Prediction of Lysine-Malonylation Sites via Sequential and Physicochemical Features

Asif Ahmed
Kenedy Sarkar
Yeazullah Aziz
Toha Khan

Department of Computer Science and Engineering

United International University

A thesis submitted for the degree of

BSc in Computer Science & Engineering

September 2018

Declaration

I, Asif Ahmed, Kenedy Sarkar, Yeazullah Aziz, Toha Khan, declare that this thesis titled, Prediction of Lysine-Malonylation Sites via Sequential and Physicochemical Features and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a BSc degree at United International University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at United International University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ______________________________

Date: ______________________________

(Asif Ahmed, Kenedy Sarkar, Yeazullah Aziz, Toha Khan)

Certificate

I do hereby declare that the research works embodied in this thesis entitled Thesis Title is the outcome of an original work carried out by Asif Ahmed, Kenedy Sarkar, Yeazullah Aziz, Toha Khan under my supervision.

I further certify that the dissertation meets the requirements and the standard for the degree of BSc in Computer Science and Engineering.

Signed: ______________________________

Date: ______________________________

Dr. Swakkhar Shatabda

Department of Computer Science and Engineering,

United International University,

Dhaka-1209, Bangladesh.

Abstract

Lysine Malonylation is Post Translational Modification responsible for Type-2 diabetes, Cancer etc. It is a challenging problem as the data from kmal studies are highly imbalanced. In this work we propose Hybrid sampling a combination of RUS and SMOTE at certain ratios in combination with mutual information feature selection, Balanced Random Forest to solve this problem.

Acknowledgements

We would like to acknowledge our parents for their support, advice and patience.

We would like thank our supervisor Dr. Swakkhar Shatabda for exposing us to an important real world problem.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Computational prediction of Lysine Malonylation can help medical researcher understand better about impact of Kmal in diseases. Computational identification can also help select some sites to experiment on lab. Working with huge amount of protein sequences to find the correct one to conduct experiments is hard. For this reasons, we like to narrow it down by our work of predicting possible Malonylation sites. This will help researcher to experiment on those particular protein sequences rather than randomly picking protein sequences.

1.1.1 Lysine Malonylation

Lysine Malonylation is a recently discovered Post Translational Modificaiton(PTM) that plays a role in Type 2 diabetes, Glucose and Fatty Acid metabolism. Malonylation was first identified in both mammalian and bacterial cells in 2011[1–6]. Existing Malonylation sites are obtained from proteomic studies. Lysine Malonylation was observed that malonylation plays a potential role in type 2 diabetes, whereas further bioinformatic analysis of the proteomic results revealed the enrichment of malonylated proteins in metabolic pathways, especially the pathways of glucose and fatty acid metabolisms[2, 7–10].

1.1.2 Site Identification

Mass Spectrometry (MS)-based experiments, Isotopic labeling, Chemical probe, Affinity enrichment and Label free quantitative proteomics are some methods that were used to identify Lysine Malonylation sites[1, 3]. Experiments are costly in terms of infrastructure and and time consuming. It is very hard to conduct these experiments for many different species at various conditions. computational methods facilitate hypothesis-driven experimental validation. Its less costly, faster and more flexible in both methods than InVitro experiments[11]

1.2 Objectives of the Thesis

The primary objective of the thesis is to improve upon existing work on Lysine Malonylation problem. Also learn from their mistake and correct those segments. The correct application of machine learning in Bioinformatics to solve a problem is more important, since it very easy to overfit and get incorrect estimation of model performance. Furthermore to apply the knowledge gained from this problem to other PTM and Protein classification problems. Coming up with new approach such as new sampling method, feature generation and selection method for highly imbalanced data can ease work on future bioinformatics problems.

1.3 Thesis Contributions

The work of Lysine Malonylation has generated four tools for better prediction. But each come with their own weaknesses. Low amount of data, obsolete proteins, mislabeled proteins in multiple different tools, highly imbalanced data are some of the motivators for our work.

Most work on Kmal provide results in metrics which are skewed by data imbalance. We use other metrics such as Cohen's kappa, auPR alongside regular ones to provide our results.

We found that obsolete proteins are not usually checked for even in latest research, but only following given dataset. There is problem even in experimentally validated positive and negative sites as there are some sites mislabeled between Mal-Lys and Malopred. There is also challenge of dummy Amino Acid.

In our work most focus has been on balancing method. Even using only structural and physicochemical features it is possible to get better results through better sampling and classifier parameter tuning. We have experimented by combining multiple undersampling methods with SMOTE based oversampling. All other works have used some form of feature selection, we found that feature selection does not always improve performance rather degrades it in some cases.

1.4 Organization of the Thesis

The thesis is organized as follows:

Chapter 2 provides related works.

Chapter 3 presents the proposed method.

Chapter 4 discusses the results and experimental analysis.

Chapter 5 presents the conclusions, summaries the thesis contributions, and discusses the future works.

Chapter 2

Related Work

2.1 Mal-Lys

Mal-Lys is the first reported work on computational prediction of Kmal sites. They have used asymmetric window for kmal classification. Their window size is 16 with 6 upstream and 9 downstream. Their method of performance measurement is only shown in ROC with LOO, 6, 8, 10 fold cross validation in combination with independent test set. [12]

Their choice of classification algorithm is SVM and no sampling is mentioned. Sequential and physicochemical properties were used for feature construction. They have used mRMR feature selection method to improve their results.

The main drawback of Mal-Lys is that they have used small training set and even smaller independent test set. Their independent test set contains only 25 positive sites, most of which belong to same protein. Unlike Sprint-Mal they mixed their kmal sites, so same proteins are in test and train set with different sites.

As ROC is not a very good measure for imbalanced dataset without other metrics to compare, the only other option is to query their web server to generate other necessary metrics.

