, the independent dataset has 13 *cis*-Golgi and 51 *trans*-Golgi proteins. Since *cis*-Golgi class has lesser data, conventionally it should be the positive class in a binary classification model. Therefore, $P = 13$ and $N = 51$, where P (N) represents positive (negative) class. From the accuracy, sensitivity and specificity values, we can now find out the TP (True Positive), TN (True Negative), FP (False Positive), FN (False Negative) in the earlier works.

From the data reported by Yang et al., since sensitivity = TP/P, we find that $TP = 11.99 \approx 12$. Therefore, $FP = 1$. Similarly, from the specificity data, we can find, $TN = 47.99 \approx 48$. From accuracy data, we can find that $TP + TN = 60$, which is consistent with the already obtained values of TP and TN. Now, we can further find that $FN = P - TP = 1$ and $FP = N - TN = 3$. Plugging these values into MCC equation gives us 0.82. So, Yang et al.'s data is consistent.

Now, let us complete the same exercise for the results reported by Ahmad et al. From the sensitivity data, we can find that $TP = 12.22 \approx 12$. If we accept it to be 12 then the sensitivity should have been 92.3, not 94. Similarly, from the specificity data, we can fine that $TN = 47.8948 \approx 48$. If we take it as 48 then specificity should have been 94.1, not 93.9. So, in both cases we find some inconsistency. Perhaps, Ahmad et al. took *trans*-Golgi to be the positive class. In that case, we should have $P = 51$ and $N = 13$. We can then calculate $TP = 47.94 \approx 48$ and $TN = 12.21 \approx 12$. Like before, the rounding off error seems too high. In both scenarios, plugging the values into the MCC equation yields, 0.82. But, the value reported in the paper is 0.86. Thus, some inconsistency has been introduced in the reported data of Ahmad et al. In fact, they made another minor reporting error: in their paper the data from (Ding *et al.*, 2011) and (Ding *et al.*, 2013) have been swapped.

Now onto a discussion about the class discriminating threshold in the isGPT regression mode. In the regression model trained with the imbalanced dataset, the accuracy and MCC values in independent testing is best when the threshold is between 0.41 to 0.47. The optimal threshold (0.44), as chosen by the cross validation methods, does fall in this range. This is not the case in the regression model trained with the SMOTE-balanced dataset. In this case, the 0.58 threshold did not yield the best performance in the independent testing. Instead, we had to set the threshold to 0.40.

To better analyze the impact of threshold, the response of different performance metrics in the independent testing while changing the threshold has been plotted in Figure 5. The Left side graph therein does confirm that the thresholds in the range 0.41 to 0.47 produce maximum MCC as well as accuracy for the independent testing in case of the

Table 4. Comparison of isGPT regression model with previous methods. Acc: Accuracy, Sn: Sensitivity, Sp: Specificity, MCC: Matthew's Correlation Coefficient.

Tools	Jackknife Cross-Validation				Independent Testing				10-fold Cross-Validation			
	Acc	Sn	Sp	MCC	Acc	Sn	Sp	MCC	Acc	Sn	Sp	MCC
Ding *et al.*, 2011	74.7	69.6	79.6	0.52	-	-	-	-	-	-	-	-
Ding *et al.*, 2013	85.4	73.8	90.5	0.65	85.9	69.2	90.2	0.58	-	-	-	-
Yang *et al.*, 2016	88.5	88.9	88.0	0.76	93.8	92.3	94.1	0.82	90.1	90.8	89.4	0.80
Ahmad *et al.*, 2017	94.9	**97.2**	92.6	0.90	94.8	**94.0**	93.9	0.86	94.9	**97.2**	92.6	0.90
isGPT	**95.9**	95.9	**95.9**	**0.92**	**95.3**	84.6	**98.0**	**0.85**	**95.4**	95.4	**95.4**	**0.91**

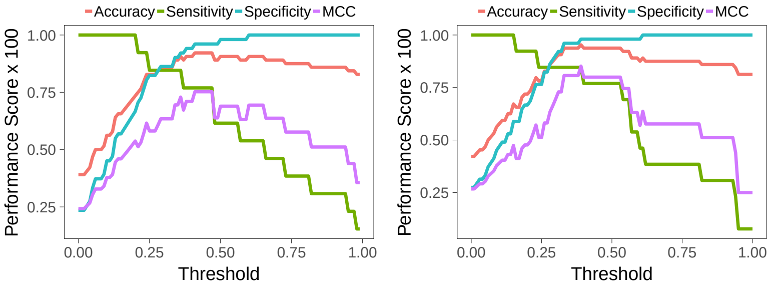

Fig. 5. Response of different performance metrics against variation of class discriminating threshold. The measurements are done on independent testing using a regression model trained on the imbalanced dataset (Left) as well as the SMOTE-balanced dataset (Right).

regression model trained using imbalanced benchmark dataset. The right side graph is plotted using a model trained with the SMOTE-balanced dataset. In this case, we observe a good range of threshold between 0.33 to 0.53, where both MCC and accuracy values are very high, with a peak MCC observed for threshold of 0.39. While the peak value (0.85) is very satisfactory, in the remaining parts of this plateau MCC remains competitive, between 0.80 to 0.81. Therefore, in our final predictor, we have set a default threshold of 0.50. The 10-fold cross validation with this threshold yields an accuracy of 94.7% and MCC of 0.90 which are competitive with state-of-the-art.

