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

Subject Section

A computational tool for the prediction of lysine succinylation sites using Random Forest & SVM based feature selection

Joyanta Basak [1,*], Arman Ashkari [1] and M.Sohel Rahman [1]

[1] Department of Computer Science, BUET, Dhaka, 1000, Bangladesh

* To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Post-translational modification (PTM) is a biological mechanism that expands protein functionality. Among the PTMs, lysine succinylation substantially alter the structural and functional properties of cellular proteins. This PTM is suspected to be implicated in numerous diseases regarding heart and blood circulation. Recent proteomic studies spotted it in wide variety of both prokaryotic and eukaryotic organisms. In reality, little is known about this particular PTM and its implications. The experimental detection of lysine succinylation is expensive, time consuming and labor extensive. Which is why fast and robust computational tools are absolutely necessary to process rapidly increasing number of sequenced proteins and correctly categorize the concealed lysine residues. In this paper, we propose an efficient model that optimizes different sequence based representations to approximate discriminating function. This model reflects the efficacy of sequence information by achieving 89% accuracy, 84.9% sensitivity, 92.8% specificity, 0.781 Matthews correlation coefficient, 0.947 AUROC and 0.946 AUPR, which is the best overall performance achieved by any predictor in lysine succinylated site prediction.
Contact: joyantabasak13@gmail.com

1 Introduction

Proteins undergo a number of covalent and enzymatic modification referred as post-translational modification (PTM) following their bio-synthesis by ribosomes. Post-translational modifications change the structural and functional properties of both prokaryotic and eukaryotic cellular proteins by modifying existing functional groups or by introducing new functional groups forming covalent bond with amino acids in proteins [1]. Among all the α-Amino acids, lysine (Lys) is a frequent target to PTMs due to its role in constructing the spatial structure of proteins and regulating the protein functions [2]. A number a PTMs target lysine such as phosphorylation, acetylation, hydroxylation, methylation and ubiquitylation. Among these PTMs, succinylation is a major one, where a succinyl group (-CO-CH2-CH2-CO-) is transferred from succinyl-CoA to the specific lysine residue of the substrate protein [3]. Lysine succinylation is considered to orchestrate comparatively more substantial change to protein conformation, stability and physico-chemical properties as succinylation transfers larger structural moiety compared to methylation and acetylation [2]. Lysine succinylation has been reported to affect enzymes involved in mitochondrial metabolism including amino acid degradation, tricarboxylic acid cycle (TCA) and fatty acid metabolism [4]. Succinylated lysine sites were also found present in histones possibly regulating chromatin structures and functions [5]. This particular PTM is also suspected to be implicated in numerous diseases, such as hepatic, cardiac, and pulmonary diseases [6]. Despite such importance, full impact of lysine succinylation on cellular physiology is still unknown.

Succinylation was first detected by mass spectrometry [7] and protein sequence alignment [8]. Zhang et al. [9] identified lysine succinylation as a new PTM and reported in 2011 . Since then various large-scale proteomic methods are used to identify lysine succinylation in organisms, including pathogenic bacteria, protozoan and parasites, fungi , mammalian cells including human and mouse, and also in plants [10][2][11][12][13][14]. The amount of protein examined and lysine succinylation sites reported is still low with respect to its presence in cells of so many different species. Primarily because conventional methods require costly experimental setup along with labor-intensive and time-consuming experimental verification

of succinylated substrates. Experimental detection demands more resource as post-translational networks get complex. To overcome this challenge, high throughput in silico tools able to accurately identify succinylation sites are highly demanded.

In recent years, a number of efficient computational tools have been developed to predict lysine succinylation sites. Zhao et al.,2015 [15] proposed SucPred, a pioneering tool, based on SVM classifier trained in a positive sample only semi-supervised learning strategy (Psol) [16]. In their work, each lysine residue is described in respect of four sequence based feature namely the auto-correlation functions (ACF), the encoding based on grouped weight (EBGW), the normalized van der waals volume (VDWV) and the position weight amino acids composition (WAAC). Another pioneering work, SVM based predictor SuccFind [17] explored amino acid composition based feature AAC and CKSAAP encoding, physico-chemical features of amino acids adjacent to succinylation sites (AAindex) and evolutionary features through local sequence clustering. In an attempt to capture the intrinsic information of protein sequence, iSuc-PseAAC [18] incorporated the position specific amino acid propensity (PSAAP) into the general form of pseudo amino acid composition whereas iSuc-PseOpt [19] incorporated the sequence-coupling effects into the general pseudo amino acid composition. The former employed SVM classifier while the later is random forest based. pSuc-Lys [20] followed the methodology of iSuc-PseOpt but adopted ensemble random forest algorithm to construct model. Although iSuc-PseOpt and the most of the recent predictors used dataset collected from the CPLM database [21], SuccinSite[22] and SuccinSite2.0 [23] tested a larger dataset. Both of them are RF-based predictors. SuccinSite2.0 used profle-based composition of k-spaced amino acid pairs (pbCKSAAP) and orthogonal binary features while SuccinSite utilized three informative encoding features, i.e., CKSAAP, binary encoding and AAindex physico-chemical features. Succinsite2.0 also made an attempt to provide species specific insight. Among the later works, SucStruct [24] and PSSM-Suc [25] predictors are based on pruned C4.5 decision tree algorithm. SucStruct considered a comprehensive set of structural features to discriminate between succinylated and non-succinylated lysine residues while PSSM-Suc captured evolutionary information through profile bigrams extracted from position specific scoring matrix(PSSM). More recent predictor, SSEvolSuc [26] subsumed both structural and evolutionary features. They calculated the matrix of transition probabilities of each amino acid in protein segment to the three secondary structure conformations (helix, strand and coil) as well as computed the PSSM. Then extracted profile bigrams from both matrices as features to implement an Adaboost algorithm with decision stumps as weak classifiers.