2.2 Malo-Pred

Malopred is followed by Mal-Lys for Kmal classification. They have used SVM for classification with no metion of any sampling methods. Physicochemical, Evolutionary and Sequential features were used for feature set construction. Information Gain was used as a feature seleciton method.[13]

They have used metrics such as auROC, MCC, Accuracy, Sensitivity, Specificity to measure their performance. Some of the positive kmal sites from Malopred are labeled negative in Mal-Lys and viceversa. The window size was chosen 25 to be optimal for their working method, with 12 Amino Acid residues on each side.

2.3 Sprint-Mal

Sprint-Mal web server is created from PLMD 3.0 database as well as previous studies. For their work they have used 1287 proteins for Mouse, 937 proteins for Humans and 112 proteins for Bacteria. The dataset they have used is highly imbalanced at ratios such as 1:11, 1:21, 1:23, 1:31 etc. The window size for their work is 17 with 8 flaking Amino Acid residues on each side.[14]

They have done 1:3 Random under sampling to negate the impact imbalance ratio. Structural, Physicochemical, Evolutionary and Sequential features were used for feature set construction. Sequence Forward feature selection was used reduce dimensionality and improve performance.

They have used metrics such as auROC, MCC, Accuracy, Sensitivity, Specificity to measure their performance. Some of their dataset is erroneous as the some of the same proteins are used in human test and train. Human test also has one duplicate proteins.

They have compared their performance to both Mal-Lys and Malopred by inputting their test set in their respective servers. Their reported result beat both Mal-Lys and Malopred. But due to obsolescence of some of their proteins, we have chosen to work with their dataset with obsolete proteins removed.

2.4 Kmal-Sp

Kmal-Sp is the most recent work on Lysine Malonylation. They have also used 25 size window with 12 flaking Amino Acid residues on each side. The have done performance comparison with all three previous works. The method of their test was to query other servers with their test set and generate comparison metrics. [15]

They have used metrics such as Precision, auROC, MCC, Accuracy, Sensitivity, Specificity to measure their performance. Obsolete proteins are present in their dataset as they were not removed from the dataset collected from Malopred.

2.5 Summary

Previous work has been done on the problem of Kmal. But the more tools developed the higher confidence for a researcher to select sites to test based on voting from multiple Kmal tools. Obsolesce, mistagging and emergence of newer tested proteins make it more important to develop a tools in the same problem domain.

Chapter 3

Proposed Method

We propose a new method for prediction of Lysine Malonylation sites in protein through sequential and physicochemical feature generation, mutual information based feature selection, ratio based undersampling through Instance Hardness Reduction and finally classification through SVM.

3.1 Data Collection

Data sets are collected from previous work on computational predication of Lysine Malonylation sites [12–14] as well as from PLMD 3.0 database [16]. The positive sites in experimentally verified protein is chosen as positive sites and rest with certain flaking amount on both upstream and downstream is chosen as negative sites.

3.2 Feature Extraction Techniques

Below we present a list of techniques used in relevant literature for feature extraction from protein.

3.2.1 Feature Extraction

- Enhanced Amino Acid Composition (EAAC)
- Composition/Transition/Distribution Composition (CTDC)
- Quasi Sequence Order (QSO)

- Pseudo Amino Acid Composition (PseAAC)

3.2.2 EAAC

EAAC is similar to AAC but it has a window that slides from N terminus to C terminus by certain amount until sequence length is reached. [17]

$$f(t, win) = \frac{N(t, win)}{L(win)}, \quad t \in \{A, C, D, ..., Y\}, \quad t = \{win1, win2, ..., win(L-K+1)\}$$

If window size is K, then the last window is, $L - K + 1$. Here, L(win) is the length of sliding window and $N(t, win)$ is the count of Amino Acid residues in that window of protein seqeunce. The feature vector here is $(L - K + 1) * 20$.

3.2.3 CTD

CTD stands for Composition Transition Distribution. It has three part CTDC, CTDT, CTDD. CTDC is the count of various properties of Amino Acids divided into groups. Hydrophobicity can be divided into polar, neutral, hydrophibic, secondary structure can be helix, strand, coil. These physicochemical properties with grouping can be found on AAINDEX.

$$f(t) = \frac{N(t)}{L}, \quad t \in \{polar, neutral, hydrophobic\}$$

The set t can be constructed also for Secondary Structure, Solvent Accessibility, Charge, Polarizibility, Van Der Waals Volume etc.

CTDT is the transition from one group to another in the same physicochemical property. A modified version can be tested with 3 transitions instead of 2.

$$f(t_1, t_2) = \frac{N(t_1, t_2) + N(t_2, t_1)}{L - 1}$$

Here, L is the protein sequence length and Amino Acid residues t_1, t_2 is in format, {(group1,group2),(group2,group3),(group3,group1)}

$$t_1, t_2 \in \{(polar, neutral), (neutral, hydrophobic), (hydrophobic, polar)\}$$

CTDD is similar to AAC in way that it count the occurence of each group for given physicochemical property then normalizes it by the length of the sequence. It does it at occurance of first residue of a given group and at $(25, 50, 75, 100)\%$ occurence of any group divided by sequence length. [17]

3.2.4 Sequence Order Coupling

If sequence length is N and protein sequence N-terminus to C-terminus is $R_1, R_2, ..., R_N$ then d is the rank of sequence order coupling. It is given by,

$$\tau_d = \sum_{i=1}^{N-d} (d_{i,i+d})^2, \quad d = 1, 2, 3..., nlag$$

Where, $d_{i,i+d}$ is entry in distance matrix such as physicochemical, chemical distance matrix between two Amino Acids. If both matrices are used then feature vector will be, $nlag * 2$. Example, if $nlag = 3$ then, $d = 1, 2, 3$. For, $d = 1$, then rank will be, τ_1 which is between R_1R_2, R_2R_3,...,$R_{N-1}R_N$. For, $d = 2$, τ_2 will be R_1R_3, R_2R_4,...,$R_{N-2}R_N$ and finally, for, $d = 3$, τ_2 will be R_1R_4, R_2R_5,...,$R_{N-3}R_N$.[17]