Finally, we discuss some key points that distinguish our work from the state-of-the-art methods. For example, we explored a large feature space, comprising of 19421 features and then selected 2800 features for training the model. The size of the selected feature set is way higher than earlier studies. Ahmad et al. Ahmad *et al.*, 2017 used 83 dimensional feature vector, while Yang et al. Yang *et al.*, 2016 selected 55 features. However, it is important to note that both the methods use features derived from the Position Specific Scoring Matrix (PSSM). The PSSM can be computed from PSI-BLAST (Altschul *et al.*, 1997) by searching the non-redundant protein database using at least three such, this is a time consuming step. Our approach, on the other hand, can extract all the necessary features from a target protein in a single pass along the sequence and then use the classifier to predict its class. On the server machine with Intel Xeon CPU E5-4617 0 @ 2.90GHz x 6, 64 GB RAM, PSSM generation for the smallest sequence (116 residues) in the test set (Accession Id: O95183) took around 10 minutes 30 seconds. For the largest sequence (Accession Id: Q55EI3, 4241 residues), almost 28 minutes were needed. In contrast, isGPT completed the prediction for the entire test set in less than two and half minutes. All of our source code, experimental results, SMOTE dataset processing and figure generation code are available in the following link: `https://github.com/srautonu/isGPT`.

4 Conclusion

In this paper, we present isGPT, an optimized model to identify sub-Golgi protein types. As the training dataset is significantly imbalanced, we use SMOTE to balance the dataset. We apply a combination of feature extraction techniques followed by a Random forest based novel feature selection technique. Finally, Support Vector Machine (SVM) is employed to train a binary classifier that can distinguish between *trans*-Golgi and *cis*-Golgi proteins. Our approach outperforms state-of-the-art techniques according to different performance metrics. Our predictor is available as a R script that can readily be applied to target protein sequences. In future, we plan to create a web service around isGPT so that it can be widely adopted.

Funding

The authors have no funding sources to declare.

References

Ahmad, J., Javed, F., and Hayat, M. (2017). Intelligent computational model for classification of sub-golgi protein using oversampling and fisher feature selection methods. *Artificial Intelligence in Medicine*, **78**, 14–22.

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, **25**(17), 3389–3402.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM.

Breiman, L. (2001). Random forests. *Machine learning*, **45**(1), 5–32.

Chang, J.-M., Su, E. C.-Y., Lo, A., Chiu, H.-S., Sung, T.-Y., and Hsu, W.-L. (2008). Psldoc: Protein subcellular localization prediction based on gapped-dipeptides and probabilistic latent semantic analysis. *Proteins: Structure, Function, and Bioinformatics*, **72**(2), 693–710.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, **16**, 321–357.

Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM.

Ding, H., Liu, L., Guo, F.-B., Huang, J., and Lin, H. (2011). Identify golgi protein types with modified mahalanobis discriminant algorithm and pseudo amino acid composition. *Protein and peptide letters*, **18**(1), 58–63.

Ding, H., Guo, S.-H., Deng, E.-Z., Yuan, L.-F., Guo, F.-B., Huang, J., Rao, N., Chen, W., and Lin, H. (2013). Prediction of golgi-resident protein types by using feature selection technique. *Chemometrics and Intelligent Laboratory Systems*, **124**, 9–13.

Doench, J. G., Fusi, N., Sullender, M., Hegde, M., Vaimberg, E. W., Donovan, K. F., Smith, I., Tothova, Z., Wilen, C., Orchard, R., *et al.* (2016). Optimized sgrna design to maximize activity and minimize off-target effects of crispr-cas9. *Nature biotechnology*, **34**(2), 184.

Elsberry, D. D. and Rise, M. T. (2000). Techniques for treating neurodegenerative disorders by infusion of nerve growth factors into the brain. US Patent 6,042,579.

Frank, E., Hall, M., and Witten, I. H. (2016). The weka workbench. online appendix for "data mining: Practical machine learning tools and techniques".

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, **11**(1), 10–18.

Jiao, Y.-S. and Du, P.-F. (2016a). Predicting golgi-resident protein types using pseudo amino acid compositions: Approaches with positional specific physicochemical properties. *Journal of theoretical biology*, **391**, 35–42.

Jiao, Y.-S. and Du, P.-F. (2016b). Prediction of golgi-resident protein types using general form of chou's pseudo-amino acid compositions: Approaches with minimal redundancy maximal relevance feature selection. *Journal of theoretical biology*, **402**, 38–44.

Rahman, M. K. and Rahman, M. S. (2017). Crisprpred: A flexible and efficient tool for sgrnas on-target activity prediction in crispr/cas9 systems. *PloS one*, **12**(8), e0181943.

S Bernardes, J. (2013). A review of protein function prediction under machine learning perspective. *Recent patents on biotechnology*, **7**(2), 122–141.

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

Su, L. J., Auluck, P. K., Outeiro, T. F., Yeger-Lotem, E., Kritzer, J. A., Tardiff, D. F., Strathearn, K. E., Liu, F., Cao, S., Hamamichi, S., *et al.* (2010). Compounds from an unbiased chemical screen reverse both er-to-golgi trafficking defects and mitochondrial dysfunction in parkinsonâŁ™s disease models. *Disease models & mechanisms*, **3**(3-4), 194–208.

Ungar, D. (2009). Golgi linked protein glycosylation and associated diseases. In *Seminars in cell & developmental biology*, volume 20, pages 762–769. Elsevier.

Uniprot-Consortium (-). Uniprot database. Last Accessed: 05 September 2017.

Yang, R., Zhang, C., Gao, R., and Zhang, L. (2016). A novel feature extraction method with feature selection to identify golgi-resident protein types from imbalanced data. *International journal of molecular sciences*, **17**(2), 218.