Studying the previous works, we note that there is still plenty of scope to improve in terms of performance. Most of state-of-the-art predictors employed unilateral features without considering the possibility of containing redundant and noisy information risking loss of some other potential value information. Some feature extraction techniques require complex intermediate step. This intermediate step can be time consuming and susceptible to error. For example, PSSM derived features requires obtaining lengthy and time-consuming alignments against large sequence databases which cannot be speed up without losing predictive value [27]. To overcome these issues, we explored a heavy set of simple features extracted from the protein sequence only. The representative set of features was optimised in a two step feature selection strategy to retain relevant sequence derived information. Our final model provided a more accurate and comprehensible predictive performance over the recent state-of-the-art predictors achieving 89% accuracy, 84.9% sensitivity, 92.8% specificity and 0.781 MCC value as well as area under the ROC curve value of 0.947. Overall superior performance of our predictor establishes itself as a promising tool for both to identify lysine succinylated sites and to cut down on number of procedural steps for experimental validation.

2 Materials & Methods

2.1 Dataset

We collected a dataset from two freely accessible PTM databases CPLA 1.0 [28] and CPLM 2.0 [21]. This benchmark dataset consists of 893 unique protein sequences with each lysine residue annotated as succinylated or non-succinylated. Utilising this dataset enabled us to compare our predictor performance with most of the recent predictors namely SSEvol-Suc, PSSM-Suc, SucStruct etc., as they had employed the same benchmark dataset. To reduce bias, we employed CD-Hit program [29] to remove homologous proteins. CD-Hit reported a maximum pairwise sequence similarity of 48% where 99% of the succinylation sites were located in proteins less than 40% similar. By eliminating proteins with more that 40% pairwise alignment, we reduced the dataset to 665 proteins. This resulting dataset has very large number of non-succinylated lysine residues compared to succinylated lysine residues. For comparison, there is 1,782 succinylated sites (positive set) and 19,614 non-succinylated sites (negative set). Ratio between positive set to negative set is almost 1:11. This highly imbalanced dataset was treated with a balancing scheme discussed in 2.4.

2.2 Model Construction Overview

Traditional experimental detection is costly and time consuming. Computational approach using machine learning methods have showed to be a prudent alternative option. We have taken such approach to model our predictor with all information extracted solely from the protein sequence. We segmented all protein sequences and each segment was processed in a feature extraction step. Several position independent features were

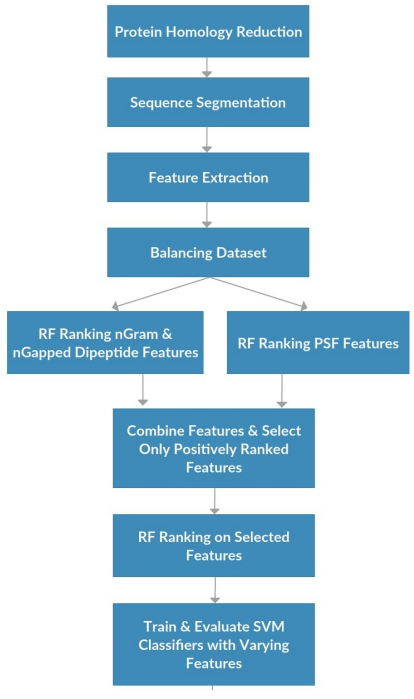

Fig. 1: Model construction workflow

extracted directly from the protein segments. These features were used to balance the highly imbalanced dataset using k-nearest neighbour cleaning treatment. Position dependent features were extracted from the balanced dataset which added more sequence information to tackle the classification problem. These position dependent features were not used

in KNN balancing as that would introduce high dimensions and would be computationally expensive. Amino acid composition, n-gram and n-gapped dipeptide based features were chosen as position independent features. Among all these features, a subset of most discriminatory features were sorted out by a 2-step feature selection step. In first step, position independent feature and Position dependent feature were ranked separately by Random Forest (RF) algorithm. A subset of each of the two feature type were picked out based on importance score. In second step, featurespace comprising the two subset of top-ranked features were merged to make a hybrid featurespace. Features in this hybrid featurespace were again ranked by Random Forest algorithm. Another subset of the top-ranked features were selected based on importance score. Finally, the predictor was computed by employing a Support Vector Machine (SVM) trained with the selected top discriminatory features. A diagram of our model construction is presented in fig.1

2.3 Feature Extraction Method

2.3.1 Sequence Segmentation

Each succinylated and non-succinylated lysine site was described by sequence based features in this study. Chou's peptide formulation was adopted to facilitate description which was widely used for studies such as enzyme specificity [30], signal peptide cleavage sites [31], protein-protein interaction [32]. According to Chou's scheme, a peptide containing candidate succinylation site can be represented as

$$S_\zeta(K) = A_{-\zeta}, A_{-(\zeta-1)}, A_{-(\zeta-2)}, ..., A_{-1}, K, A_1, ..., A_{\zeta-1}, A_\zeta$$

with lysine(K) in the centre where the subscript ζ is an integer, $A_{-\zeta}$ represents the ζ-th downstream amino acid residue from the center, A_ζ is the ζ-th upstream amino acid residue, and so forth. If the peptide contained less than ζ amino acids upstream or downstream, the incomplete half was substituted by mirroring the other half. Thus, each protein in benchmark dataset can be segmented into peptide sequences with lysine in the center. Each complete peptide sequence is to be categorized into two classes as:

$$S_\zeta(K) \in \begin{cases} S_\zeta^+(K), & \text{if center(K) is a succinylation site} \\ S_\zeta^-(K), & \text{otherwise} \end{cases}$$

Finally, the benchmark dataset can be constructed as:

$$\mathbb{S} = \mathbb{S}^+ \bigcup \mathbb{S}^-$$

where $\mathbb{S}^+$ contains all $S_\zeta^+(K)$ peptide sequences and $\mathbb{S}^-$ contains all $S_\zeta^-(K)$ peptide sequences.
The length of a peptide sample is $2\zeta + 1$. For different values of ζ different lengths of peptide segment will constitute the dataset. In this study, we have experimented with several lengths for ζ= {5,10,13,15,17,20}. The best result was achieved for $\zeta = 20$. Since our study showed predictive improvement with increased length of ζ, it may be argued that a better performance was achievable for a greater value of ζ. But due to complexity constraints the study was limited to $\zeta = 20$.