3.2.5 Quasi Sequence Order

Similar to SOC it also uses τ_d. It will have $20 + nlag$ size feature vector if 2 distance matrix is used otherwise if both used then feature vector will be $(20 + nlag) * 2$. The first 20 features are given by,

$$X_r = \frac{f_r}{\sum_{d=1}^{20} f_r + w * \sum_{d=1}^{d} \tau_d}, \quad r = 1, 2, 3, ..., nlag$$

The next 21 to $nlag$ features are defined by,

$$X_r = \frac{w\tau_d - 20}{\sum_{d=1}^{20} f_r + w * \sum_{d=1}^{d} \tau_d}, \quad r = 21, 22, 23, ..., 20 + nlag$$

Here, f_r is normalized occurence of Amino Acid type r and w is the weighting factor used as 0.1. [17]

3.2.6 PseAAC

This group of features use original properties proposed in [18, 19], which are Hydrophobicity $H_1^o(i)$, Hydrophilicity $H_2^o(i)$ and Side Chain Mass $M^o(i)$ for $i = 1, 2, ...20$ for 20 Natural Amino Acids [17]. They are normalized using formula below,

$$H_1(i) = \frac{H_1^o(i) - \frac{1}{20}\sum_{i=0}^{20} H_1^o(i)}{\sqrt{\frac{\sum_{i=0}^{20}[H_1^o(i) - \sum_{i=0}^{20} H_1^o(i)]^2}{20}}}$$

$H_2^o(i)$ and $M^o(i)$ is normalized in a similar manner.

Next correlation function is calculated which is the average value of 3 amino acid properties. This formula can be represented in compact summation format for more properties.

$$\Theta(R_i, R_j) = \frac{1}{3}\{[H_1(R_i) - H_1(R_j)]^2 + [H_2(R_i) - H_2(R_j)]^2 + [M(R_i) - M(R_j)]^2\}$$

Sequence order correlated features calculated by,

$$\theta_1 = \frac{1}{N-1}\sum_{i=1}^{N-1} \Theta(R_i, R_{i+1})$$

$$\theta_2 = \frac{1}{N-2}\sum_{i=1}^{N-2} \Theta(R_i, R_{i+2})$$

for λ order,

$$\theta_\lambda = \frac{1}{N\lambda}\sum_{i=1}^{N-\lambda} \Theta(R_i, R_{i+\lambda})$$

Where, $\lambda < N$. If f_i is the normalized occurence of amino acid i in protein sequence then a set of $20 + \lambda$ features are called Pseudo Amino Acid Composition.

$$X_c = \frac{f_c}{\sum_{r=1}^{20} f_r + w\sum_{j=1}^{\lambda}\theta_j}, \quad (1 < c < 20)$$

$$X_c = \frac{w\theta_{c-20}}{\sum_{r=1}^{20} f_r + w\sum_{j=1}^{\lambda}\theta_j}, \quad (21 < c < 20 + \lambda)$$

3.2.7 Importance of Normalization

In all of the papers we reviewed, normalization was an integrated approach in feature extraction. It brings different types of features to same scale. In most feature extraction algorithms, the last step is to divide by the window size. For example, in AAC dividing it by window size provides values between $[0, 1]$, adding all of which gives a total of 1.

Machine learning algorithms work better when different types of features are normalized respectively. If a classifier uses L2 (Euclidean) distance and the range of values vary greatly then it will provide inconsistent results as the latter feature will dominate.

3.3 Feature Selection

3.3.1 Regularization L1, L2

Regularization helps by reducing model complexity by removing unnecessary features. Logistic Regression with L1 feature selection only grows logarithmically in terms of irrelevant features.

L1 regularization, uses a penalty term which encourages the sum of the absolute values of the parameters to be small. The second, L2 regularization, encourages the sum of the squares of the parameters to be small. It has frequently been observed that L1 regularization in many models causes many parameters to equal zero, so that the parameter vector is sparse. This makes it a natural candidate in feature selection settings, where it is believed many features should be ignored. [20]

3.3.2 Mutual Information

For discrete or categorical variables, the Mutual Information(MI), I of two variables x and y is defined in terms of their joint probability and their marginal probability. [21]

$$I(x, y) = \sum_i \sum_j p(x, y) log \frac{p(x, y)}{p(x)p(y)}$$

For continuous Random Variables,

$$I(x, y) = \int_i \int_j p(x, y) log \frac{p(x, y)}{p(x)p(y)} dydx$$

3.3.3 Recursive Feature Elimination

The RFE algorithm initially fits all features to model, then each of the features are ranked with importance to model. At each iteration top ranked features are retained. It recursively selects features consiering smaller subsets. The least important features are pruned until desired number of features reached. [22]

3.4 Handling Imbalanced Data

Cost sensitive learning, Biased classifier, Sampling, Hybrid Sampling are some of the methods used to tackle imbalanced data. We propose Hybrid Sampling to improve results in highly imbalanced datasets. In Hybrid Sampling both Under and Oversampling are combined at a certain ratio such as 1:2, 1:1, 1:1.5 to achieve better results.

3.5 Workflow

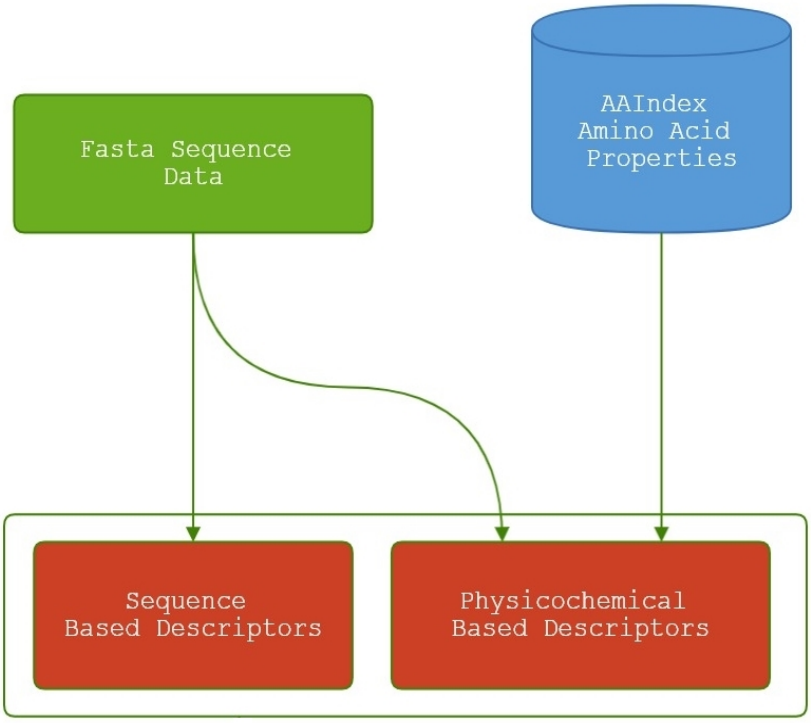
Fasta Sequence Data
AAIndex Amino Acid Properties
Sequence Based Descriptors
Physicochemical Based Descriptors

Figure 3.1: Diagram of current work flow producing best results.

3.6 Summary

Initially the problem is formulated and raw data is collected. Obsolete and erroneous data is removed before feature extraction. Next feature extraction is performed using above mentioned methods for all train, test data.

Data imbalance affects result greatly, in order reduce that Cross validation is used with independent test sets, various imbalance independent metrics are used, also a combination of undersampling and oversampling is performed.

Chapter 4

Experimental Analysis

Here, we present our findings based on various dataset presented on previous work and our independent work on different organisms. 10 Fold Cross Validation with per fold sampling and feature selection, as well as independent test set was used reduce the impact of imbalance.

4.1 Datasets

4.1.1 Sprint-Mal

In this data, there are two different training set and three different test set. Each test and training data set generated from different number of protein sequences. As original source contained obsolete proteins and duplicate proteins, these changes are reflected in table below.

The train ratio of mouse is 1:11, mouse test 1:21, human train 1:28, human test 1:22, bacteria test 1:42. Cleaned dataset for human has a much different ratio distribution than original source.

Table 4.1: Number of instances in dataset.

Dataset Name	**Number of Protein**	**Malonylation Sites**	**Non-malonylation Sites**
Mouse Train	1150	3397	37854
Mouse Test	120	322	6728
Human Train	837	1554	43589
Human Test	119	207	4570
Bacteria Test	112	44	1845

4.2 Sampling

Random Under Sampling(RUS), randomly selects a certain amount to keep for classification and it discards the rest. SMOTE is an oversampling technique that generate new samples based on current data points. We have found the combining these two at certain ratio for highly imbalanced dataset works best. The majority class is first undersampled to certain amount then minority class is oversampled to certain amount. After sampling, in the new sampled dataset majoirty class still remains majority and minority class still remains minority albeit in a smaller ratio than original. In our work undersampling the majority class by $\frac{1}{8}$ and oversampling the minority class by $\frac{1}{6}$ gives the best result. We call this hybrid sampling.

4.3 Selection of Classifiers

We use different kinds algorithms like ANN, SVM, Random Forest, Cart. Among These, SVM is widely use in Bioinformatics. SVM is also very strong algorithm for predicting binary classification problems. We choose classifier based on experimental analysis by observing multiple parameters. Weight Balanced Random Forest was chosen as our classification algorithm.

4.4 Optimal Window Size Selection

Choosing optimal window size is challenging. We test on 5k5, 6k6, 7k7, 8k8, 9k9, 10k10, 11k11, 12k12, 13k13, 14k14, 15k15, 16k16, 17k17, 18k18, 19k19 window size to test model. Based on experiment in combination with our classifier and balancing method we choose applicable window.

4.5 Performance Evaluation

In previous work computational KMal site prediction metrics such as Accuracy(ACC), Area under the Receiver Operating Characteristics Curve(AUROC), Sensitivity(SN), Specificity(SP), Mathew's correlation co-efficient(MCC), Precision(Pr) are used.

For binary classifier, let us assume TP is the number of true positive or the positive samples classified correctly, TN is the number of True Negatives or the negative samples classified correctly, FP is the number of False Positives or incorrectly classified the negative samples as positive(Type-1 error), FN is the number of False Negatives or the positive samples incorrectly classified as negative(Type-2 error). Along with increasing the number of TP, TN the secondary goal is to reduce FN as much as possible. The sensitivity equation is defined as:

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

The higher the value of sensitivity the more confidence for KMal site prediction. The value of this metric varies from 0 to 1. Specificity is 1 - Sensitivity. The specificity equation is defied as:

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

Precision is the number of samples actually positive divided by the total number of samples labled as positive. It ranges from 0 to 1. A high precision means every instance was relevent. It is defiend as:

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

Accuracy is the ratio of correctly classified instances to all instances in dataset defied as follwing:

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

Its range also varies from 0 to 1. Mathew's Correlation Coefficient (MCC) is another metric for performance evaluation. MCC is usually regarded as a balanced measure. It ranges from -1 to +1 with -1 representing negative classification correlation and +1 as positive classification correlation. MCC is defined as:

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

4.6 Experimental Results

Experimental results are shown on Mouse training data and Mouse independent data. Unless otherwise mentioned above convention holds true. All result are on 21 size window with 10 Amino Acid residues on each sides.

4.6.1 Implementation Details

The codes are implemented in Python language in combination with Scikit Learn machine learning library, as well numpy, pandas libraries. Our method is named as Hybrid in performance comparison to reflect Hybrid Balancing method. Class weight balanced Random Forest Classifier is used, all RF results shown are from balanced Random Forest Classifier. We create a server with our model for predicting Kmal sites `http://mallysml.pythonanywhere.com/`.