2.3.2 Feature Extraction Technique

We extracted position independent and position dependent features to capture sequence information. Position independent features are Amino Acid Composition (AAC), Dipeptides, Tripeptides and n-Gapped-Dipeptides (nGDip) which is also known as the Gapped Di-peptide Composition (GDPC). These features are widely used in literature of proteomic study. Feature types are presented according to the nomenclature described in S Bernardes et al. [33]. Position specific feature was first introduced in Rahman et al. [34] and their later works [35] showed it to be a useful feature in capturing sequence information. Each of these feature construction technique is briefly described below.

2.3.3 Amino Acid Composition (AAC)

Amino Acid Composition (AAC) is the normalised frequencies of each of the 20 native α-amino acids in the protein sequence. It can contribute upto 20 features to the feature vector. In our case, we normalised the feature value by dividing the count of the each amino acid with the length of the protein segment.

2.3.4 Dipeptides (DIP)

The Dipeptides (Dip) feature refer to the normalized frequency of adjacent amino acids within the sequence. It can capture some additional sequence information by contributing upto 400 features to the feature vector.

2.3.5 Tripeptides

Similar to Dip, tripeptide feature refer to the normalised frequency of three consecutive amino acids within the sequence. It can contribute upto 8000 features to the feature vector.
All these feature types derive from the generalized form of n-grams feature type where peptides of length n without gaps are considered. In our study, we extracted n-grams features, for $n = 1, 2, 3$. Although capturing higher order n-grams might have provided more sequence order information, it would have also increased computation complexity by introducing exponentially larger number of features to the feature vector.

2.3.6 n-Gapped-Dipeptides (nGDip)

The n-Gapped-Dipeptides (nGDip) feature type refers to the normalised frequency of two amino acids separated by upto n positions. It enables to represent discontinuous dipeptides. The feature value is normalized by dividing the count by the total number of possible nGDip with gap n. That is, for a sequence with length L, a nGDip is normalised by $L - n - 1$. This can contribute upto 400 dipeptide features for each value of n. Previous studies showed nGDip to carry significant information [36]. This also became evident in our future selection step. Although some works considered one specific gap [37], we incorporated nGDip features for gap upto windowsize (ζ) of sequence segments. Out of all nGDip features, we removed those with n=0 from the feature vector as they were already considered as dipeptide features.

2.3.7 Position Specific Features (PSF)

Position specific features (PSF) or position specific n-grams (PSN) refers to the occurance of specific n-grams in specific positions in the protein sequence. PSF value for a certain n-gram can be either 1 (presesnt) or 0 (absent) for a certain position in the sequence. Considering peptides of length n, there can be upto $L \times 20^n$ position specific features for a sequence length of L. However, actual number of PSF features can be much lower as each possible representation is present . We computed PSF considering n-grams for $n = 1, 2, 3$ as considering higher order n-grams could explode the featurespace. Even then the number of PSF features was very large. So we considered PSF features beginning inside the first 5 positions of downstream terminus.

Actual number of each type of features extracted from windowsize ζ= {5,10,13,15,17,20} segmented dataset is presented in the supplementary materials. Our best performing model trained on ζ=20 windowsized balanced dataset had 7885 n-gram features, 8000 nGDip features and 13115 PSF features.

2.4 Dataset Balancing

The benchmark dataset consists of 1,782 succinylated sites (positive set) and 19,614 non-succinylated sites (negative set). This highly imbalanced dataset with 1:11 ratio between positive set to negative set could lead to biased learning. Reducing the class imbalance is an important step to eliminate bias in supervised machine learning algorithms. Hence,

balancing the positive and negative sets is required. As we have ample amount of positive samples, oversampling the positive samples to balance would put enhanced burden on the computation. As such we decided to undersample the negative set to reduce imbalance. We used KNN-cleaning treatment method to under-sample dataset to balance. Chou [38] reintroduced KNN-cleaning treatment method in protein attribute prediction and it has been widely used in literature since then. KNN cleaning treatment works as follows.

- **Step 1:** Computing a distance measure between all the samples in the benchmark data set is required. In our study, we computed euclidean distances based on position independent features.
- **Step 2:** An initial cut-off is calculated by dividing the number of majority instances and minority instances. In our study, cutoff v=2 was initially set for omitting majority class instances. In simpler terms, we eliminated a negative sample if one of its 2 nearest neighbors is a positive sample based on previously calculated euclidean distance.
- **Step 3:** If imbalance between classes remained after second step, new threshold is computed by multiplying the initial cut-off with increasing integer. Then the majority class instances are omitted using newly computed cut-off. We repeated this step until both sets were almost similar in size. In our study, we iterated cut-off until threshold of v=42. That is, we removed negative samples when at least one positive sample was part of its 42 nearest neighbors. It resulted in a negative dataset of 1688 negative samples out of 19,614 negative samples.

It is worth noting that the minority class instances are not affected by KNN cleaning treatment and remains same. This algorithm only undersamples from majority class. This is an order $O(N^2d)$ algorithm where N is the number of data instances and d is the number of dimensions (number of features). Using all of the features found in feature extraction process would make the computation significantly time consuming. So only position independent features were used to calculate nearest neighbours.

2.5 Feature Selection

Employing large number of features to train classifier is computationally expensive. Moreover, performance of such predictor can be substandard as all features are not always relevant, as pointed out by a review [39]. So to develop an well performing predictor, we need to select the most relevant features that are able to express the intrinsic difference between the regions of succinylated and non-succinylated lysine. To identify relevant features, we first filtered the feature space by Random Forrest based feature filtering step as shown in fig 2. Then we employed a wrapper method to find out a more relevant subset of features.