4.6.2 Classifier Performance

Table 4.2: Classifier Comparison

<table>
<tr><th colspan="8">10 Fold on Mouse Dataset</th></tr>
<tr><th>Classifier Name</th><th>ACC</th><th>AUROC</th><th>AUPR</th><th>MCC</th><th>SN</th><th>SP</th><th>Kappa</th></tr>
<tr><td>Cart Train</td><td>0.7567</td><td>0.6988</td><td>0.1174</td><td>0.1588</td><td>0.4654</td><td>0.7824</td><td>0.1316</td></tr>
<tr><td>Cart Independent Test</td><td>0.780</td><td>0.7227</td><td>0.0687</td><td>0.1361</td><td>0.4829</td><td>0.7932</td><td>0.0943</td></tr>
<tr><td>ANN Train</td><td>0.6711</td><td>0.6935</td><td>0.1167</td><td>0.1580</td><td>0.5963</td><td>0.6777</td><td>0.1111</td></tr>
<tr><td>ANN Independent Test</td><td>0.676</td><td>0.7221</td><td>0.0717</td><td>0.1489</td><td>0.6700</td><td>0.6759</td><td>0.0813</td></tr>
<tr><td>AdaBoost Train</td><td>0.6886</td><td>0.6853</td><td>0.1141</td><td>0.1496</td><td>0.5547</td><td>0.7004</td><td>0.1094</td></tr>
<tr><td>AdaBoost Independent Test</td><td>0.712</td><td>0.7171</td><td>0.0706</td><td>0.1436</td><td>0.6054</td><td>0.7168</td><td>0.0850</td></tr>
<tr><td>SVM Train</td><td>0.6882</td><td>0.7067</td><td>0.1135</td><td>0.1640</td><td>0.5934</td><td>0.6961</td><td>0.1159</td></tr>
<tr><td>SVM Independent Test</td><td>0.683</td><td>0.6979</td><td>0.0604</td><td>0.1045</td><td>0.5510</td><td>0.6886</td><td>0.0589</td></tr>
<tr><td>BRF Train</td><td>0.7974</td><td>0.7220</td><td>0.1849</td><td>0.1854</td><td>0.4385</td><td>0.8290</td><td>0.1647</td></tr>
<tr><td>BRF Independent Test</td><td>0.814</td><td>0.7396</td><td>0.0752</td><td>0.1575</td><td>0.4693</td><td>0.8298</td><td>0.1179</td></tr>
<tr><th colspan="8">Balanced Random Forest with 'U','X' in Mouse Dataset</th></tr>
<tr><th>Classifier Name</th><th>ACC</th><th>AUROC</th><th>AUPR</th><th>MCC</th><th>SN</th><th>SP</th><th>Kappa</th></tr>
<tr><td>BRF Train</td><td>0.7803</td><td>0.7122</td><td>0.1245</td><td>0.1756</td><td>0.4495</td><td>0.8100</td><td>0.1525</td></tr>
<tr><td>BRF Independent Test</td><td>0.785</td><td>0.7327</td><td>0.0772</td><td>0.1557</td><td>0.5093</td><td>0.7978</td><td>0.1106</td></tr>
<tr><th colspan="8">Balanced Random Forest with 'U','X' in Mouse Dataset without Feature Selection</th></tr>
<tr><th>Classifier Name</th><th>ACC</th><th>AUROC</th><th>AUPR</th><th>MCC</th><th>SN</th><th>SP</th><th>Kappa</th></tr>
<tr><td>BRF Train</td><td>0.7895</td><td>0.7130</td><td>0.1247</td><td>0.1760</td><td>0.4327</td><td>0.8216</td><td>0.1556</td></tr>
<tr><td>BRF Independent Test</td><td>0.797</td><td>0.7310</td><td>0.0790</td><td>0.1613</td><td>0.5</td><td>0.8109</td><td>0.1176</td></tr>
</table>

4.6.3 Feature Selection Performance

Table 4.3: Under sampling at 1:1.25 Ratio comparison ANN

50 Feature							
Method Name	**ACC**	**AUROC**	**AUPR**	**MCC**	**SN**	**SP**	**Kappa**
MI 10 Fold	0.621	0.681	0.113	0.150	0.650	0.618	0.096
MI Independent Test	0.658	0.706	0.065	0.127	0.639	0.659	0.067
ANOVA 10 Fold	0.621	0.649	0.106	0.123	0.595	0.624	0.080
ANOVA Independent Test	0.675	0.675	0.061	0.107	0.568	0.679	0.059
100 Feature							
Method Name	**ACC**	**AUROC**	**AUPR**	**MCC**	**SN**	**SP**	**Kappa**
MI 10 Fold	0.634	0.699	0.116	0.159	0.650	0.633	0.104
MI Independent Test	0.720	0.715	0.066	0.128	0.557	0.727	0.077
ANOVA 10 Fold	0.626	0.672	0.109	0.135	0.612	0.628	0.088
ANOVA Independent Test	0.669	0.701	0.063	0.116	0.598	0.671	0.063
150 Feature							
Method Name	**ACC**	**AUROC**	**AUPR**	**MCC**	**SN**	**SP**	**Kappa**
MI 10 Fold	0.654	0.686	0.113	0.147	0.597	0.659	0.101
MI Independent Test	0.742	0.691	0.059	0.101	0.462	0.755	0.059
ANOVA 10 Fold	0.640	0.679	0.111	0.142	0.608	0.643	0.095
ANOVA Independent Test	0.734	0.695	0.062	0.114	0.503	0.744	0.071
200 Feature							
Method Name	**ACC**	**AUROC**	**AUPR**	**MCC**	**SN**	**SP**	**Kappa**
MI 10 Fold	0.648	0.695	0.115	0.155	0.624	0.650	0.105
MI Independent Test	0.681	0.702	0.063	0.116	0.581	0.685	0.065
ANOVA 10 Fold	0.626	0.685	0.112	0.145	0.633	0.626	0.094
ANOVA Independent Test	0.668	0.725	0.068	0.139	0.656	0.668	0.075
250 Feature							
Method Name	**ACC**	**AUROC**	**AUPR**	**MCC**	**SN**	**SP**	**Kappa**
MI 10 Fold	0.636	0.686	0.113	0.150	0.629	0.636	0.099
MI Independent Test	0.716	0.720	0.069	0.140	0.591	0.720	0.083
ANOVA 10 Fold	0.634	0.693	0.116	0.160	0.652	0.632	0.104
ANOVA Independent Test	0.669	0.701	0.063	0.116	0.598	0.671	0.063