2.5.1 Random Forrest Based Feature Filtering Step

Random Forest algorithm can compute importance of features by permuting out-of-bag estimation. First, it records the out-of-bag estimation error for each tree. Then, the error is recalculated by varying each variables of RF. Finally, mean decrease in accuracy is computed by taking average of difference between two errors over all decision trees and normalising it with standard deviation of the difference of the errors. Mean decrease in accuracy value of a feature signifies the feature's importance. As average of many trees are taken into consideration in RF algorithm, a single tree in Random Forest is less likely to considerably impact the prediction and thus decreasing the variance of the model. That in turn leads to more accurate ranking of the features.

Each protein segment in our dataset is described by 15885 position independent and 13115 position dependent features, 29000 features in total. Training a RF model with these many features is computationally infeasible with our available resources. To scale the high dimensions, we filtered the features in two steps.

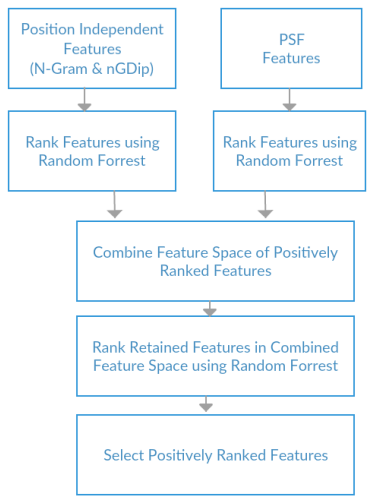

Fig. 2: Random Forrest based Feature Filtering Steps.

- **First Step:** We ranked position independent features and position dependent features separately. We found 9594 position independent features with positive importance score while rest 6291 position independent features with mean decrease in accuracy ≤ 0. We also found 2055 PSF features with positive importance score out of 13115 PSF features. We reduced the size of the both feature space by discarding the features with non-positive importance score from their respective feature vectors. Two significantly shortened feature spaces were found as a resultant.
- **Second Step:** We need to further investigate the relevancy of position independent and position dependent features among themselves. To achieve this, we combined the two future spaces into a hybrid feature space of total 11649 features. We ranked them using Rf and found 8029 features with positive importance score. We pruned the non-positive features out of the hybrid feature space. The resultant feature vectors were utilised in classification model construction.

RF based feature filtering reduced the size of the feature space by 72.3%. Figure 3 shows the reduction of individual feature types.

2.5.2 SVM Based Wrapper Method

Feature Filtering step finds a good feature subset independent of the classification model. In contrast, wrapper methods search for a model hypothesis within the feature subset search. In simpler terms, wrapper technique is a search procedure in the feature space which generate and evaluate various feature subsets. In our work, we systematically explored the feature space and evaluated each feature subset with a Support Vector Machine(SVM) classifier. Initially a feature matrix with top ranked 100 features was constructed. Then a SVM classifier with unit cost and linear kernel was trained and tested on those features in accordance with k-fold cross validation. Then we expanded the feature matrix adding next 100 features in the ranking and similarly evaluated the 200 top ranked feature set. We continued this process until we had explored the whole feature space.

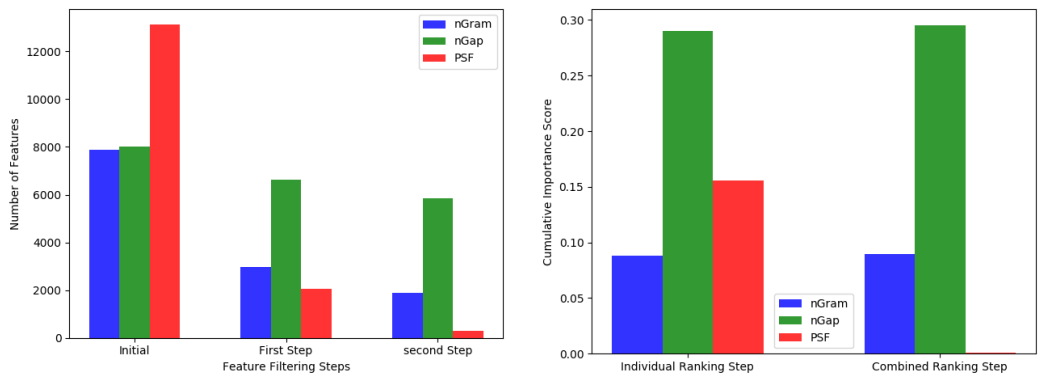

Fig. 3: Retention of individual feature types in filtering steps.(Left) Aggregate importance score of retained features of each type.(Right)

2.6 Experimental Setup & Packages

We conducted our experiments using R language (version 3.5.0) and java. Two machines were used throughout the study. • A Desktop computer with Intel Core i7-6500U CPU @ 2.50GHz×4, Windows 10, 64-bit OS and 8 GB RAM. • A server machine with Intel Xeon CPU E5-4617 0 @ 2.90GHz×6, Ubuntu 13.04 64-bit OS, 15 MB L3 cache and 64 GB RAM.

Implementation of K-NN cleaning treatment was done in java. All other scripts of the experimentation were written in R. Random Forest was used from the R package *randomForest* with default parameters. Support Vector Machine algorithm was used from the R package *e1071* with unit cost and linear kernel. we also used *ROCR* and *pracma* packages for performance analysis of our model and *ggplot2* package for plotting graphs. These packages were installed with all their dependencies in addition to pre-installed packages.

3 Results & Discussion

Evaluation of the predictor using well established testing methodologies and expressing its performance in benchmark metrics is instrumental to introspect its performance. The testing scheme and performance metrics applied are briefly discussed.

3.1 Testing Methods

There are several well established testing methodologies in the literature, most notably jackknife or leave-one-out cross validation, n-fold cross-validation and independent testing. Among them, 10-fold cross validation has been widely used in validating previous predictors. So we have also adopted 6,8,10-fold cross validation scheme to be able to compare with other state-of-the-art predictors. To conduct n-fold cross validation, the training set is divided into n equal parts. Among n parts, n-1 parts are used for training the model while the other one part is used for testing purpose. This process is repeated n times so that each part is used exactly one time for testing. The dataset can be partitioned into n parts in many different ways. Difference in partitioning can cause difference in n-fold cross validation results in each run. As such, we conducted n-fold cross validation 3 times and reported the average result. Jackknife test is essentially k-fold cross validation where k is the number of data instances. So jacknife test requires training the model k times which can be time consuming. Varying the number of features of 6 different window-sized dataset, We developed 349 SVM models for each of the cross validation scheme.

3.2 Performance Metrics

We analyzed our predictor performance in terms of accuracy, sensitivity, specificity and Matthew's correlation coefficient (MCC). These are widely used performance metrics in literature [40], [41]. We also plotted the receiver operating characteristic curve (ROC-Curve) and precision-recall curve (PR-Curve) to visualize the diagnostic ability of our binary classifier. In classifying the lysine residues into succinylated (positive) or non-succinylated (negative) class, each sample can be identified as one of the following cases based on the predicted and actual class of the sample.

- **case 1:** A positive test sample also predicted as positive is called true positive.
- **case 2:** A positive test sample wrongly predicted as negative is called false negative.
- **case 3:** A negative test sample wrongly predicted as positive is called false positive.
- **case 4:** A negative test sample also predicted as negative is called true negative.

Assume, the number of true positive, false negative, false positive and true negative instances in the data is denoted by TP, FN, FP and TN respectively. Then the relevant performance metrics are expressed as following equations.

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

$$Sensitivity = \frac{TP}{TP+FN}$$

$$Specificity = \frac{TN}{TN+FP}$$

$$Precision = \frac{TP}{TP+FP}$$

$$FalsePositiveRate = \frac{FP}{FP+TN}$$

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

$$Sensitivity = TruePositiveRate(TPR) = HitRate = Recall$$

$$FalsePositiveRate = FalseAlarmRate = 1 - Specificity$$

The receiver operating characteristic curve (ROC-Curve) depicts the trade off between hit rate and false alarm rate of a classifier. ROC-curve is a two dimensional graph in which the true positive rate (TPR) or sensitivity

SVM Models	Feature space	Number of Features	Accuracy	Sensitivity	Specificity	MCC	AUPR	AUROC
Model 1	PSF	1600	0.786	0.701	0.866	0.577	0.848	0.840
Model 2	Bigram, Trigram, nGDip	8200	0.883	0.843	0.921	0.768	0.942	0.941
Model 3	PSF, nGrams, nGDip	7900	0.890	0.849	0.928	0.781	0.946	0.947

Predictors	Accuracy	Sensitivity	Specificity	MCC	AUROC
iSuc-PseAAC	0.802	0.507	0.897	0.432	0.782
iSuc-PseOpt	0.874	0.688	**0.964**	0.708	0.946
pSuc-Lys	**0.908**	0.768	0.959	0.769	0.932
SucStruct	0.744	0.733	0.755	0.488	0.720
PSSM-Suc	0.819	0.816	0.824	0.639	0.817
SSEvol-Suc	0.875	**0.909**	0.837	0.750	0.942
Our Predictor(6-CV)	0.881	0.848	0.912	0.763	0.940
Our Predictor(8-CV)	0.886	0.845	0.925	0.774	0.945
Our Predictor(10-CV)	0.890	0.849	0.928	**0.781**	**0.947**

is plotted in Y axis and false positive rate (FPR) is plotted in X axis. As a ROC-Curve approaches left upper corner of the graph, it indicates improvement in hit rate and decline of false alarm rate. This increases the value for area under Roc-curve (auROC) signifying better performance [42].

The precision recall curve or the PR-curve depicts the the precision against the recall at various threshold settings. The top right corner of the graph is the optimum point. The closer the PR-curve is to that point, the better is the performance. It also increases the value for area under PR-Curve (auPR). ROC-curve together with PR-curve can accurately assess predictor performance even if class imbalance is present in the dataset [43].

3.3 Impact of Feature Selection Technique

We emphasized on extracting sequence information as much as possible. Prior to feature selection, 29000 features described each protein segment. Certainly all features do not carry same amount of discriminant information. Some might even be redundant or noisy. In our RF filtering step (described in 2.5.1), 39.6% position independent features (nGram, nGDip) and 84.33% PSF features was filtered out. Retained features have shown to contain some level of relevant information. Now three experiments were executed to evaluate efficacy of position independent and position dependent representations separately and in combination. All three experiments was conducted on same dataset and evaluated by 10-fold cross validation. In first experiment, positively ranked position dependent features were considered. According to the wrapper method (described in 2.5.2), we developed SVM models with expanding feature space by 200 features for successive models. Top 1600 features showed to be best position independent representation. In second experiment position independent features i.e., amino acid composition (AAC), bigram frequency, trigram frequency and n-gapped dipeptide (nGDip) features were considered. Similar SVM based wrapper phase found model developed with top 8200 features to perform best. These two experiments show that both position dependent and position independent features are capable to effectively discriminate between succinylation status of a lysine residue. But Those two types of features do not necessarily capture same information as their construction procedure is completely different. So our third experiment focused on their aggregate information. To access this information, the featurespace of previous two experiments were merged into a hybrid featurespace. Since some features might lose expressiveness in joint space, the hybrid featurespace was filtered. Subsequently, 11649 features were selected by their positive ranking, prunning out 30.08% features. In SVM based wrapper phase, the entire featurespace was searched thoroughly. The best performing SVM model suggested the top 7900 features constitute the most comprehensible feature set.

3.4 Comparison with Current Predictors

We measured the performance of our model in aforementioned metrics and a comparison with current state-of-the-art predictors is presented in 3. Although the current state-of-the-art predictors provide user friendly web server, it is not clear which proteins in the CPLM database were used to train them. Hence, it can be argued that an arbitrary validation set might not reflect comparative performance. So to eliminate any scepticism, we compared our results with the self-reported performance of respective predictors.