4.6.4 Sampling Performance

Table 4.4: Under sampling at 1:1 Ratio comparison ANN

Method Name	**ACC**	**AUROC**	**AUPR**	**MCC**	**SN**	**SP**	**Kappa**
MI 10 Fold	0.574	0.689	0.114	0.155	0.561	0.721	0.091
MI Independent Test	0.738	0.702	0.0645	0.121	0.513	0.747	0.076
ANOVA 10 Fold	0.533	0.672	0.107	0.136	0.517	0.722	0.076
ANOVA Independent Test	0.699	0.693	0.061	0.109	0.540	0.706	0.063

Table 4.5: Over sampling at 1:1 Ratio comparison ANN

Method Name	**ACC**	**AUROC**	**AUPR**	**MCC**	**SN**	**SP**	**Kappa**
ANOVA 10 Fold	0.629	0.691	0.114	0.152	0.643	0.628	0.099
ANOVA Independent Test	0.704	0.700	0.063	0.116	0.551	0.710	0.068

4.6.5 Proposed Method Results

Table 4.6: Results Comparison

Method Name	**ACC**	**AUROC**	**AUPR**	**MCC**	**SN**	**SP**	**Kappa**	**PR**
SprintMal 10 Fold	0.80	0.74	-	0.213	0.49	0.81	-	-
SprintMal Ind. Test	0.90	0.76	-	0.20	0.33	0.92	-	-
Result with 'U', 'X' residues, Balanced Random Forest, ANOVA								
Hybrid 10 Fold	0.780	0.712	0.124	0.175	0.449	0.810	0.152	0.176
Hybrid Ind. Test	0.785	0.732	0.077	0.155	0.509	0.797	0.110	0.107
Result without 'U', 'X' residues, Balanced Random Forest, ANOVA								
Hybrid 10 Fold	0.797	0.722	0.126	0.185	0.438	0.829	0.164	0.184
Hybrid Ind. Test	0.814	0.739	0.0752	0.157	0.469	0.829	0.117	0.111

4.7 10 Fold ROC Curve Balanced Random Forest

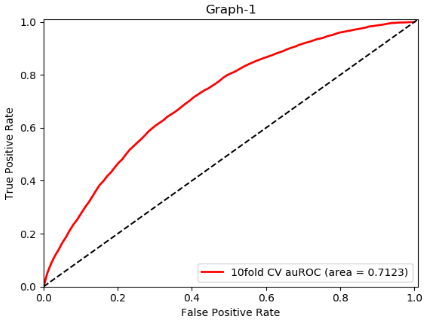

Figure 4.1: Diagram of current work flow producing best results.

4.8 Test Set Performance Comparison

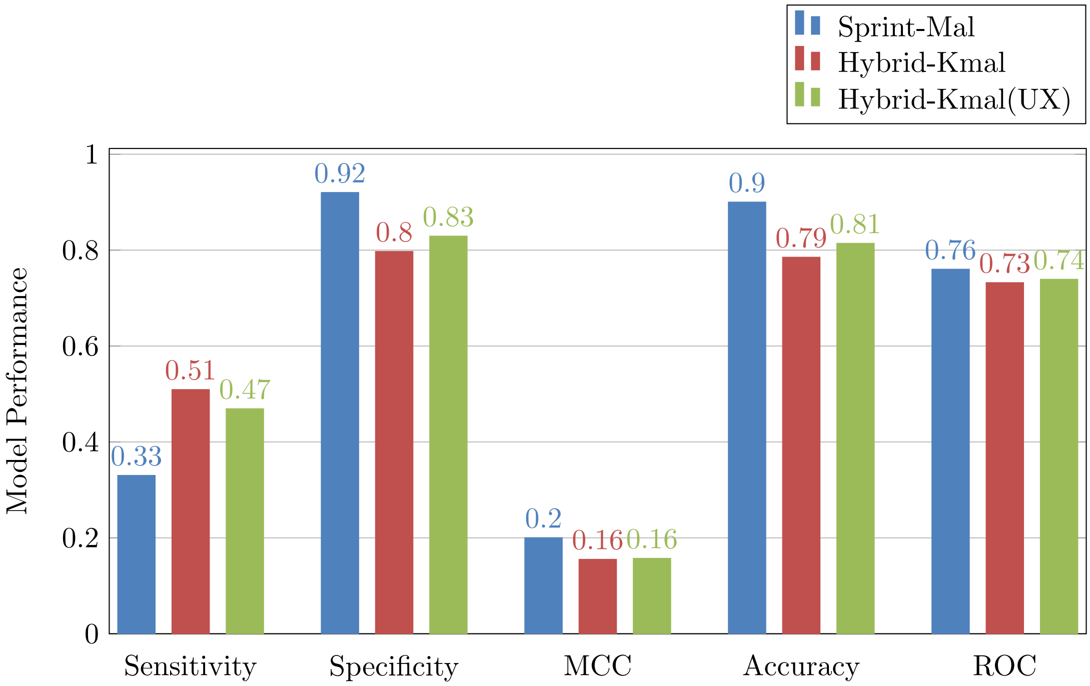

Figure 4.2: Comparison of Sprint-mal with our results (Rounded).