An effective predictor must be able to predict both succinylated sites and non-succinylated sites with high confidence. Evaluating predictor effectiveness by a single metric like accuracy alone can be deceptive in this case. Efficient detection of positive case and negative case is more reliably expressed by true positive rate or sensitivity and true negative rate or specificity respectively. In such case, area under ROC curve (AUROC) is a better indicator of general predictive performance [42] while area under PR curve (AUPR) provide complementary assessment for skewed dataset [43].

iSuc-PseAAC[18] achieved 89.7% specificity and 50.6% sensitivity. That means iSuc-PseAAC can detect most of the non-succinylated sites while it fails to detect roughly half of the succinylated site. A possible source of this bias might be the imbalance in dataset. its benchmark dataset consisted of 1167 positive samples and 3553 negative samples. This 3 times larger negative set could have biased the predictor to accurately classify negative instances in exchange of lowered hit rate. A later work by Jia et al., iSuc-PseOpt [19] also addressed the same issue and attributed it to the skewed dataset. To reduce imbalance, iSuc-PseOpt pruned out some negative samples using k-nearest neighbour cleaning treatment and increased positive samples by inserting hypothetical training samples (IHTS). Then developed the model with exactly balanced dataset. Yet, iSuc-PseOpt achieved 96.48% specificity while only 68.8% sensitivity suggesting additional source of bias. Both iSuc-PseAAC and iSuc-PseOpt segmented 896 proteins into smaller lysine centered sequence of size 15 and 31 respectively. Then they screened these segments with CD-Hit [29] to prune out samples with significant pairwise identity. CD-Hit was originally developed to cluster similar or homologous proteins. iSuc-PseAAC and iSuc-PseOpt did not present any viable intuition behind pruning similar samples. However, this step discards important sequence information that might be intrinsic to lysine succinylated regions by

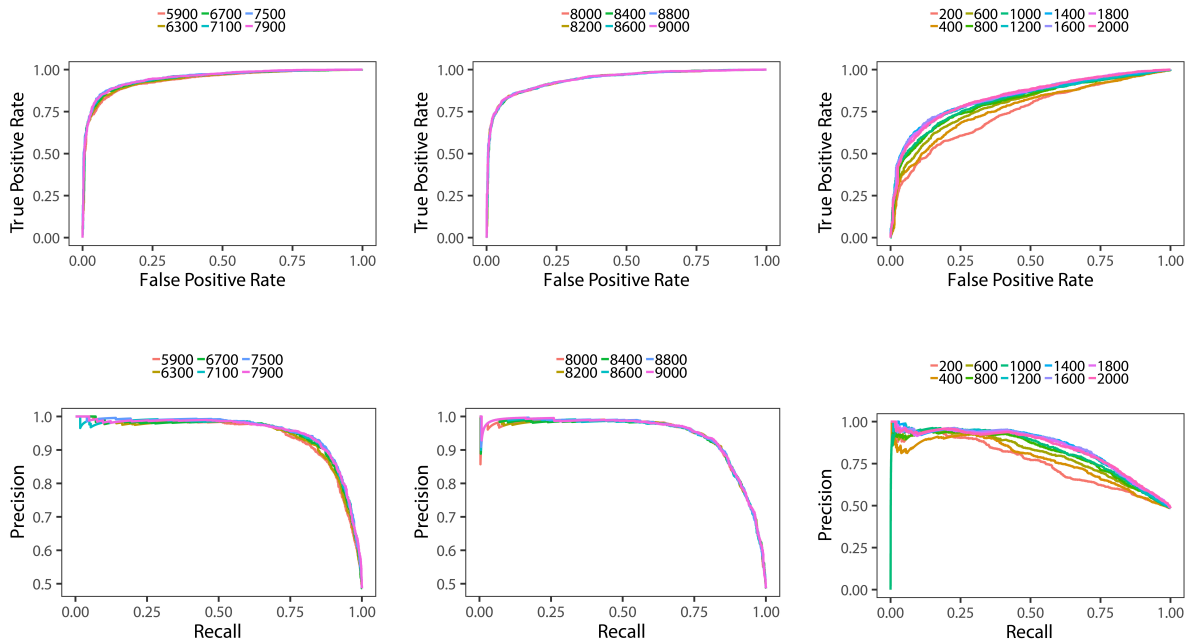

Fig. 4: ROC curve (top) and PR-curve (bottom) of some combined featured models, position independent featured models and PSF featured models (left to right) with varying number of features. The models were generated by 10-fold cross validation.

eliminating segments that are significantly smaller than proteins. In pSuc-Lys, Jia et al. [20] tried to minimize the bias by adopting an ensemble of random forests as classifying model. Five balanced dataset were prepared by combining positive samples with randomly selecting equal number of negative samples for each dataset. This asymmetric bootstrap approach along with ensemble algorithm improved the performance of pSuc-Lys. Still specificity is almost 25% higher than the sensitivity which firmly suggests a biased learning.

Decision tree based classifier SucStruct [24] used structural characteristics of peptide sequences as representations. These structural features were extracted via SPIDER2 tool which uses the sequence information . On the other hand, PSSM-Suc achieved comparatively better result by employing evolutionary information through profile bigrams on PSSM features. Current state-of-the-art predictor SSEvol-Suc benefited from both structural and evolutionary representation. However, extracting both structural features and profile bigrams requires calculating position specific scoring matrix (PSSM) first. PSSM calculation is an time consuming iterative process. PSI-BLAST (Position-Specific Iterative Basic Local Alignment Search Tool) derives a position-specific scoring matrix (PSSM) from the multiple sequence alignment of sequences detected above a given score threshold. Then the existing PSSM is used to search the database for new matches and it is updated in subsequent iterations if new sequences are detected. Accuracy and time requirement for PSSM generation may vary based on the nature of query proteins and reference database. For example, a few very large sequences can take longer than many smaller sequences as PSSM requires obtaining alignments against sequence databases multiple times [44]. Although the PSSM generation can be sped up using smaller reference protein set, it is likely to lose predictive performance in such case [27]. With increasing number of sequenced proteins in recent year, PSSM driven feature based prediction will obviously become more complex and resource consuming.

Our predictor exploited solely sequence information and achieved AUROC value 0.947 and MCC value 0.781, which are highest reported values in respective metrics among other predictors. Moreover, our model has shown 89% accuracy, 84.9% sensitivity and 92.8% specificity. In addition, we achieved 0.946 value for area under precision-recall curve (AUPR). High values for AUROC and AUPR suggests unbiased learning. This overall performance is superior to any other state-of-the-art predictors. Such feat can be attributed to,

- Meticulously Balanced dataset with protein homology reduction and KNN cleaning.
- Extensive extraction of simple features capturing important sequence information.
- Optimized feature selection retaining relevant sequence information and rigorous feature space search for most comprehensible representation.

Our experiments show that, model developed on only position independent features is better than most current predictors. Combining feature space with position specific features further pushed the performance. So it is evident that, position specific features hold some exclusive information even though they were extracted from relatively smaller region.

4 Conclusion

In this paper, we presented a robust model for effective classification of lysine succinylation sites. Our predictor exhibited performance superior to current state-of-the-art predictors in an unbiased comparison. our approach has established that sequence information alone can provide an approximation to discriminant function. Our efforts in PSF feature extraction on full sample sequences and K-NN cleaning considering full feature space was limited due to resource constraints. Nevertheless, this paper presents some prospective view that can be exploited in future works.

Acknowledgements

Funding

This work has been supported by the...

References

[1]Christopher T Walsh, Sylvie Garneau-Tsodikova, and Gregory J Gatto Jr. Protein posttranslational modifications: the chemistry of proteome diversifications. *Angewandte Chemie International Edition*, 44(45):7342–7372, 2005.

[2]Xiaolong Li, Xin Hu, Yujing Wan, Guizhen Xie, Xiangzhi Li, Di Chen, Zhongyi Cheng, Xingling Yi, Shaohui Liang, and Feng Tan. Systematic identification of the lysine succinylation in the protozoan parasite toxoplasma gondii. *Journal of proteome research*, 13(12):6087–6095, 2014.

[3]Ran Rosen, Dörte Becher, Knut Büttner, Dvora Biran, Michael Hecker, and Eliora Z Ron. Probing the active site of homoserine trans-succinylase. *FEBS letters*, 577(3):386–392, 2004.

[4]Jeongsoon Park, Yue Chen, Daniel X Tishkoff, Chao Peng, Minjia Tan, Lunzhai Dai, Zhongyu Xie, Yi Zhang, Bernadette MM Zwaans, Mary E Skinner, et al. Sirt5-mediated lysine desuccinylation impacts diverse metabolic pathways. *Molecular cell*, 50(6):919–930, 2013.

[5]Zhongyu Xie, Junbiao Dai, Lunzhi Dai, Minjia Tan, Zhongyi Cheng, Yeming Wu, Jef D Boeke, and Yingming Zhao. Lysine succinylation and lysine malonylation in histones. *Molecular & Cellular Proteomics*, 11(5):100–107, 2012.

[6]Matthew Alleyn, Mason Breitzig, Richard Lockey, and Narasaiah Kolliputi. The dawn of succinylation: A posttranslational modification. *American Journal of Physiology-Cell Physiology*, 2017.

[7]Ole Nørregaard Jensen. Modification-specific proteomics: characterization of post-translational modifications by mass spectrometry. *Current opinion in chemical biology*, 8(1):33–41, 2004.

[8]Brian T Weinert, Christian Schölz, Sebastian A Wagner, Vytautas Iesmantavicius, Dan Su, Jeremy A Daniel, and Chunaram Choudhary. Lysine succinylation is a frequently occurring modification in prokaryotes and eukaryotes and extensively overlaps with acetylation. *Cell reports*, 4(4): 842–851, 2013.

[9]Zhihong Zhang, Minjia Tan, Zhongyu Xie, Lunzhi Dai, Yue Chen, and Yingming Zhao. Identification of lysine succinylation as a new post-translational modification. *Nature chemical biology*, 7(1):58–63, 2011.

[10]Gozde Colak, Zhongyu Xie, Anita Y Zhu, Lunzhi Dai, Zhike Lu, Yi Zhang, Xuelian Wan, Yue Chen, Yoon H Cha, Hening Lin, et al. Identification of lysine succinylation substrates and the succinylation regulatory enzyme cobb in escherichia coli. *Molecular & Cellular Proteomics*, 12(12):3509–3520, 2013.

[11]Mingkun Yang, Yan Wang, Ying Chen, Zhongyi Cheng, Jing Gu, Jiaoyu Deng, Lijun Bi, Chuangbin Chen, Ran Mo, Xude Wang, et al. Succinylome analysis reveals the involvement of lysine succinylation in metabolism in pathogenic mycobacterium tuberculosis. *Molecular & Cellular Proteomics*, 14(4):796–811, 2015.

[12]Weibo Jin and Fangli Wu. Proteome-wide identification of lysine succinylation in the proteins of tomato (solanum lycopersicum). *PloS one*, 11(2):e0147586, 2016.

[13]Longxiang Xie, Juan Li, Wanyan Deng, Zhaoxiao Yu, Wenjie Fang, Min Chen, Wanqing Liao, Jianping Xie, and Weihua Pan. Proteomic analysis of lysine succinylation of the human pathogen histoplasma capsulatum. *Journal of proteomics*, 154:109–117, 2017.

[14]Yumei Zhang, Guangyuan Wang, Limin Song, Ping Mu, Shu Wang, Wenxing Liang, and Qi Lin. Global analysis of protein lysine succinylation profiles in common wheat. *BMC genomics*, 18(1):309, 2017.

[15]Xiaowei Zhao, Qiao Ning, Haiting Chai, and Zhiqiang Ma. Accurate in silico identification of protein succinylation sites using an iterative semi-supervised learning technique. *Journal of theoretical biology*, 374:60–65, 2015.

[16]Kousik Kundu, Fabrizio Costa, Michael Huber, Michael Reth, and Rolf Backofen. Semi-supervised prediction of sh2-peptide interactions from imbalanced high-throughput data. *PloS one*, 8(5):e62732, 2013.

[17]Hao-Dong Xu, Shao-Ping Shi, Ping-Ping Wen, and Jian-Ding Qiu. Succfind: a novel succinylation sites online prediction tool via enhanced characteristic strategy. *Bioinformatics*, 31(23):3748–3750, 2015.