4.9 10 Fold Performance Comparison

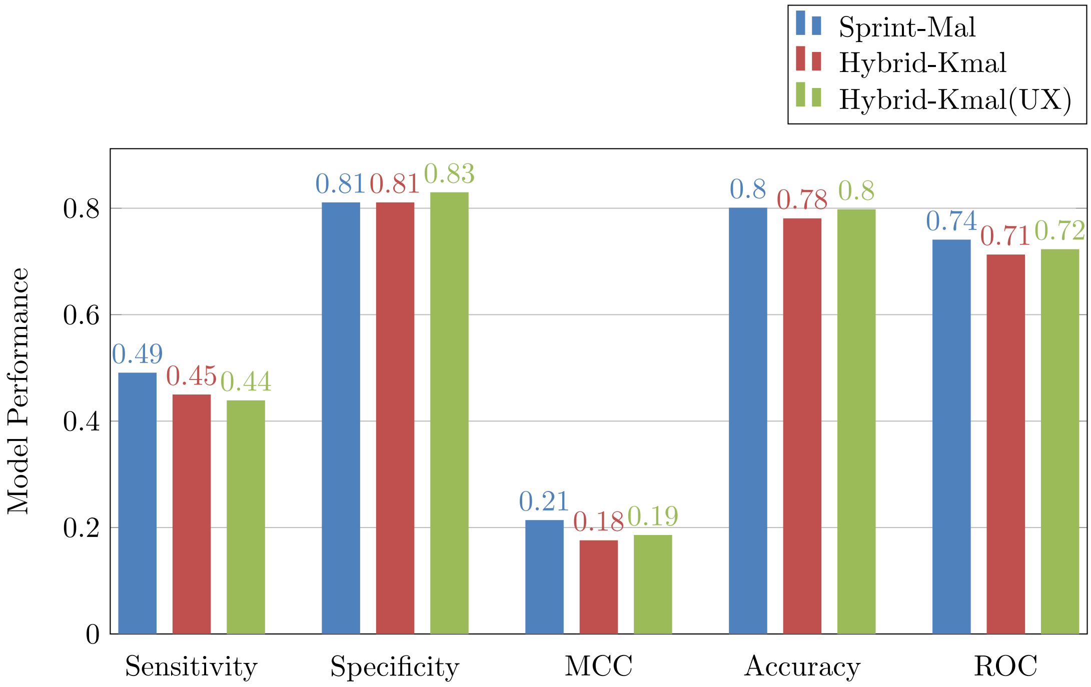

Figure 4.3: Comparison of Sprint-mal with our results (Rounded).

4.10 Summary

In this chapter we show performance of different window size selection, feature selection methods, feature generation methods, sampling and different classifiers. From these based on the best result we choose optimal parameters for our whole model.

Chapter 5

Conclusion

5.1 Summary

In this work we first collected data set for Kmal. The data was cleaned to exclude obsolete proteins. Then based on that dataset we selected the best window and good features sets. Since data is highly imbalanced, based on experimentation we found that hybrid sampling at a certain ratio for this data set works better. Feature selection was done to further improve performance on the best chosen classifier.

5.2 Conclusions

A bioinformatics research is as good as its data. In two previous papers Mal-Lys, Malopred we saw some negative peptides are labeled as positive and positives labeled as negative. Even with our current data source there are duplicate proteins for same human test set and overlapping same proteins in both human train, test.

Window selection is very important as a good window will have all necessary markers for good feature generation. Feature generation is also important as to encode most information for discrimination. In our work of Kmal Enhanced Amino Acid Composition in combination with CTD features work best. For heavily imbalanced data hybrid sampling with ratio works best. Our Hybrid sampling method undersamples majority class by $\frac{1}{8}$ and minority class is over sampled to $\frac{1}{6}$. Weight Balanced Random Forest with 250 estimators provide the best results and classification time advantage. ANOVA further improves the result by taking top ranked features from full feature set.

In previous work, most result contribution came from evolutionary and structural information. Here, with our hybrid balancing method and only using sequential and physicochemical features, our results are almost similar to them.

5.3 Future Work

Finding appropriate feature set with best hyper parameters is one of the important goals. Based on that the best Hybrid balancing ratio to improve performance further is target.

We want to create a web server with all currently available datasets to cover as much query as possible. Also we want to apply the knowledge gained to further improve other Protein Lysine based Modifications such as Ubiquitination, Acetylation, Succinylation, Sumoylation, Glycation, Methylation.

Bibliography

[1] X. Bao, Q. Zhao, T. Yang, Y. M. E. Fung, and X. D. Li, "A chemical probe for lysine malonylation," *Angewandte Chemie International Edition*, vol. 52, no. 18, pp. 4883–4886, 2013. 1, 2

[2] Y. Du, T. Cai, T. Li, P. Xue, B. Zhou, X. He, P. Wei, P. Liu, F. Yang, and T. Wei, "Lysine malonylation is elevated in type 2 diabetic mouse models and enriched in metabolic associated proteins," *Molecular & Cellular Proteomics*, vol. 14, no. 1, pp. 227–236, 2015. 1

[3] M. D. Hirschey and Y. Zhao, "Metabolic regulation by lysine malonylation, succinylation and glutarylation," *Molecular & Cellular Proteomics*, pp. mcp–R114, 2015. 2

[4] G. Colak, O. Pougovkina, L. Dai, M. Tan, H. te Brinke, H. Huang, Z. Cheng, J. Park, X. Wan, X. Liu *et al.*, "Proteomic and biochemical studies of lysine malonylation suggests its malonic aciduria-associated regulatory role in mitochondrial function and fatty acid oxidation," *Molecular & Cellular Proteomics*, pp. mcp–M115, 2015.

[5] Y. Nishida, M. J. Rardin, C. Carrico, W. He, A. K. Sahu, P. Gut, R. Najjar, M. Fitch, M. Hellerstein, B. W. Gibson *et al.*, "Sirt5 regulates both cytosolic and mitochondrial protein malonylation with glycolysis as a major target," *Molecular cell*, vol. 59, no. 2, pp. 321–332, 2015.

[6] C. Peng, Z. Lu, Z. Xie, Z. Cheng, Y. Chen, M. Tan, H. Luo, Y. Zhang, W. He, K. Yang *et al.*, "The first identification of lysine malonylation substrates and its regulatory enzyme," *Molecular & cellular proteomics*, pp. mcp–M111, 2011. 1

[7] A. L. Santos and A. B. Lindner, "Protein posttranslational modifications: roles in aging and age-related disease," *Oxidative Medicine and Cellular Longevity*, vol. 2017, 2017. 1

[8] M. Gallego and D. M. Virshup, "Post-translational modifications regulate the ticking of the circadian clock," *Nature reviews Molecular cell biology*, vol. 8, no. 2, p. 139, 2007.