[18]Yan Xu, Ya-Xin Ding, Jun Ding, Ya-Hui Lei, Ling-Yun Wu, and Nai-Yang Deng. isuc-pseaac: predicting lysine succinylation in proteins by incorporating peptide position-specific propensity. *Scientific reports*, 5:10184, 2015.

[19]Jianhua Jia, Zi Liu, Xuan Xiao, Bingxiang Liu, and Kuo-Chen Chou. isuc-pseopt: identifying lysine succinylation sites in proteins by incorporating sequence-coupling effects into pseudo components and optimizing imbalanced training dataset. *Analytical biochemistry*, 497:48–56, 2016.

[20]Jianhua Jia, Zi Liu, Xuan Xiao, Bingxiang Liu, and Kuo-Chen Chou. psuc-lys: predict lysine succinylation sites in proteins with pseaac and ensemble random forest approach. *Journal of theoretical biology*, 394:223–230, 2016.

[21]Zexian Liu, Yongbo Wang, Tianshun Gao, Zhicheng Pan, Han Cheng, Qing Yang, Zhongyi Cheng, Anyuan Guo, Jian Ren, and Yu Xue. Cplm: a database of protein lysine modifications. *Nucleic acids research*, 42(D1):D531–D536, 2014.

[22]Md Mehedi Hasan, Shiping Yang, Yuan Zhou, and Md Nurul Haque Mollah. Succinsite: a computational tool for the prediction of protein succinylation sites by exploiting the amino acid patterns and properties. *Molecular BioSystems*, 12 (3):786–795, 2016.

[23]Md Mehedi Hasan, Mst Shamima Khatun, Md Nurul Haque Mollah, Cao Yong, and Dianjing Guo. A systematic identification of species-specific protein succinylation sites using joint element features information. *International journal of nanomedicine*, 12:6303, 2017.

[24]Yosvany López, Abdollah Dehzangi, Sunil Pranit Lal, Ghazaleh Taherzadeh, Jacob Michaelson, Abdul Sattar, Tatsuhiko Tsunoda, and Alok Sharma. Sucstruct: Prediction of succinylated lysine residues by using structural properties of amino acids. *Analytical biochemistry*, 527:24–32, 2017.

[25]Abdollah Dehzangi, Yosvany López, Sunil Pranit Lal, Ghazaleh Taherzadeh, Jacob Michaelson, Abdul Sattar, Tatsuhiko Tsunoda, and Alok Sharma. Pssm-suc: Accurately predicting succinylation using position specific scoring matrix into bigram for feature extraction. *Journal of theoretical biology*, 425:97–102, 2017.

[26]Abdollah Dehzangi, Yosvany López, Sunil Pranit Lal, Ghazaleh Taherzadeh, Abdul Sattar, Tatsuhiko Tsunoda, and Alok Sharma. Improving succinylation prediction accuracy by incorporating the secondary structure via helix, strand and coil, and evolutionary information from profile bigrams. *PloS one*, 13(2): e0191900, 2018.

[27]Shandar Ahmad and Akinori Sarai. Pssm-based prediction of dna binding sites in proteins. *BMC bioinformatics*, 6(1):33, 2005.

[28]Zexian Liu, Jun Cao, Xinjiao Gao, Yanhong Zhou, Longping Wen, Xiangjiao Yang, Xuebiao Yao, Jian Ren, and Yu Xue. Cpla 1.0: an integrated database of protein lysine acetylation. *Nucleic acids research*, 39(suppl_1):D1029–D1034, 2010.

[29]Weizhong Li and Adam Godzik. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22 (13):1658–1659, 2006.

[30]Kuo-Chen Chou. A sequence-coupled vector-projection model for predicting the specificity of galnac-transferase. *Protein Science*, 4(7):1365–1383, 1995.

[31]Kuo-Chen Chou. Prediction of protein signal sequences and their cleavage sites. *Proteins: Structure, Function, and Bioinformatics*, 42(1):136–139, 2001.

[32]Jianhua Jia, Zi Liu, Xuan Xiao, Bingxiang Liu, and Kuo-Chen Chou. ippi-esml: an ensemble classifier for identifying the interactions of proteins by incorporating their physicochemical properties and wavelet transforms into pseaac. *Journal of theoretical biology*, 377:47–56, 2015.

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

[34]M Saifur Rahman, Md Khaledur Rahman, M Kaykobad, and M Sohel Rahman. isgpt: An optimized model to identify sub-golgi protein types using svm and random forest based feature selection. *Artificial intelligence in medicine*, 84: 90–100, 2018.

[35]M Saifur Rahman, Swakkhar Shatabda, Sanjay Saha, M Kaykobad, and M Sohel Rahman. Dpp-pseaac: a dna-binding protein prediction model using chou's general pseaac. *Journal of theoretical biology*, 452:22–34, 2018.

[36]Jia-Ming Chang, Emily Chia-Yu Su, Allan Lo, Hua-Sheng Chiu, Ting-Yi Sung, and Wen-Lian Hsu. Psldoc: Protein subcellular localization prediction based on gapped-dipeptides and probabilistic latent semantic analysis. *Proteins: Structure, Function, and Bioinformatics*, 72(2):693–710, 2008.

[37]Runtao Yang, Chengjin Zhang, Rui Gao, and Lina Zhang. 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, 2016.

[38]Kuo-Chen Chou. Some remarks on protein attribute prediction and pseudo amino acid composition. *Journal of theoretical biology*, 273(1):236–247, 2011.

[39]Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517, 2007.

[40]David Martin Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. 2011.

[41]Douglas G Altman and J Martin Bland. Diagnostic tests. 1: Sensitivity and specificity. *BMJ: British Medical Journal*, 308(6943):1552, 1994.

[42]Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27 (8):861–874, 2006.

[43]Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine*

learning, pages 233–240. ACM, 2006.

[44]Stephen F Altschul, Thomas L Madden, Alejandro A Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–3402, 1997.