[9] P. R. Gajjala, D. Fliser, T. Speer, V. Jankowski, and J. Jankowski, "Emerging role of post-translational modifications in chronic kidney disease and cardiovascular disease," *Nephrology Dialysis Transplantation*, vol. 30, no. 11, pp. 1814–1824, 2015.

[10] S. Westermann and K. Weber, "Post-translational modifications regulate microtubule function," *Nature Reviews Molecular Cell Biology*, vol. 4, no. 12, p. 938, 2003. 1

[11] K.-C. Chou, "Some remarks on protein attribute prediction and pseudo amino acid composition," *Journal of theoretical biology*, vol. 273, no. 1, pp. 236–247, 2011. 2

[12] Y. Xu, Y.-X. Ding, J. Ding, L.-Y. Wu, and Y. Xue, "Mal-lys: prediction of lysine malonylation sites in proteins integrated sequence-based features with mrmr feature selection," *Scientific reports*, vol. 6, p. 38318, 2016. 4, 7

[13] L.-N. Wang, S.-P. Shi, H.-D. Xu, P.-P. Wen, and J.-D. Qiu, "Computational prediction of species-specific malonylation sites via enhanced characteristic strategy," *Bioinformatics*, vol. 33, no. 10, pp. 1457–1463, 2016. 5

[14] G. Taherzadeh, Y. Yang, H. Xu, Y. Xue, A. W.-C. Liew, and Y. Zhou, "Predicting lysine-malonylation sites of proteins using sequence and predicted structural features," *Journal of computational chemistry*, 2018. 5, 7

[15] Y. Zhang, R. Xie, J. Wang, A. Leier, T. T. Marquez-Lago, T. Akutsu, G. I. Webb, K.-C. Chou, and J. Song, "Computational analysis and prediction of lysine malonylation sites by exploiting informative features in an integrative machine-learning framework," *Briefings in Bioinformatics*, 2018. 5

[16] H. Xu, J. Zhou, S. Lin, W. Deng, Y. Zhang, and Y. Xue, "Plmd: an updated data resource of protein lysine modifications," *Journal of Genetics and Genomics*, vol. 44, no. 5, pp. 243–250, 2017. 7

[17] Z. Chen, P. Zhao, F. Li, A. Leier, T. T. Marquez-Lago, Y. Wang, G. I. Webb, A. I. Smith, R. J. Daly, K.-C. Chou *et al.*, "ifeature: a python package and web server for features extraction and selection from protein and peptide sequences," *Bioinformatics*, vol. 1, p. 4, 2018. 8, 9

[18] K.-C. Chou, "Prediction of protein cellular attributes using pseudo-amino acid composition," *Proteins: Structure, Function, and Bioinformatics*, vol. 43, no. 3, pp. 246–255, 2001. 9

[19] ——, "Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes," *Bioinformatics*, vol. 21, no. 1, pp. 10–19, 2004. 9

[20] A. Y. Ng, "Feature selection, l 1 vs. l 2 regularization, and rotational invariance," in *Proceedings of the twenty-first international conference on Machine learning.* ACM, 2004, p. 78. 11

[21] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005. 11

[22] X. Zhang, X. Lu, Q. Shi, X.-q. Xu, E. L. Hon-chiu, L. N. Harris, J. D. Iglehart, A. Miron, J. S. Liu, and W. H. Wong, "Recursive svm feature selection and sample classification for mass-spectrometry and microarray data," *BMC bioinformatics*, vol. 7, no. 1, p. 197, 2006. 11

[23] T. K. Ho, "Random decision forests," in *Document analysis and recognition, 1995., proceedings of the third international conference on*, vol. 1. IEEE, 1995, pp. 278–282. 27

[24] W. N. Venables and B. D. Ripley, "Tree-based methods," in *Modern Applied Statistics with S.* Springer, 2002, pp. 251–269. 27

[25] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986. 27

[26] M. W. Gardner and S. Dorling, "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," *Atmospheric environment*, vol. 32, no. 14-15, pp. 2627–2636, 1998. 27

[27] K.-I. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural networks*, vol. 2, no. 3, pp. 183–192, 1989. 27

Appendix A

Classifier Description

A.1 Random Forest

Random forest is an ensemble learning method for classification. It operates by creating many decision trees with different attributes[23].

A.1.1 Hyper-parameters

The main parameter for random forest is the forest size or number of trees. The more trees in the forest will get better results but the computation cost will higher and also time consuming.

Type of decision tree can also impact on random forest. Gini Index tree[24] and Information Gain tree[25] is mostly use in random forest.

A.2 Artificial Neural Networks

ANN is inspired by the biological neural networks, for example, Human Brain. ANN can be parallelized to take advantage of GPU cores, this result huge performance gains over CPU computed classification algorithms[26, 27]. In our work we have used feed forward multilayer perceptron.

A.2.1 Hyper-parameters

In feed-forward neural networks, a hidden layer is a vector of many neurons, who are connected to next layer and is not visible as the network output. The more layers added the more complex decision boundary is created to classify data points. With more layer the computation time increases.

Activation function defines output of a Neuron. There are multiple activation functions such as Tanh, Relu, Sigmoid etc. Sigmoid and Tanh squeeze the values to a narrow range. As more layers are stacked there is minor output change on large input change, this is the vanishing problem. This problem is solved by Rectified Linear units.

Learning rate is used to reduce error rate in classification. With higher the learning rate, it may overshoot optimal range and provide bad results. In case of low learning rate, it may take a long time to converge.

L2 penalty can be applied to ANN to perform regularization.

Appendix B

Acronyms

SVM: Support Vector Machine
Kmal: Lysine Malonylation
RF: Random Forest
CART: Classification and Regression Trees
SP: Specificity
SN: Sensitivity
ROC: Receiver Operating Characteristics
AUC: Area Under the Curve
MCC: Mathews Correlation Coefficient
AUPR: Area Under Precision Recall Curve
ACC: Accuracy
MI: Mutual Information
RF: Random Forest
BRF: Balanced Random Forest
SVM: Support Vector Machine
ANN: Artificial Neural Networks